

ss - finance - government - military - manufacturing - computers - internet

[This banner was supplied by SAFE Audit](#)

# Tricks of the Internet Gurus

- [Introduction](#)
  - Overview
  - Foreword
  - About the Authors

## Part I - First Things First

- [Chapter 1 - Getting Connected: The Guru's Options](#)
- [Chapter 2 - Access at All Costs](#)
- [Chapter 3 - Tricks of the Trade: Tools](#)
- [Chapter 4 - Where's the Complaint Department? Or, What to Do When Things Don't Work](#)

## Part II - Everyday Guru Facilities

- [Chapter 5 - E-Mail](#)
- [Chapter 6 - Faxing from the Internet](#)
- [Chapter 7 - FTP](#)
- [Chapter 8 - Telnet](#)
- [Chapter 9 - Usenet](#)
- [Chapter 10 - World Wide Web](#)

## Part III - A Guru's Diversions

- [Chapter 11 - Online Entertainment for the Internet Guru](#)
- [Chapter 12 - Listservs and Mailing Lists](#)
- [Chapter 13 - MBone: The Internet's Multimedia Backbone](#)
- [Chapter 14 - Community Computing](#)

## **Part IV - Finding Stuff: You Name It, a Guru Can Find It (and So Can You)**

- [Chapter 15 - Life with Archie](#)
- [Chapter 16 - Veronica](#)
- [Chapter 17 - Discussion Forums](#)
- [Chapter 18 - Gopher](#)

## **Part V - Tricks of the Professional Internet Gurus**

- [Chapter 19 - Internet in The Classroom: How Educators Use the Internet](#)
- [Chapter 20 - Business and Commerce](#)
- [Chapter 21 - Finding Information You Want](#)
- [Chapter 22 - Art on the Internet](#)
- [Appendix A - Uniform Resource Locators \(URLs\)](#)
- [Appendix B - Signature Files](#)
- [Appendix C - Standards and Where To Find Them](#)



- [Tricks of the Internet Gurus](#)
- [Tricks of the Internet Gurus](#)
  - [Copyright © 1994 by Sams Publishing](#)
  - [Trademarks](#)
- [Overview](#)
- [Foreword](#)
  - [--by James "Kibo" Parry](#)
- [About the Authors](#)
- [Introduction](#)
  - [--by Billy Barron](#)

# Tricks of the Internet Gurus

# Tricks of the Internet Gurus

Philip Baczewski  
Billy Barron  
Kevin Barron  
Robert Bickford  
Jon Callas  
Earl Fogel  
Tod Foley  
Kenny Greenberg  
David Gulbransen  
Joseph Janes  
John Katsaros  
Dave Kinnaman  
Max Metral  
Kevin Mullet  
James "Kibo" Parry  
Joseph Poirier  
Lou Rosenfeld  
Margaret Russell-Gonzalez  
Kevin Savetz  
Peter Scott  
Stephen Volan

Sams Publishing  
201 West 103rd Street

Indianapolis, Indiana 46290

**Copyright © 1994 by Sams Publishing**

FIRST EDITION

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. For information, address Sams Publishing, 201 W. 103rd St., Indianapolis, IN 46290.

International Standard Book Number: 0-672-30599-2

Library of Congress Catalog Card Number: 94-67090

97 96 95 94 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit, the number of the book's printing. For example, a printing code of 94-1 shows that the first printing of the book occurred in 1994.

*Composed in Helvetica and MCPdigital by Macmillan Computer Publishing*

*Printed in the United States of America*

## **Trademarks**

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

## **Publisher**

*Richard K. Swadley*

## **Associate Publisher**

*Jordan Gold*

## **Acquisitions Manager**

*Stacy Hiquet*

## **Managing Editor**

*Cindy Morrow*

## **Acquisitions Editor**

*Grace Buechlein*

## **Development Editor**

*Phillip W. Paxton*

## **Production Editor**

*David Bradford*

## **Editors**

*Anne Owen*

*Marla Abraham*

*Adam Banta*

*Fran Hatton*

*Sean Medlock*

## **Editorial Coordinator**

*Bill Whitmer*

## **Editorial Assistants**

*Carol Ackerman*

*Sharon Cox*

*Lynette Quinn*

## **Technical Reviewers**

*Billy Barron*

*Philip Baczewski*

## **Interviewer**

*Tod Foley*

## **Marketing Manager**

*Gregg Bushyeager*

## **Cover Designer**

*Nathan Clement*

## **Book Designer**

*Alyssa Yesh*

## **Director of Production and Manufacturing**

*Jeff Valler*

## **Imprint Manager**

*Juli Cook*

## **Manufacturing Coordinator**

*Paul Gilchrist*

## **Production Analysts**

*Dennis Clay Hager*

*Mary Beth Wakefield*

## **Graphics Image Specialists**

*Teresa Forrester*

*Clint Lahnen*

*Tim Montgomery*

*Dennis Sheehan*

*Greg Simsic*

*Susan VandeWalle*

*Jeff Yesh*

## **Page Layout**

*Elaine Brush*

*Mary Ann Cosby*

*Rich Evers*

*Ayanna Lacey*

*Kim Scott*

## **Proofreading**

*Don Brown*

*Mona Brown*

*Cheryl Cameron*

*Kimberly K. Hannel*

*Donna Harbin*

*Beth Rago*

*Suzanne Tully*

*Karen Walsh*

*Holly Wittenberg*

## **Indexer**

*Greg Eldred*

# **Overview**

Foreword	
Intro	
<b>Part I</b>	<b>First Things First</b>

1	Connection Requirements
2	Access at All Costs
3	Tricks of the Trade: Tools
4	Where's the Complaint Department?
<b>Part II</b>	<b>Everyday Guru Facilities</b>
5	E-mail
6	Faxing from the Internet
7	FTP
8	Telnet
9	Usenet
10	WWW
<b>Part III</b>	<b>A Guru's Diversions</b>
11	Online Entertainment
12	Listservs and Mailing Lists
13	Mbone: The Internet Multimedia Backbone
14	Community Computing
<b>Part IV</b>	<b>Finding Stuff: You Name It, a Guru Can Find it (and So Can You)</b>
15	Archie
16	Veronica
17	Discussion Forums
18	Gopher
<b>Part V</b>	<b>Tricks of the Professional Internet Gurus</b>
19	Internet in the Classroom
20	Business and Commerce
21	Research on the Internet
22	Art on the Internet
Appendix A	Uniform Resource Locators
Appendix B	Signature Files
Appendix C	Standards and Where to Find Them
Index	

# Foreword

--by James "Kibo" Parry

Yesterday I was talking to my dog, Spot.

"Hey, Spot," I said, "with the massive amount of cash this Foreword has brought me, I felt generous, and so I bought you your very own computer."

Spot looked across the room to where I'd set up the machine next to his water dish. Still shiny with that putty-gray factory finish, it was the very latest model—the IBM Dog Computer 3000 with a cordless pawboard and a squeaky mouse! A 999,000-baud modem was connected to Spot's very own phone line. "Gosh," drooled Spot, "Thank you, Kibo! It's the most wonderful thing anyone ever gave a dog like me! Now I can log on to that amazing new Information Superhighway!"

"Uh, Spot, before you do, you know how to use the Internet, right?"

Spot barked happily and ran in a little circle. "Sure I do! I've seen *The Lawnmower Man* and *WarGames* and every episode of William Shatner's awesome *Tekwar*. I know all about this stuff. First you put on your virtual reality helmet and your glowing blue neon datasuit, then you fly through the maze, and laser beams come out of your eyes and you have to shoot the computer viruses before they turn you into a skeleton and then you have twenty seconds to guess the three-letter password before all the nuclear missiles in the world come out of the screen at you!!!" Spot turned on his new computer and the screen lit up:

>

Spot cried. "Waah! This sucks! Kibo, you tricked me. I hate you!"

"Now, now, little Spot, it wasn't I who cruelly deceived you. It was the *media*. Remember, everything can be blamed on the media. Who told you it was called 'The Information Superhighway'?"

"The media," sobbed Spot.

"And who told you it was a super-cool new thing they just invented last week?"

"The media," sobbed Spot.

"And who told you it was a dangerous and scary thing that could molest your children while they sleep?"

"The media!" bawled Spot. "Oh, how I hate them! They're evil. I guess I don't know anything about what the so-called Information Superhighway is really like. Kibo, you'll have to show me what to do here."

I leaned over and typed in Spot's password, DOG\$BONE. Spot was now logged in to the world's most powerful information service, Dogidy.

WELCOME TO DOGIDY™, AN INFORMATION SERVICE WITH A DIFFERENCE.

A JOINT VENTURE OF MacBURGER AND COLA-SODA.

You have 13 new pieces of mail from long-lost friends and people who want to date you.

There are 637 news articles clipped for you, on subjects that you care deeply about, with all others omitted.

The President of the United States would like to talk to you personally via videoconferencing. He is offering you a Cabinet post.

You are our one millionth customer! Press "Y" now to receive a year's free supply of your favorite brand of dog food.

Spot burst into tears again. "Waah! I can't *read*, I'm just a *dog*!"

The "Information Superhighway," for some reason, has become the hot new topic. *Every* magazine has done at least one article on it, usually emphasizing its hip trendiness.

The Internet has been around for decades in various forms. Heck, you could even trace its precursors back to the laying of the first transcontinental phone cable. Early computer networking often involved computers making phone calls to each other. Then, someone came up with the bright idea of connecting special lines directly between the computers so that they could communicate faster, and without paying the phone company. Various computers were connected to different networks, such as ARPAnet, Bitnet, and so on. Eventually these networks were linked together in various ways, so that now, effectively, all the well-connected computers in the world are on one network—the Internet. The computers communicate with one another in various ways—via cables, via microwave dishes, via satellite, and yes, some still call each other on the phone.

The commercial "information services" are now getting in on the game as well—if you have a CompuServe account, you can send mail to someone who has an America Online account, through the magic of the Internet. Businesses are offering electronic catalogs of their goods over the network, so that you can shop with your computer. You can send letters to people ranging from Conan O'Brien to the President over the network, without having to pay for a stamp. (Whether you pay for the network is different, but there certainly are ways of getting on for free.) You can browse through massive libraries of information, and even download entire books within minutes.

Ten years ago, you could have done many of the same things. You couldn't have sent electronic mail to President Clinton (or even President Reagan), and CompuServe users were forced to chat only with other CompuServe users, but the basic capabilities for dealing with large quantities of information rapidly were there. So why is there this sudden craze for the "Information Superhighway?"

Is it because someone coined that damned term?

Is it because computers are slowly getting easier to use?

Because they're rapidly getting more powerful?

Or is it just a "hundredth-monkey" situation, where enough people have gotten "into" it so that it's now chic?

I don't know, but the number of people connected to the Internet, and the quality of their connections, and the quantity of information moving over the network, continues to increase. I've had access for only eight years, but in that time, I've seen an amazing increase in the amount of *stuff* that happens on the network. For instance, on my favorite Internet service, Usenet, I can recall when I could read most all the articles in

one day. Today, there are over 9600 *topics* on Usenet (it'll be ten thousand by the time the book is printed!), and many of them contain enough articles to keep you busy for several hours.

As the network grows and grows, the users (and their interfaces) need to grow more sophisticated just to be able to keep up with the flood of information. To be a "power user" these days, you need finely-honed skills that let you find the information or service or discussion you want, without having to wade through ten zillion irrelevant things. Just *finding* the documentation on how to do something (let alone *reading* possibly hundreds of pages of it) can be a difficult task.

Speaking of reading hundreds of pages, Spot finally managed to teach himself to read by following the self-guided tutorial in GNU EMACS. In just fifty easy lessons, Spot could read like a pro! With newfound enthusiasm, he logged into Dogidy and read the first screenful of text.

Spot was so excited by the availability of billions of bytes of information that he instantly forgot how to read. Poor Spot!

## About the Authors

**Philip Baczewski** (baczewski@unt.edu) is the Assistant Director of Academic Computing at the University of North Texas in Denton, Texas. Baczewski holds a doctoral degree in music composition, and in addition to his activities in computing support and instruction, he is a published composer of classical music and a researcher in the field of music cognition.

**Kevin Barron** is the system manager for the Institute for Theoretical Physics, at the University of California, Santa Barbara. Kevin has been roaming the Net for more than 11 years, and for the past several years he has taught "Beyond E-Mail", a seminar on navigating the Internet. He is also a contributing author to *The Internet Unleashed*.

**Billy Barron** (billy@metronet.com) is currently the network services manager for the University of Texas at Dallas and has an M.S. in Computer Science from the University of North Texas. Billy is best known for his previous work on the CICNet Electronic Journal Project and the electronic book *Accessing Online Bibliographic Databases*. His articles can regularly be found in publications such as the *Internet Society Newsletter* and *ConneXions*.

**Robert Bickford** (rab@well.sf.ca.us) is a software consultant who lives in Marin County, California, often hacking late into the night, and (usually) enjoying it immensely. His .sig file reads: "I recognize that a class of criminals and juvenile delinquents has taken to calling themselves 'hackers,' but I consider them irrelevant to the true meaning of the word—just as Mafia members call themselves 'businessmen' but nobody pays that fact any attention."

**Jon Callas** is a founder and Director of Technology of World Benders, Inc. He has nearly twenty years' experience in computing, and has been a denizen of the Internet since 1978 or 1979 (he can't remember which), when it was still the ARPAnet. He has worked on many things including satellite ground systems, operating systems, windowing systems, and cross-platform networked communications applications. He was trained as a classical musician, and became a mathematician when he learned that there is a reason for the expression "starving musician." He went into software after learning that mathematicians don't fare much better. He is also a silversmith, needing a hobby that produces things that are material.

**Earl Fogel** (earl.fogel@usask.ca) is a programmer and systems analyst at the University of Saskatchewan in Canada, where he runs a variety of Gopher and World Wide Web servers, as well as performing general UNIX support. In days gone by, he received a M.Sc. in Computer Science from the University of British Columbia, developed educational software at the Ontario Institute for Studies in Education, and developed the Hytelnet software for UNIX and VMS systems.

A writer and interactive game designer from Los Angeles, **Tod Foley** is the author of seven role-playing games and senior editor of *PIX-Elation*, a nationally-distributed magazine covering developments in the virtual reality industries. His wide body of work includes the *CyberSpace* role-playing game (Iron Crown Enterprises, 1989/1992) and live theatrical role-playing experiences such as *Ghosts in the Machine* (1992), *Mediatrix* (1994) and *World of Ideas* (1994), as well as numerous magazine articles, editorials, and essays. Tod performs his cyber-civic duty as Internet Services Manager for the IICS (the International Interactive Communications Society), and sits on the Board of Directors for VRASP (the Virtual Reality Alliance of Students and Professionals). In his dwindling free time, he is also a composer and electronic musician.

**Kenny Greenberg** (kgreenb@panix.com) is a neon artist who makes neon by hand. He owns and operates Krypton Neon in Long Island City, NY, where he has created neon sets for several Broadway productions including "The Who's Tommy," "Miss Saigon," and "Grease." His neon work also appears in the MGM feature film "Six Degrees of Separation," the neon sets for the 1994 Grammy Awards, the World Financial Center (New York), the Socrates Sculpture Park (New York), the International Design Center of New York, and at Ars Electronica in Linz, Austria. Kenny writes a monthly column, "Artist-at-Large" for *Internet World* magazine and maintains New York Art Line, the Gopher Arts area on panix.com. On the Web his home page URL is <http://www.panix.com/kgreenb/kghome.html>.

**David Gulbransen** (dgulbran@indiana.edu) has been working in computer support at Indiana University for four years. He will be graduating in December with a degree in Scenic Design and Computer Imaging Technology. He is currently employed at the ACCESS MicroCenter at Indiana University, an office that offers students, staff, faculty, and departments consulting services on both hardware and software. He has had over five years' experience on the Internet, including organizing a local Internet Service Provider.

**Joseph Janes** (janes@umich.edu) is assistant professor in the School of Information and Library Studies at the University of Michigan. His research interests include user evaluation of information and the networked information environment. He teaches, in collaboration with Lou Rosenfeld, a course on Internet Resource Discovery and Organization, as well as other courses in technology, searching for information, and statistics.

**John Katsaros** (katsaros@netcom.com) is president of Collaborative Marketing, Los Altos, CA, a Silicon-Valley-based marketing and sales consulting company that helps high-tech companies improve their sales. His latest book, *Selling High Tech*, published by Probus Publishing, is a guide for helping high-tech companies increase sales and gain market share in competitive markets.

**Dave Kinnaman** (kinnaman@world.std.com) is a systems analyst for a state education agency in Austin, Texas. His work entails labor market and economic analysis for improving coordination and planning of education and training programs. Dave has over 10 years experience in classroom and jobsite training, and feels salary and benefit levels (and working conditions) for classroom teachers are not adequate. He says unnecessary limitations too often impede Internet access in our schools. Dave asks, "Why are classroom teachers the last group of college-educated professionals in North America uniformly without a

telephone at their desks?"

**James "Kibo" Parry** is a highly prominent Usenet user; one Usenet group named for him, alt.religion.kibology, has at least 88,000 subscribers. Born in 1967, he studied computer engineering before getting a degree in professional writing. He works as a graphic designer and writer in Boston. Kibo spends his free time watching bad movies for fun.

**Max Metral** (memetral@mit.edu) is a graduate student at the MIT Media Lab. He is currently studying intelligent agents in cooperation with Apple Computer. He has also worked at General Magic, where he worked with intelligent resource discovery in large information spaces (much like the World Wide Web).

**Kevin Mullet** (kevinm@rice.edu) has spent much of the past ten years helping people use computers and networks. Currently, he works in a team that provides technical and operational support for Rice University campus network as well as Sesquinet, a Texas regional Internet provider. When he's not troubleshooting network problems, proselytizing about the Net, or drinking far too much coffee, he divides his time between writing about the Internet, finding ways to make it more accessible to regular folks, and trying to reconcile the OSI seven-layer network model with the Sanskrit chakra system. Kevin has a World-Wide Web home page at <http://is.rice.edu/~kevinm/>

**Joseph Poirier** (snag@acca.nmsu.edu) is a software engineer for Network Design Technologies, Inc., where he designs and implements object-oriented telecommunications network optimization software. He is known as Snag on several muds.

**Lou Rosenfeld** (lou@mich.edu) was formerly assistant librarian for Information Technology Development, University Library, University of Michigan, where he helped shape the Library's Gopher server and Internet training workshops. Currently, Lou is vice president of Argus Associates, Inc., a consulting firm specializing in Internet training and systems design. He also is a doctoral student and instructor at the University of Michigan's School of Information and Library Studies.

**Margaret Russell-Gonzalez** designs human-computer interface and tests the usability of software for World Benders. She began writing for Marvel Comics and has been a writer and sometime engineer for over 20 years. She notes the main difference between GUI and a comic book is that the sound effects are more fun. She considers the Internet a tool that darn well ought to be easy and meaningful to use. Ms. Russell-Gonzalez holds an M.S. from Cornell University.

**Kevin Savetz** (savetz@northcoast.com) is an author and Internet aficionado based in Humboldt County, California. He is author of *Your Internet Consultant: the FAQs of Life Online* (Indianapolis: Sams Publishing, 1994) and publishes the Internet Services FAQ and the Unofficial Internet Book List.

**Peter Scott** (scottp@herald.usask.ca) is the manager of small systems at the University of Saskatchewan Libraries. The author of Hytelnet and other hypertext software programs, he is a frequent speaker at Internet conferences, and the president of the Saskatoon Free-Net Association.

**Stephen Volan** is the editor of *The Right Foot* series of guidebooks to colleges for newcomers, published by Tall Order Press. He is also the executive director of Blue Marble Information Services, an Internet service provider in Bloomington, Indiana, where he is also the Cyberspace columnist for *The Ryder* magazine. A 1994 graduate of Indiana University, Volan got bitten by the personal-computing bug at Carnegie Mellon University in Pittsburgh in the mid-eighties. He is a long-time Macintosh user who nevertheless remembers life before graphic user interfaces, and sometimes pines for a command-line

interface.

# Introduction

--by Billy Barron

The *Internet guru* is a hard person to define. No test exists to qualify a person as an Internet guru. It is not always obvious who is and is not a guru. Often, I am talking to some novices and they mention that someone is an "Internet guru." When I meet this person, it is frequently the case that the supposed guru is giving out wrong and misleading information. Therefore, the first criteria of being an Internet guru is that other Internet gurus must be able to respect the accuracy your Internet knowledge.

If you watch Internet gurus in action, they usually show extremely strong feelings about the Internet. Almost all, if not all, find the Internet to be an exciting and fun place. This interest is critical to any Internet guru; the Internet is so large and changing so fast that without a strong passion, a guru will become a nonguru within a couple of years.

An Internet guru knows that the Internet only works due to information sharing that is mostly for free. The guru, therefore, must give something back to the Internet to improve the network. It might be a piece of software, documentation, frequently answering posts on Usenet constructively, or even just having a useful Gopher or Web server.

The guru knows that he or she does not and cannot know everything about the Internet. Therefore, this books offers a great deal, even to the guru. For example, while I was reading this book, I learned quite a bit about a subject that I knew nothing about, like programming muds. The guru, though, has many tools *and* an ability to learn that compensates for gaps in knowledge. Related to this, the Internet guru has contacts and/or friendships with other gurus that provide invaluable information.

Every guru is different in knowledge, background, career, and personality. This is important to remember. Some are not even computer professionals and instead are librarians, scientists, artists, musicians, or students. Basically, they can be from any place and any walk of life. For a brief sampling of Internet gurus, consider a few of the authors who wrote parts of this book. In many ways, I may be one of the most typical of Internet gurus. I am very strong in end-user Internet services (WWW, Gopher, OPACs, FTP sites, and so on) and can even manage routers, but I know very little about network management or muds. Another author, Kevin Mullet, is one of the people who holds Internet connectivity together within Texas—he knows network management inside and out (among other aspects of the Internet). Kenny Greenberg is also an Internet guru, even though he is first and foremost an artist. The other authors all come from different backgrounds.

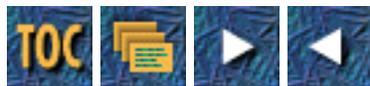
Now that you may have a vague idea of what an Internet guru is, you may want to become one yourself. Fortunately, it is much easier to define the steps in becoming a guru than defining what a guru is. The steps I recommend:

1. Become a proficient Internet user. Many books and training sessions are available to help you get to this state.
2. Learn about the society and politics of the Internet. Books, such as *The Internet Unleashed* (Indianapolis: Sams Publishing, 1994), will help with this. However, nothing substitutes for some

time spent on mailing lists and Usenet news.

3. Decide which parts of the Internet excite you. Concentrate on learning these. As I said earlier, nobody knows everything about the Internet.
4. Do some background reading. This book will definitely help you here, but reading some network documents such as RFCs is also important.
5. Try out what you have learned—as long as it is legal and ethical. No Internet guru evolves without spending a lot of hands-on time on the network itself.
6. When your experiments fail (and they sometimes will), either go back to Step 4 or ask on mailing lists or newsgroups, and then try again. The urge to give up may be strong at times, but don't give in!

*Good luck* on becoming an Internet guru. Also, make sure you *have fun* while reading *Tricks of the Internet Gurus*, or you have been missing the point.



# Tricks of the Internet Gurus

- [Tricks of the Internet Gurus](#)
- [Overview](#)
- [Foreword](#)
- [About the Authors](#)
- [Introduction](#)
- [Part I](#)
  - [First Things First](#)
- [1](#)
  - [Getting Connected: The Guru's Options](#)
- [2](#)
  - [Access at All Costs](#)
- [3](#)
  - [Tricks of the Trade: Tools](#)
- [4](#)
  - [Where's the Complaint Department? Or, What to Do When Things Don't Work](#)
- [Part II](#)
  - [Everyday Guru Facilities](#)
- [5](#)
  - [E-Mail](#)
- [6](#)
  - [Faxing from the Internet](#)
- [7](#)
  - [FTP](#)
- [8](#)
  - [Telnet](#)
- [9](#)
  - [Usenet](#)
- [10](#)
  - [World Wide Web](#)
- [Part III](#)
  - [A Guru's Diversions](#)

- [11](#)
  - [Online Entertainment for the Internet Guru](#)
- [12](#)
  - [Listservs and Mailing Lists](#)
- [13](#)
  - [MBone: The Internet's Multimedia Backbone](#)
- [14](#)
  - [Community Computing](#)
- [Part IV](#)
  - [Finding Stuff: You Name It, a Guru Can Find It \(and So Can You\)](#)
- [15](#)
  - [Life with Archie](#)
- [16](#)
  - [Veronica](#)
- [17](#)
  - [Discussion Forums](#)
- [18](#)
  - [Gopher](#)
- [Part V](#)
  - [Tricks of the Professional Internet Gurus](#)
- [19](#)
  - [Internet in The Classroom: How Educators Use the Internet](#)
- [20](#)
  - [Business and Commerce](#)
- [21](#)
  - [Finding Information You Want](#)
- [22](#)
  - [Art on the Internet](#)
- [A](#)
  - [Uniform Resource Locators \(URLs\)](#)
- [Appendix B](#)
  - [Signature Files](#)
- [C](#)

- [Standards and Where To Find Them](#)



- [Introduction](#)

- [--by Billy Barron](#)

# Introduction

**--by Billy Barron**

The *Internet guru* is a hard person to define. No test exists to qualify a person as an Internet guru. It is not always obvious who is and is not a guru. Often, I am talking to some novices and they mention that someone is an "Internet guru." When I meet this person, it is frequently the case that the supposed guru is giving out wrong and misleading information. Therefore, the first criteria of being an Internet guru is that other Internet gurus must be able to respect your Internet knowledge.

If you watch Internet gurus in action, they usually show extremely strong feelings about the Internet. Almost all, if not all, find the Internet to be an exciting and fun place. This interest is a critical part of any Internet guru. The Internet is so large and changing so fast that without a strong passion, a guru will become a nonguru within a couple of years.

An Internet guru knows that the Internet only works due to information sharing that is mostly for free. The guru, therefore, must give something back to the Internet to improve the network. It might be a piece of software, documentation, frequently answering posts on Usenet constructively, or even just having a useful Gopher or Web server.

The guru knows that he or she does not and cannot know everything about the Internet. Therefore, this book offers a great deal, even to the guru. For example, while I was reading this book, I learned quite a bit about a subject that I knew nothing about, like programming muds. The guru, though, has many tools *and* an ability to learn that compensates for gaps in knowledge. Related to this, the Internet guru has contacts and/or friendships with other gurus that provide invaluable information.

Every guru is different in knowledge, background, career, and personality. This is important to remember. Some are not even computer professionals and instead are librarians, scientists, artists, musicians, or students. Basically, they can be from any place and any walk of life. For a brief sampling of Internet gurus, consider a few of the authors who wrote parts of this book. In many ways, I may be one of the most typical of Internet gurus. I am very strong in end-user Internet services (WWW, Gopher, OPACs, FTP sites, and so on) and can even manage routers, but I know very little about network management or muds. Another one of the authors, Kevin Mullet, is one of the people who holds Internet connectivity together within Texas--he knows network management inside and out (among other aspects of the Internet). Kenny Greenberg is also an Internet guru, even though he is first and foremost an artist. The other authors all come from different backgrounds.

Now that you may have a vague idea of what an Internet guru is, you may want to become one yourself. Fortunately, it is much easier to define the steps in becoming a guru than defining what a guru is. The steps I recommend:

1. Become a proficient Internet user. Many books and training sessions are available to help you get

to this state.

2. Learn about the society and politics of the Internet. Books, such as *The Internet Unleashed* (Sams Publishing: Indianapolis, 1994), will help with this. However, nothing substitutes for some time spent on mailing lists and Usenet news.
3. Decide which parts of the Internet excite you. Concentrate on learning these. As I said earlier, nobody knows everything about the Internet.
4. Do some background reading. This book will definitely help you here, but reading some network documents such as RFCs is also important.
5. Try out what you have learned--as long as it is legal and ethical. No Internet guru exists without spending a lot of hands-on time on the network itself.
6. When your experiments fail (and they sometimes will), either go back to Step 4 or ask on mailing lists or newsgroups, and then try again. The urge to give up may be strong at times, but don't give in!

*Good Luck* on becoming an Internet guru. Also, make sure you *have fun* while reading *Tricks of the Internet Gurus*, or you have been missing the point.





○ [Foreword](#)

■ [--by James "Kibo" Parry](#)

# Foreword

**--by James "Kibo" Parry**

**Yesterday I was talking to my dog, Spot**

"Hey, Spot," I said, "with the massive amount of cash this Foreword has brought me, I felt generous, and so I bought you your very own computer."

Spot looked across the room to where I'd set up the machine next to his water dish. Still shiny with that putty-gray factory finish, it was the very latest model—the IBM Dog Computer 3000 with a cordless pawboard and a squeaky mouse! A 999,000-baud modem was connected to Spot's very own phone line. "Gosh," drooled Spot, "Thank you, Kibo! It's the most wonderful thing anyone ever gave a dog like me! Now I can log on to that amazing new Information Superhighway!"

"Uh, Spot, before you do, you know how to use the Internet, right?"

Spot barked happily and ran in a little circle. "Sure I do! I've seen *The Lawnmower Man* and *WarGames* and every episode of William Shatner's awesome *Tekwar*. I know all about this stuff. First you put on your virtual reality helmet and your glowing blue neon datasuit, then you fly through the maze, and laser beams come out of your eyes and you have to shoot the computer viruses before they turn you into a skeleton and then you have twenty seconds to guess the three-letter password before all the nuclear missiles in the world come out of the screen at you!!!" Spot turned on his new computer and the screen lit up:

>

Spot cried. "Waah! This sucks! Kibo, you tricked me. I hate you!"

"Now, now, little Spot, it wasn't I who cruelly deceived you. It was the *media*. Remember, everything can be blamed on the media. Who told you it was called "The Information Superhighway"?"

"The media," sobbed Spot.

"And who told you it was a super-cool new thing they just invented last week?"

"The media," sobbed Spot.

"And who told you it was a dangerous and scary thing that could molest your children while they sleep?"

"The media!" bawled Spot. "Oh, how I hate them! They're evil. I guess I don't know anything about what the so-called Information Superhighway is really like. Kibo, you'll have to show me what to do here."

I leaned over and typed in Spot's password, DOG\$BONE. Spot was now logged in to the world's most powerful information service, Dogidy.

WELCOME TO DOGIDY[tm], AN INFORMATION SERVICE WITH A DIFFERENCE.

A JOINT VENTURE OF McDONALDS AND COCA-COLA.

You have 13 new pieces of mail from long-lost friends and people who want to date you.

There are 637 news articles clipped for you, on subjects that you care deeply about, with all others omitted.

The President of the United States would like to talk to you personally via videoconferencing. He is offering you a Cabinet post.

You are our one millionth customer! Press "Y" now to receive a year's free supply of your favorite brand of dog food.

Spot burst into tears again. "Waah! I can't *read*, I'm just a *dog*!"

The "Information Superhighway," for some reason, has become the hot new topic. *Every* magazine has done at least one article on it, usually emphasizing its hip trendiness.

The Internet has been around for decades in various forms. Heck, you could even trace its precursors back to the laying of the first transcontinental phone cable. Early computer networking often involved computers making phone calls to each other. Then, someone came up with the bright idea of connecting special lines directly between the computers so that they could communicate faster, and without paying the phone company. Various computers were connected to different networks, such as ARPAnet, Bitnet, and so on. Eventually these networks were linked together in various ways, so that now, effectively, all the well-connected computers in the world are on one network—the Internet. The computers communicate with one another in various ways—via cables, via microwave dishes, via satellite, and yes, some still call each other on the phone.

The commercial "information services" are now getting in on the game as well—if you have a CompuServe account, you can send mail to someone who has an America Online account, through the magic of the Internet. Businesses are offering electronic catalogs of their goods over the network, so that you can shop with your computer. You can send letters to people ranging from Conan O'Brien to the President over the network, without having to pay for a stamp. (Whether you pay for the network is different, but there certainly are ways of getting on for free.) You can browse through massive libraries of information, and even download entire books within minutes.

Ten years ago, you could have done many of the same things. You couldn't have sent electronic mail to President Clinton (or even President Reagan), and CompuServe users were forced to chat only with other CompuServe users, but the basic capabilities for dealing with large quantities of information rapidly were there. So why is there this sudden craze for the "Information Superhighway?"

Is it because someone coined that damned term?

Is it because computers are slowly getting easier to use?

Because they're rapidly getting more powerful?

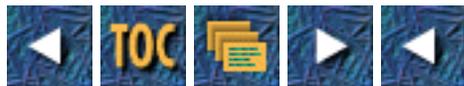
Or is it just a "hundredth-monkey" situation, where enough people have gotten "into" it so that it's now chic?

I don't know, but the number of people connected to the Internet, and the quality of their connections, and the quantity of information moving over the network, continues to increase. I've had access for only eight years, but in that time, I've seen an amazing increase in the amount of *stuff* that happens on the network. For instance, on my favorite Internet service, Usenet, I can recall when I could read most all the articles in one day. Today, there are over 9600 *topics* on Usenet (it'll be ten thousand by the time the book is printed!), and many of them contain enough articles to keep you busy for several hours.

As the network grows and grows, the users (and their interfaces) need to grow more sophisticated just to be able to keep up with the flood of information. To be a "power user" these days, you need finely-honed skills that let you find the information or service or discussion you want, without having to wade through ten zillion irrelevant things. Just *finding* the documentation on how to do something (let alone *reading* possibly hundreds of pages of it) can be a difficult task.

Speaking of reading hundreds of pages, Spot finally managed to teach himself to read by following the self-guided tutorial in GNU EMACS. In just fifty easy lessons, Spot could read like a pro! With newfound enthusiasm, he logged into Dogidy and read the first screenful of text.

Spot was so excited by the availability of billions of bytes of information that he instantly forgot how to read. Poor Spot!





*Tricks of the Internet Gurus* represents a monumental departure from ordinary Internet books. These authors are not just folks who decided to learn about the Internet to write a book; for them, it's a way of life! Here's a quick rundown of material presented by these Internet movers-and-shakers.

**Chapter 1: Connection Requirements**—Kevin Barron examines just what is involved in making a connection.

**Chapter 2: Access at All Costs**—David Gulbransen and Steve Volan cover methods for those who just can't bear to be away from the Net. Sometimes access is easy, but what do you do when it isn't?

**Chapter 3: Tricks of the Trade: Tools**—Earl Fogel presents an overview of the tools no Internet guru should be caught without.

**Chapter 4: Where's the Complaint Department?**—Kevin Mullet discusses what Internet gurus do when they have a problem. What to do when good tools go bad?

**Chapter 5: E-Mail**—Even if you don't have a full Internet connection, an e-mail gateway can provide at least a partial fix. Philip Baczewski presents his perspective.

**Chapter 6: Faxing from the Internet**—Kevin Savetz presents some of the more fascinating ways to convert all-electronic communiques to the world of the fax.

**Chapter 7: FTP**—Philip Baczewski demonstrates what FTP gurus can accomplish.

**Chapter 8: Telnet**—Peter Scott reviews what it takes to become and remain a telnet guru.

**Chapter 9: Usenet**—James "Kibo" Parry offers survival advice for all Usenet gurus and junkies.

**Chapter 10: WWW**—Billy Barron helps you ensure guru status with coverage of one of the hottest- and fastest-growing Internet tools: the World Wide Web.

**Chapter 11: Online Entertainment**—Joseph Poirer and Tod Foley step beyond using muds and MOOs and into the world of programming your own.

**Chapter 12: Listservs and Mailing Lists**—Philip Baczewski demonstrates a guru's perspective of some of the most-enduring tools on the Internet.

**Chapter 13: MBone: The Internet Multimedia Backbone**—Kevin Mullet provides understanding of this growing, audio-visual branch of the Internet medium.

**Chapter 14: Community Computing**—Kevin Mullet covers the marriage of local society and the Internet.

**Chapter 16: Veronica**—Billy Barron covers everything needed to exploit this complex search tool.

**Chapter 17: Discussion Forums**—Philip Baczewski presents the best of this communications tool.

**Chapter 18: Gopher**—Billy Barron ensures you are never at the mercy of a Gopher.

**Chapter 19: Internet in the Classroom**—Dave Kinnaman explores the best ways to make educational use of the Internet.

**Chapter 20: Business and Commerce**—John Katsaros demonstrates how to conduct business on the Internet.

**Chapter 21: Research on the Internet**—Louis Rosenfeld and Joseph Janes show just how all of that information can be channeled into useful research.

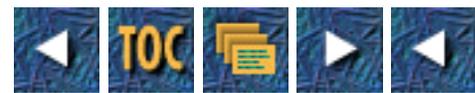
**Chapter 22: Art on the Internet**—Kenneth Greenberg presents the Internet from a unique perspective: that of the artist.

Last but not least, Max Metral adds material on URLs, .sig files, and standards that no guru should be without.

**Appendix A: Uniform Resource Locators**

**Appendix B: Signature Files**

**Appendix C: Standards and Where to Find Them**





■ [Interviews](#)

## Interviews

*Tricks of the Internet Gurus* contains a number of interviews by those who best represent what the Internet is and where it is going. Here's a quick rundown of these exciting net.personalities:

**Chapter 2:** Hyperattentive net.surfer, founder of the Aleph e-mail list, and researcher/compiler of the widely-circulated and regularly-updated net.guide *High Weirdness By Email*, Mitchell Porter has been described as "Australia's most dedicated net.activist."

**Chapter 3:** Patrick Kroupa, also known as "Lord Digital," is a widely respected hacker who first rose to prominence in the computer underground during the early eighties. Patrick eventually burnt out on the hacker lifestyle, an event that ultimately caused him to question the very meaning of his existence.

**Chapter 5:** Chief Scientist at First Virtual Holdings, Inc., Nathaniel Borenstein is a world-rekowned doctor of computer science and a veritable legend of the online world. His most famous work is undoubtably MIME, the Internet standard format for interoperable multimedia data.

**Chapter 9:** "Kibo," the net.persona of graphic artist and ontological guerilla James Parry, is a virtual god to many Usenet users, who read of his exploits (or just hang out with him) in any of the several newsgroups devoted to discussion of the man, the myth, and whatever other weird thoughts pop into their peculiar little heads.

**Chapter 12:** Paco Xander Nathan, also known as [pacoid@io.com](mailto:pacoid@io.com), is a programmer, writer, lecturer, and exemplary entrepreneur from 15 minutes into the future. He is perhaps best and most widely known as the cofounder of FringeWare Inc., a decidedly alternative hardware/software vendor and publishing company that does a majority of its business online.

**Chapter 16:** Fred Barrie and Steven Foster are the creators of Veronica, a powerful Gopher-based program that continually gathers and maintains a list of menu items on all Gopher servers worldwide.

**Chapter 17:** As Conferencing Manager for The WELL, Gail Ann Williams is known by thousands of users as <gail>. Reading and posting in dozens of topic categories daily, her job is to help The WELL's Conference "hosts" keep their users' conversations active and "on thread."

**Chapter 19:** Janet Murray is the librarian at a comprehensive public high school serving 1,500 students in Portland, Oregon, and SysOp of the HI TECH TOOLS for Librarians BBS (FidoNet 1:105/23). She is also a cofounder of K12Net.





- [Part I](#)
  - [First Things First](#)

# Part I

## First Things First





- [1](#)
  - [Getting Connected: The Guru's Options](#)
    - [—by Kevin Barron](#)
      - [Connection Categories](#)
      - [Determining Needs](#)
      - [Dialup Hardware](#)
      - [The Dialup Connection](#)
      - [SLIP/PPP: the Proxy Connection](#)
      - [UUCP](#)
      - [ISDN](#)
      - [AirNet](#)
      - [Leased Line](#)
      - [Summary](#)
      - [References and Further Reading](#)
        - [Books](#)
        - [Online Materials](#)

# 1

## Getting Connected: The Guru's Options

—by Kevin Barron

One of the most basic hurdles for the would-be Internet user is "How do I get connected?" It is also a difficult question to answer because it is not merely a matter of filling out some forms at a utility company office. Unlike other services such as phone or cable TV, there is no single provider and no unified service offerings to choose from. Granted, there are service providers who make it relatively simple to connect to the Internet; each service is different, however, offering varying types of service and pricing structures. Trying to sort out the differences among these services can make the process of connecting more confusing, instead of simplifying it as intended.

To make the process somewhat more comprehensible, this chapter takes a look at the services offered in light of how they can match your needs. The first order of business, then, is to describe the types of services available in terms of how you connect to them. The next step is to determine your needs and

match them to the corresponding connection category. In this way, it will become apparent which type of connection is best for you.

## Connection Categories

Connections to the Internet can be described in four basic categories: *dialup*, *proxy*, *direct* and *leased*.

A *dialup* connection is attained through a modem or similar device. The connection typically provides between 2400 bps (bits per second) and 28,800 bps throughput, which is usually more than adequate for e-mail and for running processes on the remote host. While this type of connection is usually adequate for line-oriented access, it is not suitable for full-screen programs. For instance, utilities such as Mosaic expect to be able to address the full screen of a machine that is directly connected to the internet. To make full use of many of the current Internet resources, then, a more advanced connection is required—at minimum, a proxy connection.

A *proxy* connection is also typically made through a modem, but it has special provisions that enable full Internet access. SLIP (Serial Line IP) and PPP (Point-to-Point Protocol) are two examples of such provisions (both discussed later in this chapter). Tools such as FTP, Gopher and Mosaic are thereby available for use just as if the connection were directly on the Internet.

*Direct* connections are obtained when the machine being used has a dedicated circuit to the Internet. This can still be a remote connection, but it has the full functionality of a directly connected machine. A good example is ISDN (Integrated Services Digital Network), which is discussed later in this chapter.

Lastly, the *leased* line is a circuit leased from the local telephone company that provides a permanent address on the Internet. The advantage of a permanent address is that it enables services such as FTP or Gopher to be set up. The leased line varies in throughput from 9600 bps to 45,000,000 bps (45 Mbps).

## Determining Needs

One of the first steps in deciding on the most appropriate type of Internet connection is to determine your needs. An individual connecting from home may be well served by a low-speed modem and a PC (I survived many years with a terminal and a 2400 bps modem). On the other hand, if you need to transfer large amounts of data, or you want a more direct connection, you will want to consider other options.

One option is to obtain a data circuit from your local telephone company; another approach is to buy the fastest modem supported by the service provider. Note that while the data circuit is likely to provide a faster connection, it generally ties you to a single service provider. On the other hand, purchasing your own modem means that you can simply dial the service provider who best meets your needs. But then, how do you determine your needs?

To help determine your needs, it may be useful to consider the following categories:

- *Type of connection* (individual, group, or enterprise)
- *Frequency of use* (occasional, frequent, or constant)
- *Throughput* (low, medium/occasionally high, or high/frequently high)

Although these categories are not necessarily mutually exclusive, and they can readily change, they are useful as a starting point. Of particular significance is whether the connection is individual or enterprise. The hardware required to make an individual connection can be extremely simple—a terminal and modem will suffice. Monthly charges, where applicable, typically run no more than the fixed monthly portion of a phone bill (approximately \$15). On the other end of the scale, an enterprise connection may require thousands of dollars worth of equipment, with monthly charges also in the thousands of dollars.

A group connection is distinct from both the individual and enterprise in that the requirements are quite different. Usually this type of connection is required when two or more individuals will be using the connection simultaneously—for example, a satellite office where five employees need to access the network on a frequent basis. If more than one person will be using the connection at a given time, a single modem may not suffice. In fact, even several modems may not be the most efficient or cost-effective solution.

Depending on the frequency of use and the amount of data being transferred, a leased line may be best. Even high speed modems can be a bottleneck when it comes to supporting data-intensive applications. Moreover, economies of scale dictate that shared use of a medium-grade connection is more cost effective than simply multiplying individual connections. If the phone line being used for the modem is being charged usage rates (as business phones are), the point at which a leased line becomes cost effective is approximately three hours per day. Generally speaking, the costs of setting up a leased line are high, but these costs are quickly amortized by the significantly lower monthly costs.

The speed of leased lines varies from 9600 bps to 45 Mbps (referred to as *T3*). Due to their cost, *T3* lines are generally used only by service providers in their backbone networks (links to other regions or service providers). Typically, an enterprise connection varies from a fractional 56 Kbps or *DS-0* (19.2 Kbps) to a *T1* (1.544 Mbps). For the sake of comparison, it is interesting to note that a standard Ethernet is nominally 10 Mbps. The best method to determine the "size of the pipe" required for your site is to talk it over with some service providers in your area. The questions you should be prepared to answer are:

1. How many users do you plan to support?
2. What kinds of services will they want to use?
3. How much Internet traffic do you anticipate?
4. How deep are your pockets? (!)



**Note:** To obtain an updated list of Internet service providers offering dialup access look in: [ftp.netcom.com:/pub/info-deli/public-access/pdial](ftp://netcom.com:/pub/info-deli/public-access/pdial) or get it from the mailserv at MIT. Send the following message:

```
send usenet/news.answers/pdial  
to mail-server@rtfm.mit.edu.
```

Usually, the right answer is to buy as big a pipe as you can afford. Then plan on doubling its size at least once. The growth of the Internet is due not only to the addition of new users, but also to the increasing use by existing users. New uses for the Internet are being discovered constantly—many of them requiring substantial network bandwidth. Probably the best example of a network-intensive application is Mosaic. Due partly to its popularity and partly to its multimedia nature, Mosaic can load a network down very quickly. The image and audio files that are downloaded through Mosaic are so large that a few users is all it takes to bring a LAN to its knees.

---



## Are You a Hacker?

[by Robert Bickford](#)

Are you a *hacker*? How would you know? If all you know about the word is what you've seen on the evening news or read in a magazine, you're probably feeling indignant at the very question! But do those magazine-selling headlines really describe what a hacker is?

Some time ago I defined a hacker as "any person who derives joy from discovering ways to circumvent limitations" (*MicroTimes*, December 1986). The definition has been widely quoted since that time, but unfortunately it has not yet made the evening news in the same way as a teenager who robs a bank with his telephone.

Does that teenaged criminal fit my definition? Possibly. Does that fact mean that all, or even most, hackers are criminals? (Does that fact make all or most hackers teenagers?) Of course not! So why is there such widespread misinformation about hackers? Very simply, it's because the criminal hackers, or *crackers*, have been making news, while the rest of us are virtually invisible. For every irresponsible fool writing a virus program, there are at least twenty software engineers earning a living ". . .discovering ways to circumvent limitations." When the much-publicized Internet worm was released by an irresponsible hacker, hundreds of other hackers applied their considerable talents to control and eradicate the problem. The brilliance and creativity brought to this task are typical of the kind of people—hackers—that my definition is meant to describe.

Working on the yearly Hackers Conferences has been a mixed experience. On the one hand, helping to bring together 200 of the most brilliant people alive today, and then interacting with them for an entire weekend, is immensely rewarding. On the other hand, trying to explain to others that the Hackers Conference is not a Gathering of Nefarious Criminals out to Wreak Havoc upon Western Civilization does get a bit wearing at times. Also, trying to convince a caller that repeatedly crashing his school district's computer from a pay phone will not, emphatically *not*, qualify him for an invitation to the conference can be a bit annoying. None of this would be a problem if we hadn't let a small minority—the *crackers*—steal the show, and become associated with the word *hacker* in the minds of the general public. The attendees at the Hackers Conferences—many of whom hold Ph.D.s and/or are Presidents or other upper management of Fortune 500 companies—are (quite understandably) very indignant at being confused with these crackers.

Taking myself as an example (no, I don't have a Ph.D.—my only degree is from the "School of Hard Knocks"—and no, I'm not working in management), when this article was first published (1989), I was writing software for a company that builds medical image-processing equipment. My code controls a product that can, and often does, improve the quality of medical care, reduce the cost, or both. When I develop a piece of software that goes around some limit, I feel very happy and can often be seen with a silly grin plastered across my face. When some ignorant reporter writes a story that equates the work I do with expensive but childish pranks committed by someone calling himself a hacker, I see red.

Are you a hacker? If you want to break rules just for the sake of breaking rules, or if you just want to hurt or "take revenge" upon somebody or some company, then forget it. But if you delight in your work, almost to the point of being a workaholic, you just might be. If finding the solution to a problem can be not just satisfying but almost an ecstatic experience, you probably are. If you sometimes take on problems just for the sake of finding the solution (and that ecstatic experience that comes with it), then you almost certainly are. Congratulations! You're in good company, with virtually every inventor whose name appears in your high school history book, and with the many thousands of brilliant people who have created the computer revolution.

What can we do about all that bad press? Meet it head on! Tell the people you work with that you're a hacker and what that means. If you know somebody whose work habits, style, or personality make them pretty clearly a hacker, tell them so and tell them what you mean by that. Show them this piece!

Meanwhile, have fun finding those solutions, circumventing those limitations, and making this a better world thereby. You are an Artist of Technology, a Rider of the Third Wave, and at least you can enjoy the ride!

Bob Bickford is a software consultant who lives in Marin County, California, often hacking late into the night, and (usually) enjoying it immensely. His .sig file reads: "I recognize that a class of criminals and juvenile delinquents has taken to calling themselves 'hackers,' but I consider them irrelevant to the true meaning of the word—just as Mafia members call themselves 'businessmen' but nobody pays that fact any attention." Bob can be reached through InterNet at rab@well.sf.ca.us.

An edited version of this article appeared in Microtimes in early 1989. Copyright (c) Robert Bickford, 1989, 1992.

---

## Dialup Hardware

So, what hardware do you need for a dialup connection? As previously mentioned, at minimum, a terminal and modem will suffice. Nominally that means a PC (including Macs), terminal-emulating software, and a modem. Depending on the state of your addiction(!), you may also need a separate phone line—lest the other members of your household turn violent towards your computer. With the advent of ISDN and other high-speed services to the home, it may eventually become unnecessary to use the phone line for data. In the meantime, it is still the most common access method.

The modem has undergone many changes since the early days of acoustic couplers (headphones for the telephone handset). The newly developed V.fast standard provides for a transmission speed of

28,800—doubling the previous speed of 14.4 Kbps. Although cynics may question the need for greater speed, it does allow for services that would otherwise be unusable (Mosaic and PPP, for example). With V.fast as an established standard, any modem supporting the standard can connect to any other V.fast modem at the higher speed—there is no longer a need to rely on proprietary standards of a particular modem vendor to exceed the 14,400 standard.

Another benefit of the established V.fast protocol is that it drives down the price of those modems that don't support it. The older protocols supporting 9600 or 14,400 are often provided and are more than adequate for many users' needs. As with other aspects of computers and networking, the rule of thumb is to buy the best you can afford (and then avoid reading the sale items in the computer ads!). With fancier modems becoming available, the simpler modems become less expensive. It pays then, to know what all the buzzers and bells are for, and whether you actually need them.

Speed is not the only consideration in buying a modem. It is essential to ensure that the modem you purchase supports the error-correction and compression protocols used by the service provider you plan to use. These protocols (see Table 1.1) are standard features on newer modems, but you should be aware of some inconsistencies. MNP5 and PEP/Turbo PEP, for example, are proprietary—not all modems support them. The best policy is to check with the service provider to see which protocols are supported or required.

The various standards applied to modems have been called V-dots due to their CCIT naming convention (V.32, V.42, and so on). V-dots, along with their MNP counterparts, specify the speed, compression, and error-checking standards (see Table 1.1).

**Table 1.1. Transmission protocols.**

<i>Protocol</i>	<i>Specification</i>
V.22 (or Bell 212A)	1200 bps
V.22bis	2400 bps
V.27ter	4800 bps
V.32	9600 bps
V.32bis	14400 bps
V.fast	28800 bps
V.42	LAP-M error control
V.42bis	Data compression

As modems increase in speed, they approach the theoretical limit of what can be accomplished over voice-grade phone lines. Even with clever compression protocols, any modem will have to retransmit when line noise is encountered. With sufficient line noise, the modems are effectively forced back to a slower speed. The connection is dropped altogether, occasionally, if the modems sense that the carrier signal has been lost.



**Note:** A loss-of-carrier signal is particularly problematic when using a cellular phone to connect via modem. One of the main problems is that cellular technology incorporates in-band signaling. This signaling is inaudible because the cellphone is designed to mask it. Unfortunately for the modem, this masking also makes the carrier signal inaudible, so the net result is that the connection is lost. To get around this problem, it is necessary to do some bit-twiddling on the modems. If your modem has the capability, you should increase the detection time for carrier loss. The down side of increasing that setting is a slight delay in the time it takes to disconnect after you log out.

---

To configure a modem for use, simply follow the manufacturer's guidelines. The most important consideration is to ensure that the interface speed on the modem matches the speed of the serial port you connect it to. If your modem is set at 14,400, while your serial port is set to 9600, you will have extreme difficulty communicating with your modem!

When you are sure you can communicate with your modem, you can go ahead with configuring your terminal-emulating software. Once again, it is important that the settings are matched; in this case, you need to match the serial port and the terminal-emulating software. Typically, the settings you want to use are *8 data bits, No parity and 1 stop bit (8,N,1)*.

---



**Note:** For great guides on setting up a Telebit modem on Sun workstations, DOS, Apple, VMS, and more, FTP the appropriate guide from: [ftp.telebit.com:/pub/tech-support/modems/setup-guides](ftp://ftp.telebit.com:/pub/tech-support/modems/setup-guides).

---

## The Dialup Connection

When you get your modem all wired up and ready to go, you must choose from the many online services available. Some of the more popular services include CompuServe, America Online (AOL), and The WELL. Chapter 2 deals with online services in depth, but for this discussion, these services will suffice as examples of the dialup connection. Other examples of dialup connections include colleges that have dialup modem pools, local BBSs, and businesses that have dialup access for employees. Essentially, any machine that is on the Internet and has a modem attached can potentially serve as a dialup connection.

As an alternative to the direct connection, the dialup option offers some distinct advantages. Starting with a local BBS and continuing through the large commercial services such as CompuServe and America Online, the dialup connection provides a simple means of accessing the Internet. All that is required is a PC and a modem. Access is made through a local phone call, or sometimes an 800 number.

---



**Tip:** To find numbers for local BBSs, look in the local computer weekly or the computer supplement of larger papers. Often, there will be listings of BBSs that have Internet connectivity. Alternately, local computer stores are good sources of information—some even have their own BBS established.

---

Another advantage of the dialup connection is that so many of the complexities of Internet services are

dealt with by the service provider. The Domain Name Service (DNS), sendmail configuration, and all the other headaches faced by network administrators are taken care of for you. All you have to be concerned with is your modem, PC, and terminal-emulating software.

In the same vein, there is usually a certain amount of disk space that you can use without having to worry about backups. At least, someone is supposed to be backing up your disk space—right? At any rate, having the extra disk space is always a nice benefit.

The main drawback to using a dialup connection is that it usually does not allow the full range of Internet services. Some systems have only e-mail access to the Internet (although it's amazing how much you can accomplish using just e-mail access). And even among those services that do offer full access (FTP, Gopher, and so on), there is still a limitation in that you are connecting through a third party. For example, if you transfer files using FTP, you would have to transfer them first to the service provider's machine, then transfer the files down to your PC. Worse yet, certain applications, such as Mosaic, expect your machine to be directly connected and will not work at all.

Having said that, though, I should point out that the dialup option is often one of the most cost-effective and sensible solutions. One of the most common examples of the dialup option is connecting via your employer's or nearby college's modem pool. Depending on the policies at the college, a *community account* can be obtained for around \$15/month. Also, it is becoming more common to be able to use SLIP or PPP on these connections, thus effectively simulating a direct connection. If SLIP or PPP is available, it is well worth considering this connection option.

## SLIP/PPP: the Proxy Connection

In the dark ages of computer networking (not long after ISDN was coined), network hackers were trying to come up with a method of extending their networks using modems. The net result was *Serial Line over IP*, or *SLIP*. The specification for SLIP—RFC 1055—describes it as a nonstandard. Nonetheless, SLIP is an extremely simple and effective scheme for transmitting Internet packets over a serial line, such as a modem. As a result of its straightforward implementation and use, SLIP gained significant acceptance. There are many public-domain versions that work in conjunction with MAC, DOS, and various UNIX flavors.

The main advantage of a SLIP connection is that the remote machine (for example, the PC at home) becomes a full member of the Internet community for the duration of the call. This means that the full suite of Internet tools—Telnet, FTP, Gopher, Mosaic, and the like—can all be run as if the machine were directly connected to the network. Of course, the speed of the modem and the load on the machine hosting the SLIP connection both effect a somewhat delayed response time, but the connection evokes a sense of being directly attached to the local network.

The downside of SLIP is that because of its nonstandard nature, there are many different implementations and no clear method to conform the variations. Moreover, the simplistic nature of SLIP does not provide for many of the complexities of contemporary network environments. SLIP supports only IP, which leaves out OSI, DECnet, Appletalk, Novell Netware, and others. Moreover, SLIP requires that both ends of the connection be completely preconfigured, making nomadic computing very difficult to support. Lastly, SLIP provides no mechanism for authenticating the connection.

The solution to these problems was the development of *PPP* (*Point-to-Point Protocol*) by the Internet

Engineering Task Force (IETF). PPP represents the culmination of several years of work by the IETF's Point-to-Point Working Group. The protocol, as defined in RFC 1548, is much more full-featured than SLIP.

Some of the advantages of PPP include the capability to negotiate compression technique and maximum frame size; the provision for control characters to be "escaped" so that they can be passed through to the application; and the capability to provide low-level authentication, quality monitoring, and loopback detection. Fortunately, these additional features do not come at the cost of making a connection more difficult to establish. As the connection is being made, PPP can negotiate all the variables that might be problematic.

One of the features of PPP that is bound to make it increasingly popular is the support for nomadic computing. This feature provides the capability to make connections from any point-of-presence on the Internet, regardless of network addressing.

When the connection is made, the PPP address negotiation permits the assignment of an IP address from a pool of addresses. The nomadic computer can then use that address for the duration of the session. After the session has ended, the address is deallocated and can be reused for another connection. This reallocation of addresses will be particularly appreciated as network administrators begin to run out of assignable addresses.

Although there are many public-domain versions of SLIP and PPP available, installation is not a trivial task. The support provided by software vendors is often worth the \$100[nd]\$700 for the commercial package. Arguably, there is also added reliability in the commercial implementation—Table 1.2 contains examples of commercial SLIP/PPP software.

**Table 1.2. SLIP/PPP software providers.**

<i>OS</i>	<i>Software</i>	<i>Vendor</i>
DOS/Windows	Chameleon	Netmanage
Mac	MacSLIP	Trisoft
UNIX	MST-PPP	Morning Star

It is also possible to implement SLIP/PPP on a router. Several vendors have incorporated SLIP/PPP, a router, and a modem in one package. Examples include the NetBlazer from Telebit, and the NetHopper from Rockwell International. MorningStar also has a router with SLIP/PPP incorporated. Even though it is convenient to have a box with all this functionality packaged together, it is nonetheless an expensive approach when compared with installing the software and modem separately—assuming that you already have a machine to install them on.

On the whole, PPP provides a more robust and configurable connection than SLIP. Some implementations of PPP (Morning Star's, for example) actually include a SLIP option to interoperate with older, SLIP-only configurations. In sum, new services being established for dialup networking should be implemented using PPP. Even existing services running SLIP should be given serious consideration as prospects for converting to PPP.

## UUCP

Although it is not strictly a part of the Internet protocols, UUCP bears some discussion because it has been a significant adjunct to the Internet. Probably best known as the transport mechanism for news, UUCP is also used to provide e-mail to remote machines. Because most implementations of UNIX include UUCP, and it requires only a phone line and modem to use, UUCP is very economical. The catch is that the setup procedures are very arcane, and it does not provide full connectivity.

---



**Note:** News (discussion groups) is carried out over Usenet, which is a pseudo-network of all computers that carry news. UUCP used to be the main transport mechanism for news; hence, it was often confused with Usenet. These days, however, news is now more often distributed over TCP/IP using the Internet, so it is more evident that UUCP and Usenet are not synonymous.

For more on this topic, see Chapter 9, "Usenet."

---

Primarily, UUCP is useful for exchanging files with a remote host—particularly e-mail. In a situation wherein a remote machine is self-sufficient and only requires an exchange of e-mail, UUCP is a good choice. Such cases are becoming increasingly rare, however. More often, it is the case that remote machines are all-the-more isolated if they can't connect to the Internet; yet it is inefficient for the users of a remote machine to have to manually connect to retrieve e-mail.

One solution is to use both UUCP and TCP/IP: UUCP-over-TCP Daemon (UUCPD). If you have multiple users on a machine that is only occasionally connected to the Internet, UUCPD is an efficient means of dealing with e-mail. This is particularly true if the link that is being used to connect incurs a usage charge. If there is no usage charge, but the link is slow, it might still make more sense to implement UUCPD rather than relying on interactive for e-mail. UUCPD relies on TCP/IP, so there must be a SLIP or PPP connection made to support it; otherwise, UUCPD is configured identically to the standard UUCP.

Eventually, we are supposed to get high-speed digital connections to the home—our on-ramps to the superhighway. But don't hold your breath waiting for your neighborhood to be wired; you might be turning bluer than a Smurf before you see that kind of connectivity. In the meantime, the modem will continue to be the principal method of connecting from the home and similar single-user environments. Having said that, though, I should point out that ISDN seems to be nearing deployment in many metropolitan areas.

## ISDN

ISDN, which stands for Integrated Services Digital Network, has been called variously, "I Still Don't kNow", and "It Still Doesn't Network" due to its long-awaited implementation. Amazingly, ISDN dates back to 1968, when it was conceived as a means of integrating voice, teletext, videotex, and data. Obviously, the network landscape was quite different in those early days; consequently, ISDN was designed based on the assumption that 64 Kbps was plenty of bandwidth. Moreover, it was assumed that the same equipment used to route telephone traffic would be the best method to handle data.

Despite its shortcomings, however, ISDN is quite likely to serve as the high-speed connection to the home for the next several years. The main reason is that it is extremely expensive to rewire existing neighborhoods with fiber optic cable. ISDN works on existing copper cables and is the telephone companies' (telcos') preferred approach to bringing data communications into the home. Ironically, one of the main stumbling blocks to ISDN's implementation has been the telco's insistence on a metered fee structure (much like the way business phones are charged). Recently, much pressure has been placed on the telcos to use a flat fee for ISDN use. The flat fee would mean that users would be charged a single monthly fee only, instead of having to worry about the length of each call. If the flat rate is implemented, acceptance of ISDN would almost certainly be guaranteed, and widespread deployment would follow. On the other hand, if a split fee is used—meaning that during business hours the metered rate applies and during off-business hours the flat rate applies—chances are it will not be well received.

A service similar to ISDN that already exists in most areas is Switched 56k (SW56k). This service enables you to connect to other sites running the same service. It also interoperates with ISDN. SW56k requires special cabling (whereas ISDN does not), but is analogous to regular phone circuits in that it takes some time to make the connection (ISDN is much faster).

The biggest drawback to SW56k is that it is a metered service, like a business phone: each call is billed depending on the length of the call and more if it is long distance. The alternative to SW56k is a dedicated 56k (DDS) connection, which costs more initially, but is billed at a flat rate. The dedicated line is fixed between two points and is therefore not as flexible as SW56k.

Some telcos are offering single-line ISDN service, which is a 56 Kbps line, whereas others add the two 64 Kbps channels for a maximum throughput of 128 Kbps. If you are considering ISDN service, your best bet is to contact your local telephone company to see what services are available. General information can be found at the [info.bellcore.com](http://info.bellcore.com) FTP site. Also, regional telcos sometimes have information services available online—for example, Pacific Bell has a Gopher server at [gopher.pacbell.com](http://gopher.pacbell.com).

## AirNet

An alternative option to connect a remote site to the Internet is what I call AirNet. A number of manufacturers have developed products to connect remote computers over several miles. Using spread-spectrum technology, these devices usually broadcast at approximately 900 Mghz. A receiver unit is required at the Internet site, but because at least one manufacturer has developed a PC version (NCR's WaveLAN), the total cost is not prohibitive.



**Note:** As with any broadcast RF signal, obtaining good signal strength is based on a number of factors. Because the FCC regulates the power of the transmitters, the main variables that can improve signal strength are the antenna and its cabling. A low-loss cable and, particularly, a high-gain antenna will go a long way to aiding a weak signal. Larsen Electronics in Washington state make a high-gain (12-dB) yaggi antenna that is well-suited to this purpose.

---

The principal limitation with the AirNet connection is that the units cannot be more than a few miles apart (usually 2-3) and they must have line-of-sight visibility to establish a link. Ultimately, their

application is limited. With a bandwidth of 1-8 Mbps, however, the AirNet approach clearly surpasses the ISDN throughput. As a result, AirNet may come to serve a niche market.

## Leased Line

For the enterprise connection, you should consider nothing short of a leased line. Too often the needs of the enterprise are underestimated, resulting in costly equipment being replaced to provide a connection with higher throughput. As stated earlier, allow for double your current need and count on doubling that amount at least once.

Leased lines come in a variety of sizes and flavors. The size (or speed) varies from 9600 bps to 45 Mbps. At the bottom of the scale is the 9600 bps connection over an analog line. At the other end is the 45 Mbps (T3) connection over a digital line. T3 lines are typically used for backbone links between major networks. A good example is the NSFnet, which employs T3 lines for its national backbone network. Most enterprise connections can be adequately served by much less expensive circuits. The NSFnet itself began with 56 Kbps (DS-0), which is a data circuit over ordinary voice lines.

The next step up is a fractional T1. A T1 circuit is 1.544 Mbps, but often telcos provide fractional T1s at reduced rates. These usually start at 64 Kbps and increment by 64 Kbps, up to 1.544 Mbps.

The hardware required for a leased line varies depending on the type of circuit. Generally speaking, though, *each* end requires a CSU/DSU (a device for digital conversion) and a router. The exception is analog lines that can be used with modems.

The router is what determines which traffic is destined for the other side of the link versus the local network. The router can be a process running on a UNIX machine, but for faster circuits, it is more efficient to have a dedicated router (Cisco, Proteon, Wellfleet, and so on). This is one of those factors that can contribute to getting the maximum throughput from your Internet link. If you use a UNIX machine as a router and the machine gets loaded down with other processes, the routing performance is going to suffer. It is advisable to use a dedicated router, or at least a UNIX router on a machine that is stripped of the other processes.

## Summary

In determining the needs for an Internet connection, it becomes possible to match an appropriate type of circuit. An individual who wants to be able to run the full suite of Internet tools would probably be well served by a PPP connection running over a high-speed modem. At the same time, a satellite office with half a dozen employees who run data-intensive applications over the network would probably not be very happy with PPP over a modem. A better match for such a group connection would be ISDN or fractional T1.

At the enterprise level, it is essential to plan for growth. Although an ISDN circuit may be sufficient for the initial phase of implementation, bear in mind that it does not scale. An IS circuit is 64 Kbps, whereas a full ISDN circuit tops out at 128 Kbps. A more appropriate solution would probably be a fractional T1 because there is more room for growth—from 64 Kbps all the way to 1.544 Mbps.

The types of connections to the Internet vary greatly, and new methods continue to evolve. Determining which method is most appropriate can be complicated and frustrating. By becoming aware of your needs

and some of the connection options, however, it is possible to make a rational decision. With some luck, the decision will prove to be the right one and provide for the growth of Internet use you are bound to witness.

## References and Further Reading

### Books

Bang, Steve, et al, *The Internet Unleashed* (Indianapolis: Sams Publishing, 1994).

Estrada, Susan, *Connecting to the Internet* (Sebastopol, CA: O'Reilly & Associates, 1993).

Kessler, Gary, *ISDN: Concepts, Facilities, and Services*, second edition (New York: McGraw-Hill, 1993).

Lynch, Daniel C., and Rose, Marshall T. eds., *The Internet System Handbook* (Reading, MA: Addison-Wesley, 1993).

Quarterman, John, and Smoot, Carl-Mitchell, *The Internet Connection* (Reading, MA: Addison-Wesley, 1993).

### Online Materials

FYI 16: Connecting to the Internet, Gopher:  
[is.internic.net:/infoguide/getting-connected/about-internet-access/fyi16](http://is.internic.net:/infoguide/getting-connected/about-internet-access/fyi16)

ISDN FAQ: send e-mail to [mail-server@rtfm.mit.edu](mailto:mail-server@rtfm.mit.edu) with the message:  
send usenet/news.answers/isdn-faq

General Internet Information, Gopher:  
[is.internic.net:/infoguide](http://is.internic.net:/infoguide)

National ISDN, FTP: [info.bellcore.com:/pub/ISDN](http://info.bellcore.com:/pub/ISDN)

PacBell ISDN, Gopher: [gopher.pacbell.com:1/pacbell\\_bbs/ISDN](http://gopher.pacbell.com:1/pacbell_bbs/ISDN). FTP:  
[ftp.pacbell.com:/PacBell/info/ISDN](http://ftp.pacbell.com:/PacBell/info/ISDN).

PDIAL (Public Dialup Internet Access List); send e-mail to [mail-server@rtfm.mit.edu](mailto:mail-server@rtfm.mit.edu) with the message:  
send usenet/news.answers/pdial

PPP FAQ; send mail to [mail-server@rtfm.mit.edu](mailto:mail-server@rtfm.mit.edu) with the message:

send usenet/news.answers/ppp-faq/part1  
send usenet/news.answers/ppp-faq/part2  
send usenet/news.answers/ppp-faq/part3

...  
send usenet/news.answers/ppp-faq/part8

RFC1055 SLIP, RFC1548 PPP; send e-mail to [rfc-info@isi.edu](mailto:rfc-info@isi.edu) with the message:

Retrieve: RFC

Doc-ID: RFC1055

Retrieve: RFC

Doc-ID: RFC1548

Sutterfield, Robert A., "Low-Cost IP Connectivity," paper presented to Sun User Group, December 1991. Deals with SLIP versus PPP. Available by FTP from:  
<ftp.uu.net:vendor/MorningStar/papers/sug91-cheapIP.ps.Z>

US Dial-Up Service Providers, Gopher:

<is.internic.net:/infoguide/getting-connected/us-dialup>





- [2](#)
  - [Access at All Costs](#)
    - [--by David Gulbransen and Stephen G. Volan](#)
    - [Method #1: BBS Access](#)
    - [FreeNets](#)
    - [Online Service Providers](#)
      - [MCI Mail \(1-800-444-6245\)](#)
      - [GEnie \(1-800-638-9636\)](#)
      - [Prodigy \(1-800-776-3449\)](#)
      - [CompuServe \(1-800-848-8990\)](#)
      - [America Online \(Info: 1-800-227-6364\)](#)
    - [Internet Service Providers](#)
      - [Delphi \(1-800-695-4005\)](#)
      - [Netcom \(1-800-501-8649\)](#)
      - [Performance Systems International \(1-800-82-PSI-82\)](#)
    - [Access Through Higher Education](#)
    - [Mitchell Porter](#)
      - [--by Tod Foley](#)
    - [Employer-Provided Access](#)
    - [Cooperative Access](#)
    - [Alternative Nets](#)
    - [Summary](#)

## 2

# Access at All Costs

[--by David Gulbransen and Stephen G. Volan](#)

For most of its existence, the Internet has been considered a free resource by its users. People got used to the idea that they could log on anytime they wanted and draw from resources on the Net at will. The

costs of operating computers on the Internet may have been invisible to most users, but the Internet has never been *free*.

In its early days (see note), the Internet was paid for by the National Science Foundation (NSF). When the NSF withdrew its funding, the Internet was paid for by the universities and research facilities whose students and researchers used it. Now that you can access the Internet without being affiliated with some university or Department of Defense project, you are expected to help pay for using it as well.



**Note:** Some might quibble with the Internet's NSF origins, but it all depends on how far back you want to go. Yes, the original "net" was a Department of Defense Advanced Research Project Agency venture, but the TCP/IP internetwork that really blossomed into what we know as the Internet consists mostly of what was the NSF backbone.

---

There are still free ways to gain access to the Internet, but they're not without their problems. Free access isn't always reliable. Some methods of free access are not particularly ethical. You can try to rely on free access to the Internet, but if you want a dedicated connection, through which you can reliably receive e-mail or obtain information, paying for the service is the best way to ensure your connection will be there when you need it.

In this chapter, we discuss the many types of services that provide full or partial Internet access. To understand the differences between services, keep in mind that it's easy to confuse a *method* of communication with the *service* that employs that method to communicate. Various online services use one or more protocols (methods) for exchanging data. All these protocols and services have an exasperating tendency to use the word *net* in their names, rendering them even less distinguishable. Pay close attention.

## Method #1: BBS Access

Bulletin board systems have become an established, entry-level way for home computer enthusiasts to communicate with each other and exchange information on a local level. A bulletin board system, often referred to by the initials *BBS*, is often run by computer hobbyists because they enjoy discussing their areas of interest with other computer users. Most BBSs don't make a profit; in fact, most are paid for out of the pocket of the system operator. However, as the home computer market has expanded, and more and more people have jumped on the e-mail/Internet bandwagon, sysops have discovered that their BBSs don't have to be so limited.

Almost all BBSs have some sort of internal method for sending e-mail among its own users. Depending on which software a bulletin board is using for its interface, you may also be able to send e-mail messages to users at other BBSs in the area. With the popularity and growth of BBSs in the mid-to-late eighties, it was inevitable that some standards of communication between BBS systems would emerge. The most well-known protocol used by BBSs is FidoNet.

FidoNet software runs on DOS machines, which makes it popular in remote areas of the world as well as in North America. Each BBS operator using FidoNet exchanges mail with other BBSs in the middle of the night, during a period called "National Mail Hour." Each sysop is free to operate their FidoNet BBS

as they see fit, as long as they adhere to a few rules, one of which is that they not refuse to pass on FidoNet mail.

Someone out there with a FidoNet BBS linked it to an Internet machine and thereby created a gateway to the Internet, making it possible to send mail between the two networks. Depending on the services and fees of the BBS, and the whim of the sysop, you might be able to send and receive e-mail for a pittance or no charge at all.

You should be aware that using a FidoNet site to mail to the Internet has its drawbacks. First, FidoNet sites don't allow large volumes of traffic, and they monitor incoming traffic, both to prevent users from posting mass mailings to the Internet or from subscribing to Internet mailing lists. Both privileges would generate quite a burden on the FidoNet system, more than most sysops care to incur. Furthermore, mail can take a while to reach its destination. FidoNet sites usually exchange batches of data once a day; some FidoNet boards also pool their mail, to retrieve it in very large batch files. Your mail might take anywhere from one day to two weeks to deliver, depending on how far it's going. BBS mail might not save you time or convenience.

Many BBSs are beginning to connect to the Internet directly. These boards often obtain their connections from a larger organization via the UNIX-to-UNIX Copy Program (UUCP) protocol. Though the name implies a BBS using UUCP runs UNIX, this isn't always the case. However, UUCP connections usually offer other Internet services as well, because more than just mail can be exchanged using UUCP. For example, Usenet (home of the Internet's famed discussion and newsgroups) originated as a UUCP connection between two universities. A number of BBS operators have picked up Usenet newsgroups from larger organizations with a fuller Internet feed. Ask the sysops of your nearest BBSs if they carry newsgroups.

How do you go about finding a good local BBS? The best way is to talk to other computer enthusiasts in your area. Local computer dealers or computer special-interest groups are two helpful resources for finding your first few boards. Another resource for BBS lists is the magazine, *Computer Shopper*, which publishes a list of BBSs each month in order by state and area code. (The list is so big now that they split it and run half every other month.) When you find your first BBSs and begin talking with other users and sysops, you'll find that word-of-mouth is the best resource for finding new BBSs—after all, who better to fill you in on the latest developments than the people who run and use them?

## FreeNets

This idea is not complex: a FreeNet provides, as a public service to its users, free access to e-mail and, in some places, to Internet connectivity. It is important to remember, however, that FreeNets were not born to provide Internet access. FreeNets are primarily BBSs that exist to encourage community discourse. Today, over 30 US cities have FreeNets, there is a significant number of FreeNets in Canada, and many others are cropping up all over the world.

Here's a listing of some of the more established FreeNets:

**Table 2.1. FreeNets in the United States.**

<i>Name</i>	<i>City</i>	<i>State</i>	<i>Contact</i>

AlohaFreeNet	Honolulu	Hawaii	mathews@gold.chem.hawaii.edu
AzTeC Computing	Tempe	Arizona	askins@asu.edu
Baton Rouge FreeNet	Baton Rouge	Louisiana	73052.3620@compuserve.com
Charlotte's Web	Charlotte	N. Carolina	shsnow@vnet.net
Cleveland FreeNet	Cleveland	Ohio	(216)368-3888
CPBIFreeNet	Hartford	Connecticut	steela@csusys.ctstateu.edu
Detroit FreeNet	Detroit	Michigan	raine@gdls.com
Heartland FreeNet	Peoria	Illinois	(309) 674-1100
Houston FreeNet	Houston	Texas	joe_abernathy@pcworld.com
KC FreeNet	Kansas City	Kansas	josbourn@tyrell.net
LA FreeNet	Los Angeles	California	ablumin@eis.calstate.edu
Maine FreeNet	Freeport	Maine	efrey@mmp.org
MiamiFreeNet	Miami	Florida	currye@mail.seflin.lib.fl.us
Mobile FreeNet	Mobile	Alabama	geoffp@netcom.com
New Orleans FreeNet	New Orleans	Louisiana	nrrmc@uno.edu
North Texas FreeNet	Dallas	Texas	kenlc@tenet.edu
Pittsburgh FreeNet	Pittsburgh	Pennsylvania	iddings@clp2.clpgh.org
Philadelphia FreeNet	Philaelphia	Pennsylvania	torgj@delphi.com
Richmond FreeNet	Richmond	Virginia	kguyre@vcu.edu
Santa Fe Meta Verse	Santa Fe	New Mexico	grizz@lanl.gov
Twin Cities FreeNet	Minneapolis	Minnesota	chie@stolaf.edu
404 FreeNet	Atlanta	Georgia	mike_bernath@solinet.net

This list isn't meant to be comprehensive by any means, so if you don't see your city listed here, don't give up hope. For a complete and up-to-date list of FreeNets, you can obtain one via FTP at:

nptn.org

(login: anonymous)

(password: *your\_email\_address*)

Path: /pub/nptn/nptn.info

File: nptn.affil-organ.list

The concept of the FreeNet was born in Cleveland in 1986. The Cleveland FreeNet, sponsored by Case Western Reserve University, was the effort of Dr. Thomas Grundner (tmg@nptn.org), also the founder of the National Public Telecomputing Network. The NPTN (also in Cleveland) exists to help aid FreeNet projects and public telecomputing in general. Their FTP site contains many documents which might be helpful if you were interested in trying to start a FreeNet in your own area.

FreeNet access is not necessarily a bargain. The first risk of FreeNets is being involved with municipal governments and worrying about the legalities of free speech. Users of the Internet transmit a lot of information that might be offensive to some community groups or individuals. If, however, administrators start to censor what comes through their sites, they risk the chance of losing their right to claim common carrier status. They would then become responsible for all the information that might come through their sites. You should be prepared for many legal grey areas when operating any sort of BBS.

The other problem with FreeNets is the limitation of resources. A good ratio of users to phone lines is about 15:1, at which users are assured of obtaining access without delays that are too long, or without being forced to log on at inconvenient times such as the middle of the night. The problem with providing such a low user/modem ratio in large areas is cost. Because the very idea of a FreeNet is not to charge for services, expenses for phone lines, modems, and so forth are usually on a tight budget set by the municipal government. Therefore, FreeNets often have problems with too many users wanting to use the system.

The solution to this problem is to make the board very attractive to community users, and not Internet citizens. Most FreeNets have many local discussion groups but limit the amount of actual Internet access. For example, some FreeNets will allow you to read Usenet newsgroups, but not to post to them. Most all will allow Internet e-mail, but they frown on mailing lists. Virtually no FreeNet allows telnet access to other sites, or access to interactive services, such as Internet Relay Chat (IRC). Some even put restrictions on file transfers (FTP).

When you are considering FreeNet access, you should keep these types of restrictions in mind. The local discussion groups and the limited access most FreeNets provide might be enough to satisfy your communication needs; but, then again, they might not.

Some FreeNets will provide access to people who can telnet into their BBS. Because FreeNets are generally funded by the local community, you will find that local residents take priority. If a FreeNet does offer remote access, you are still faced with the problem of paying long-distance rates to dial in, or finding telnet access from somewhere locally—which defeats the purpose of looking far away for an account. On top of that, you are still going to have to pay for your nonlocal access. The charges average between \$10-15 a month for nonlocal access, so unless your municipality has a FreeNet, don't expect it to be free for you.

If a FreeNet does exist in your area, chances are very good that they will have some tie to the library system. Local BBSs are also a good source of information about services like FreeNets. If there is no FreeNet in your area, starting your own is one way to get access to the Internet, but it's also a lot of work. Raising funds, obtaining the equipment, and finding volunteers to help run the FreeNet might take up more time than you'd like to give up.

## Online Service Providers

Online service providers are like giant BBSs, only the proprietors are large corporations, not basement hobbyists. You log into their private network of machines and then have access to databases, mail, and discussion groups that they own, maintain, or vend (many providers act as a vendor for the databases of other companies).

Online services have traditionally been closed systems. If you had an account on CompuServe, for example, you could only communicate with other users on CompuServe, and the discussion groups were also limited to other CompuServe subscribers. Additional resources would only become available if the service provider willed it so. In the wake of the loosening of regulations against commercial traffic on the Internet, users are clamoring for access to everyone else in the world, so the large online service providers are adding Internet services to their lists of features.

Internet access is still additional, however. Most services usually start an extra meter running when you use their Internet features. Additionally, because the Internet is not the primary function of these services, chances are you won't be getting full Internet access. For example, most of the online services maintain deposits of shareware and other programs that users must pay to download. Because the major online services can't make a profit if users can get the same shareware for free somewhere else, most online services restrict access to standard Internet features such as FTP. Experienced users will find the resources of the Internet itself much more extensive—and probably much cheaper.

Hidden charges are also a problem with online services. Sure, they might provide Internet mail, but some services charge per message, some charge per *character*. The time of day you call might also result in an additional charge; usually a premium rate is charged during business hours. You should be sure to carefully read all the agreements that scroll by your screen the first time you dial in to an online service. The advertisements that they place in magazines won't mention all these surcharges, but the usage agreements will have to.

Online services are suffering from growing pains as well. The number of users flocking to the better services is increasing quite rapidly, and how that affects your usage depends on how well the company manages growth. The classic example of this pitfall is America Online, which became so popular that one day in 1994 its phone systems were unable to handle the crush of users vying for logon access, and the service ground to a halt for hours.

The good thing about all the interest in online services is that competition has created a way for you to try systems before you commit to using them. One way to maintain free access for quite some time is to switch from company to company, using the trial periods and never paying for service. Obviously, it makes e-mail correspondence unreliable. Many of the online services will also assume that you want to sign up as a customer unless you cancel. If you do take advantage of the trial periods, make sure to cancel your account before they bill you.

Nearly all online services require you to use special software to access their systems. Most have developed graphical user interfaces (GUIs) of some sort, designed to make it easier for you to find resources on their systems. How helpful these systems are varies: some are organized quite well, others are a lot of flash. Many of the services provide you with this software free because it only works with their service.

The following sections provide a breakdown of some of the more popular services, and what they have to offer.

### **MCI Mail (1-800-444-6245)**

MCI Mail is just what it says—mail, and nothing else. If you need e-mail access, it might be for you, but it's costly: \$35 a year, plus \$0.50 for the first 500 characters, \$0.10 for the next 1000, up to 10,000.

## **GEnie (1-800-638-9636)**

GEnie, a subsidiary of General Electric, has a reputation for very slow e-mail (sometimes up to several weeks for delivery), and many surcharges. The basic rate is \$8.95/month for 4 hours of access time. After that, you pay up to \$9.50/hour for daytime access. GEnie's local services are also limited, so if you don't live in an area they cover, you'll pay a connection charge plus long distance for 2400-bps service.

## **Prodigy (1-800-776-3449)**

Prodigy, the child of IBM and Sears, is the most commercial of the commercial services. Prodigy displays advertisements along the bottom of every new screenful of information. Most of its Internet services, including e-mail, have additional fees (for example, \$17.95 for 100 messages a month), and using the e-mail gateway is quite a challenge. The base rates are \$7.95/month for 2 hours of access time, or \$14.95/month for unlimited access.

## **CompuServe (1-800-848-8990)**

Because CompuServe has been around the longest, it has very extensive services and more than 2 million users. Many computer and software vendors have forums on CompuServe, and many sell products via CompuServe. It also has some of the most extensive discussion groups of any online service. The GUIs are well designed. The pitfall of CompuServe is its cost—most of the neat features cost you extra (for example, 60 free Internet messages, \$0.15 each afterwards). CompuServe has a startup fee of \$39.95 (although \$25 is credited to extended services) and an \$8.95/month access fee.

## **America Online (Info: 1-800-227-6364)**

America Online (AOL) has had some trouble with managed growth, but the popularity of this service should speak well of what it has to offer. AOL has some extensive resources as well, a large number of discussion groups, as well as Internet e-mail and Usenet newsgroups. The user interface is nicely designed and easy to use, although it is a bit slow for experienced users. The major advantage for AOL is cost. There are no hidden charges on AOL; all the services are included in your monthly fee. When you weigh the GUI against this, some waiting might not be so bad. AOL charges \$9.95/month for 5 hours of access time, \$3.50 for each additional hour.



**Note:** Because of what we have called "trouble with managed growth" (and what others might call "flooding the Net with newbies"), AOL has a bad reputation on the Internet. A posting on a Newsgroup by an AOL user is likely to be ignored or—more likely—made fun of. Few people on the Net take AOL seriously, but when you look at online service providers, AOL does have a Net presence. It's only a matter of time before new users from other providers arrive, and then the focus of insults will be shifted to them.

---

## Internet Service Providers

The other category of commercial online service providers are set up specifically to give you access to the Internet, rather than restrict you to full access only to their userbases. ISPs function similarly to the online services just described, in that you can connect to them through a local dial-in number. Instead of providing content (large databases, discussion groups, and the like), however, the ISPs provide you with an account on a machine running UNIX or some other multiple-user operating system. They generally make a number of programs available to help you process your e-mail or browse newsgroups. Other than that, the ISPs basically leave you on your own to explore the Internet, with few restrictions and little guidance.

The popularity of the Internet has launched many ISPs, some of which provide national access, others that operate regionally or locally. The types of accounts that ISPs provide make quite a difference to you depending on your experience. A sophisticated user might expect the flexibility of a full UNIX shell account, but a novice might find it confusing. If you aren't comfortable with UNIX or VMS (the operating system on DEC VAXes) you should make sure the ISP offers some sort of menu system to enable you to run the programs you want. When you are dealing with this type of unbridled Net access, customer support also becomes an issue. What kind of support is standard might be a big factor in choosing your ISP.

There are two principal national ISPs, Delphi and Netcom. A local ISP, however, might give you many advantages, including lower cost and better technical support. To find a local ISP near you, contact the Network Information Center, usually called InterNIC. You should be aware that the popularity of the Internet has slowed the response time at InterNIC considerably, so expect to wait a long while for information. You can phone InterNIC at 1-619-455-4600 or send e-mail to:

`info@internic.net`

Some ISPs also talk about membership in the Commercial Internet eXchange (CIX). When the Internet was funded by the NSF, and commercial network traffic was prohibited, CIX was formed to enable businesses a means of obtaining Internet access. Now that the Internet has been all-but-deregulated, CIX membership doesn't really mean much of anything other than membership in an organization of commercial providers. Most local ISPs will not be CIX members, due to the organization's fee requirements. CIX has recently announced that they are going to refuse Internet traffic on their routers from non-CIX members. This might cause delays or bounced mail from ISPs who are not CIX members. If you are going with any of the larger national or regional ISPs that shouldn't affect your service, but it never hurts to check.

Here are descriptions of a few of the larger Internet service providers, to give you an idea of the services they offer. Remember, this is not in any way comprehensive; there are hundreds of others, so shop around.

### **Delphi (1-800-695-4005)**

Delphi is a VAX Cluster in Boston. Unless you live in Boston, you have to log into Delphi via SprintLink or Tymnet. Provided they have a local number in your location, and you call between 6pm and 6am, connection is free. Call between 6am and 6pm and you will pay a \$9.00/hr. connect-time charge. Delphi also has a \$19 startup fee and several plans for service, the most attractive being the 20/20

plan: \$20.00 for 20 hours a month. You should think carefully about the plan you choose, though; Delphi charges another fee if you switch plans.

Delphi's interface is also clumsy and provides no command line. You are limited to the menu choices that Delphi provides, which can easily get to be a dozen levels deep.

### **Netcom (1-800-501-8649)**

Netcom operates on many Sun Workstations located around the country. That gives them more local dial-in numbers. If there isn't a Netcom number near you, then use SprintNet and Tymnet, with the same charges mentioned previously. Netcom's system gives you access to a UNIX shell. If you aren't comfortable with UNIX, Netcom has a menu system and many online help files. Don't rely too heavily on Netcom phone support, though; they have a lot of trouble returning phone calls due to their extraordinary growth. Netcom charges a \$20 startup fee, and a fixed rate of \$19.50 a month, or \$17.50 if your account is auto-billed.

### **Performance Systems International (1-800-82-PSI-82)**

PSI has been in the Internet business a long time. The company was formed to provide Internet connections to defense contractors and research institutions, which were originally the only users of the Internet. That translates into incredibly reliable service. They offer InterRamp, which allows individuals access to the Internet for a \$29 startup fee and \$29/month—but you must pay for three months in advance, and they strongly encourage you to pay \$84 for the third-party interface software they recommend, bringing your grand startup total to a steep \$200.

The most significant advantage that most ISPs—particularly the smaller, locally based ones—can provide is a flat-rate fee structure. Instead of charging you a fee for each hour of connect time, many ISPs are offering service for a single monthly fee, much like basic cable television or local telephone service. The companies that charge for every file you download (notably online providers, such as CompuServe) have done so for some time because they started when the only networks were private and proprietary.

Two factors allow general online providers to keep metering your usage. The first is owning their own national network of local dial-up numbers and can charge whatever they like for access. (Netcom, for one, is making local *points of presence* available around the country, but charging a flat rate, which sets the tone for future competition on that aspect of Internet service.)

The second factor is possessing a critical mass of offerings, regardless of how much it costs to provide the offerings. Shareware is shareware, and you shouldn't have to pay to download it. But if an Internet site offers free access to its archives and is so overwhelmed by seekers of free software or data that you can't get through, where do you turn? Furthermore, some individuals and companies will always have valid reasons not to offer their intellectual property for free to anyone, and so will turn to companies that can guarantee them compensation for their work.

With the advent of the Internet, the ultimate public network, the measured-service pricing strategy of private-network service providers is going the way of the dinosaur and the full-service gas station. Your best deal is to find an Internet service provider who will charge you as little as possible for the time you stay connected.

## Access Through Higher Education

### Mitchell Porter

--by **Tod Foley**

"06-11-92 7.30 p.m., qut link lab.

"today i was busted at the commerce labs, sort of. a middle-aged guy—probably a lecturer—stuck his head in the room and said which subject are you doing? i said nothing at the moment, he asked which course am i enrolled in? i said science, something about the labs at the computer science dept were. . . he started to suggest that i should leave when i suggested it myself and did so.

"i could probably creep back in late at night or something but it wouldnt be the same. so tomorrow morning i shall head for griffith, but really it looks as if my current phase of intense net activity is over. it's strange and unpleasant, but e.g. on the way here i found myself composing farewells etc., and thinking how such a 'spectacular' departure (forced off the net! etc. etc.) would probably positively affect my rep as a net-personality. . ."

—from the net.diaries of Mitchell Porter

Hyperattentive Net surfer, founder of the Aleph e-mail list and researcher/compiler of the widely-circulated and regularly-updated net.guide *High Weirdness By Email*, Mitchell Porter has been described as "Australia's most dedicated net.activist." His unique and prolific diaries, systematic outpourings, and epic works-in-progress can be found stashed in directories all over the world, and his roving eyes are constantly on the watch for related threads throughout the Usenet and Elist universes. So important are his online activities to him that he has, upon occasion, resorted to some extreme means—from personal divestment to begging, borrowing, and stealing terminal time—to stay in touch with the Net.

**TF:** There have been a number of times in the past when you mentioned the degrees you'd gone to just to get access—at one time you sold most of your possessions, at others you have taken long bus rides to get to a terminal; in one piece I remember quite well you even questioned the morality and ethics of your presence at the terminal you were writing from (based upon the fact that an actual student might otherwise have been using that machine).

**MP:** In fact, I sold the only possessions I had that I *could* sell (my books) in order to finance those long bus rides, which I was taking daily. At the time processes were unfolding (on the Aleph Elist) that I felt I had to be a part of, and a campus twenty kilometers away was my only point of access to the Net. I suppose that when I wrote the comment you remember, it was close to the end of semester and the lab was full, and so my presence might have prevented someone from finishing an assignment.

**TF:** What do you think now of these acts?

**MP:** A number of times I've struggled desperately to be online for reasons that proved fruitless: in July 1993, when I thought that some Grand Synthesis or Theory of Everything might come of the struggle, and later that year, when Doctress Neutopia joined Aleph [see "The Rise and Fall of an E-Mail List" in Chapter 5, "E-mail"]. In both situations, I made great efforts which led nowhere, and so when I compare

the ambitions I had with what I actually achieved, the point of it all seems questionable. In my first period of desperately intense Internet access, at the end of 1992, I produced the first incarnation of High Weirdness by Email, which might count for something. Legitimate access has been such a rare experience, that perhaps three-quarters of the thousands of words I've come up with while online were composed in situations where I had reason to be looking over my shoulder. My attitude was that it is important to do what *you* think is important, even if no one else agrees with you, or if you must break rules to do it. One is not setting out to oppose authority, but one does so in order to do whatever one sees to be done.

**TF:** Has your attitude changed at all since those days?

**MP:** Basically, I don't think my attitude has changed. I don't now feel the need to get online so often quite as urgently, perhaps because I now have a rented computer at home, but there were long desperate periods in which I was aware that what I was doing was draining and dangerous, but felt that it needed to be done. (The strength of this feeling depended on the confidence I had at the time in the various causes I espouse.) I might question now the urgency, or even the point, of a number of the things I was trying to do—but the same attitude, that it is possible for one's own judgement as to what must be done to override any number of rules, remains.

*"so anyhow, if i cant get on at griffith, tomorrow i will go to gavin or michael, beg an hour's use of telnet, post alt.qixnews CFD to alt.config, finish off high weirdness 1.1, unsubscribe to everything. i will go see jack later tonight as well - maybe i could have nyxmail forwarded to his account or something. i will really miss leri, as i think do all the other people who are unsubscribing owing to the flow of mail. i think that is a sign of just how important what is happening there is."*

**TF:** What were some of the tricks you used to get to the terminals in the first place?

**MP:** My most desperate periods have all been times in which I had no means of access from home, and something urgent was happening online. In such a situation one needs an Internet-connected terminal and an account to which one can connect. Apart from the rare occasion in which I have dialed into an account from the home of a friend, I have always used terminals on university campuses (more on this later). Sometimes this has simply meant getting online through the help of acquaintances there. They would log me in to their accounts, and then I would telnet from there to my American accounts, and try to process several hundred messages (a week's worth) in an hour or two. My legitimate accounts, most of the time, have all been American. These are the accounts by means of which I read and posted to Usenet and subscribed to mailing lists, on systems like Nyx and Launchpad, which allowed just about anyone who could get to the point of logging in, to at least *apply* for an account on the system.

As I mentioned, I have always used networked terminals on university campuses. This has generally meant using terminals in a student computing lab. What made this possible was that the labs were used by many people, mostly students, and so I could be just another face in the crowd. I had no account on the local system, so everything I ever did was by way of telnet. In some situations an anonymous telnet server could already be found in a menu or subdirectory; in other cases, I had to bring the program with me on disk. In the early days, I found out where these labs were and what to do through contacts in the hacking subculture of one of the three universities here, who were in a constant state of cold war with their own computing administration, as a result of which their computing privileges were constantly being curtailed. Everywhere security loopholes were constantly being opened and closed, so that one never knew whether this might be one's last day online.

**TF:** Did you ever get caught?

**MP:** A number of times, and at least once on every campus. Sometimes this happened at moments when I was simply word-processing a file to be uploaded, and not actually online. In some places the security guards came to recognize me, since I would be the last person in the lab at night—that's how far I pushed my luck, and sometimes I was caught out (for example, they would ask to see my student card).

On one occasion, I was using telnet and typing away, sensed someone sitting next to me, and looked up to see two staff members from the main computing center of that campus. I had met them once before, when using two "public" terminals there. However, I was now in a lab five minutes' walk from that centre, so I presume they had been monitoring remote network activity, noticed that someone was connecting to launchpad.unc.edu, traced the terminal and its physical location, and sent their enforcers down to the lab to find me. We had a brief dialogue, along the lines of:

"You know you shouldn't be doing this."

"I know, but I thought it was more important that I do this, than obey the rules," an opinion which evidently made my discoverers quite unhappy. In any case, they didn't attempt to detain me, and I left fairly quickly. I have never been formally charged with anything by anyone, so arguably I've received very lenient treatment. I also have no clue how much, if anything, I've cost the universities in question through my use of the network. I was once told that they paid for a constant quota of satellite bandwidth, and so at worst I was simply slowing traffic down, but this sounds potentially bogus.

*". . . Just this afternoon I got another "file system full" error message, so I couldn't read further mail, but I have been getting maybe 150+ messages a day. . . I am subscribed to: leri-l, future culture, anarchy-list, zendo, URANTIAL, objectivism, space-investors (have seen this once), HIT-LIST (as of today), weird-l, FWAKE-L, Will, Extropians. . . can't recall any others. Periodicals: I get Non Serviam, Purps, Phirst. I am also subscribed to FNORD-L at nyx, and. . ."*

**TF:** When I lost Net access for a couple of months last year, I found myself experiencing strange, nameless emotions I had never felt before. At times I felt hopelessly disconnected from the world—or at least from the world that most mattered to me. But at other times I realized that, as Adam Fast once said, the Net goes on without you, and when you get back online, you see that it's still pretty much the same as it ever was. This was both enlightening and depressing to me. How does it feel for you to be without Net access for an extended period of time?

**MP:** It is immensely frustrating not to be able to connect. It can be positively harmful to one's psychology if this occurs at a period of intellectual intensity involving online dialogues, or if one is worried about what might be happening in one's absence. This is my experience at any rate. Knowing that the technical means are there for you to do something, that you have urgent uses for these resources, but that you cannot use them: This is a formula for anguish, especially when what you wish to do is difficult enough to begin with, since it seems that entirely unnecessary difficulties are being added to the task.

**TF:** What do you do in your periods of "exile?"

**MP:** In times of "exile," if I could reach a computer, I tended to write diary entries, or compose files to be uploaded when I next got the chance. If there was no computer, I'd do all this on paper. Aleph's

disintegration towards the end of 1993 removed one of my main "attractors," and so I had less reason to be striving constantly to be online. As you say, that proved enlightening and depressing, as I was led to question the point of everything that had transpired there. The upshot was that in 1994, I turned to longer-term individual projects, some of which are now approaching fruition.

**TF:** You've gone through so much to remain in contact with your Net peers, and have so freely disseminated your thoughts and experiences—you're obviously driven by some powerful motivators. To be sure, the public sharing of knowledge and discovery is a major part of this, but you certainly possess personal, idiosyncratic reasons as well—your Net diaries seem to be a good example of the intersection between these two zones of interest. Did the Net *cause* you to become more open and helpful than you originally considered yourself, or did it simply heighten a natural tendency you had toward being prolifically self-aware?

**MP:** Firstly, the Net created an opportunity for openness and helpfulness on a scale that I hadn't encountered at all. An account of a trip might be read by thousands of people; a single piece of information, translated to electronic form, thereby potentially comes within reach of all of today's Net-millions (and tomorrow's billions). For someone who was already motivated, as I was, to communicate certain concepts to as broad an audience as possible, this aspect of the Net was a profound attraction. But certainly my online experience has stimulated and focused such aspirations. When I first found, in late 1992, a forum (the Leri-L mailing list) which I was willing to tell just about anything, I did so, which was no doubt cathartic. But with time I have become more discriminating (with the occasional period of frantic desperation!), owing to a growing sense of purpose—that is, of why I'm here.

An initial impulse was to spread the word about all sorts of ideas, mostly future-oriented and horizon-expanding. Later, as I developed a sense of the Net as a thing to be explored in itself, I began trying to network between the different forces I already saw here. I now see myself as pursuing a rather complex but fairly specific agenda. Nonetheless, being oneself while online means exposing oneself to completely unpredictable influences; the openness often expressed in individual .plans (and now, Web home pages) suggests to me that many people are finding this "risk" worthwhile. You can analyze this in terms of a will to power, incidentally; the more people know, or can know of you, the better they will understand you—and so the more opportunity there is that you can join forces productively.

"Late last week Nirad uploaded about 2 Megs of text files to slopoke for me; Lara has the disks they're on at the moment, so she can see what I put out for the world to see under the name 'Scriptures,' but the files are listed in HWbE v2.0. They included High Weirdness by Email, 7 volumes of Scripture of V\R, the text of The Human Evasion, 'Letters Between Enemies' & 'Venom Crystals,' 'Waves Forest' [incomplete], 'Golem XIV' [incomplete], the text of 'The Conspirators' Hierarchy,' and a few shorter files (including 'all24').

*"I got mail from pauls@umich.edu after my announcement on alt.slack / alt.discordia / alt.religion.kibology / alt.zines; he took \*everything\* and put it in a special directory at redspread, /pub/Weirdness (but created a special directory, /zines/Scripture - the first time that the Scriptures had their own directory anywhere). Jon at the Nameless Site got weird2-0.doc, then i got mail from the guy who runs slopoke ftp, & he created /pub/Weirdness also, and said I should put future uploads in /pub/incoming/mitch. Finally today I checked quartz.rutgers.edu, and there was a new directory: /pub/Scripture. Also HWbE v2.0 has now been posted to alt.slack, alt.discordia, and alt.zines. Things travel quickly these days!"*

Given the history of the Internet, it is no surprise that many of the organizations connected to it are educational institutions. The Internet is growing in popularity in the business and home world, but in education, the Net is nearly a way of life.

In the U.S., virtually all large colleges and universities, both public and private, allow their students some degree of Internet access. In many countries throughout the world, Internet access at a university is also possible. Even small colleges, and many community colleges and vocational schools, have Internet connections.

You can use the access provided by most educational institutions to your advantage. There are several means of obtaining access through an educational institution—some more upfront than others.

First, most educational institutions will allow you to purchase network access on their computer systems. With large state-funded schools, this is almost always the case. Most schools have usage policies that can be stretched to fit just about any kind of personal access to the Net (anything associated with the general education of its citizens), providing you with justification for an account. (It might be more difficult to get access for your business.) The Net is an education in and of itself.

The major pitfall to purchasing a university account is the exorbitant cost. Because the schools have a primary obligation to their students, the cost that they set for non-students can be quite prohibitive. For example, most schools are going to charge you for file storage space on their system, CPU usage time, and connect-time charges, all of which can add up very quickly.

**Table 2.2. The cost of Internet access through a typical university (based on average rates on the slowest machines).**

<i>Activity</i>	<i>Daily Cost</i>	<i>Monthly Cost</i>
Reading E-mail	\$2.40	\$72.00
Sending E-mail	\$2.08	\$62.50
Reading News	\$2.27	\$68.10
Posting News	\$2.50	\$75.00
Storage of Files	\$0.45	\$13.50
Totals	\$9.70	\$291.10

Assuming you don't want to mortgage your home or skip any student loan payments, what can you do to avoid this expense? Consider becoming a part-time student.

The primary mission of any college or university is to educate its students. You will find that most institutions feel that students should take advantage of the Internet's resources as an educational tool (like the library), which is why they provide Internet access to students. The university usually absorbs most of the costs of access as part of its educational mission, though it may charge students a technology fee (ranging from \$20 to \$100 per semester) that covers a mere fraction of the actual costs.

Most colleges offer courses to the community as a community service, or as continuing education programs. That means you can simply register for a night class, and keep current on a topic of

interest—or learn something new—while you enjoy your Net access. You might even be able to take a class that would help you better utilize your Net access. (The actual policies will vary at each institution, of course, but Net access is usually easy to come by.)

For the desperate, there are other means of access. These might not be ethical, but if you're a junkie and need a fix, they might satisfy.

Many universities today offer public computing facilities for students. These labs provide the programs that students need to connect to the e-mail system or network at the university. But these also might help provide *you* with access to the Internet. You should be aware, though, that using labs provided for students might cause you more trouble than it is worth. Many universities restrict usage of clusters to ID-carrying members of the university.

If NCSA Telnet (a communications program developed by the National Center for Supercomputing Applications at the University of Illinois) is installed on the machines in the public facilities, you are in business. Using telnet you can connect to other resources on the Internet, including those services that provide cheap or free Internet accounts to those with telnet access. You won't need to dial into these services because the machines at the university's public facility are essentially dialed-in already.

One drawback of using a campus computing cluster is that you have to go there to use it, and even if they don't expect you to show an ID card, you have to look reasonably like a student whenever you go. On the other hand, a campus cluster is a great place to learn about the Internet; if you notice someone else surfing the Net and you have a question, ask them. People who compute are still mighty neighborly.



For more information about obtaining accounts accessible by telnet only, you can write:

Attn: Guest Accounts  
Free Software Foundation  
545 Technology Square  
Room 426  
Cambridge, MA 02139 USA

Include a self-addressed, stamped envelope for reply. Please attach a note specifying whether you are requesting a new account or wishing to reactivate an already existing one.

---

The students at the school themselves might also be a resource for Internet access. Although it is certainly a violation of ethical usage policy and nearly all academic computing policies, account sharing will get you access to the Internet. The greater the size of the institution, the greater your advantage in this case. As long as you are careful not to abuse the account (for example, don't harass people via e-mail, or engage in some activity that will bring you to the attention of the school's computing services), chances are no one will ever know.

The resources at the schools are limited, and no one wants to pay highly trained computer professionals to monitor all the student accounts all the time. With a little caution, you can get away with this quite easily.

The consequences could be high; if you're caught, the student could lose his or her computing privileges and face suspension—or even expulsion. This type of activity may even be illegal in your state; you both could conceivably be prosecuted. Some precautions you will definitely want to take:

- Don't brag about what you are doing.
- Use the person's name that is on the account, or make up a completely different nickname, when sending mail to people you don't know or signing posts to newsgroups. If people start to notice e-mail and newsgroup posts using a name obviously different from your student friend's user-ID, it may put them on the alert.
- Don't dial into the account long distance or outside the area where students might live. Not many institutions have the equipment to monitor dial-in usage, but some certainly do, and they will eventually notice the anomaly.
- Use common sense, but be prepared to deal with the potential consequences of getting caught. In the end, it might not be worth the money you saved.

## Employer-Provided Access

Many companies are jumping on the Internet bandwagon these days. You'd be hard-pressed to find a company in the computer industry that does not have Net access, but more and more, companies that are not directly related to the computer biz are installing local area networks and purchasing Internet connections. If a large LAN is installed where you work, you might be able to swing a Net account through your place of employment.

As usual, the larger the organization is, the more likely it is to have Net access. It still costs at least \$1500 a month for a company to maintain a continuous 56K-bps Internet feed, so small companies are less likely to have the monetary resources for their own Net connections. But say that your company is large enough to have its own LAN. You have a computer on your desk, and so does everyone else who works there. That's a start. If you and your co-workers can exchange interoffice e-mail, you're on your way to Internet access.

There are several methods companies use for internal mail. If it's a small LAN networked via Ethernet, chances are very good that they use a personal-computer-based mail system designed for an office environment, such as Lotus Notes or WordPerfect Office. These packages are designed for internal mail, but they also have ways of linking to the Internet for mail. It's a matter of converting the protocol they use for mail to an Internet-compatible mail protocol such as Simple Mail Transfer Protocol (or SMTP—a mail protocol commonly used by UNIX machines on the Internet). Setting up an SMTP gateway for a WP Office environment is no easy task, though, and not something you can do off the cuff.

If the computer on your desk is a terminal, or a PC acting as a terminal, or a UNIX workstation, then chances are high that you use a mail system that is already based on SMTP or X.400 (another protocol in use on the Internet) and the only thing you need to do is find out the specific syntax used on your system to route Internet mail outside the office. This is as simple as sending mail to your system administrator. (If you don't know who they are, some good guesses are "root" or "operator".)

Perhaps your company does have Internet access. Perhaps you have obtained an account (or convinced management to make accounts available to razor-sharp employees such as you). Consider that it's going to be your company's rules you play by. Do you really want your sole means of Net access to be through your employer?

Because your employer owns the system you are using for your access, *your employer* decides on usage policies. The areas of electronic law and e-mail privacy are being tested in courts as we speak, and you don't want to end up a test case. You should be very conscious of this fact, and make sure to get all of the policies in writing before you start surfing the net.

The first and foremost issue with employer-provided access is privacy. Your company may decide (and this is within its rights to do so) that the system may only be used for work-related activities. That means they can monitor the system, and even go as far as to *read your e-mail* and search your files to determine that they are being used for work. There are many horror stories on the Net about employers who have read e-mail, and worse, fired employees for what they considered abuse of their systems.

Personal use is the other big issue. Many companies pay a substantial amount of money for access to the Internet, and they don't want you e-mailing your friends or perusing rec.pets.iguanas on their time. That means they might not make Usenet News or IRC Chat available on their system.

You could push the limits of your company's personal-use policies. There's a newsgroup for everything, and chances are you could find a handful related to the work that you do. Slip in a personal newsgroup here and there, and no one is the wiser. Only you can determine how far you can stretch the policy at your company—or if your savings can afford the possible consequences.

Does all this mean that work is out as a link to the Net? The key thing to remember about this elaborate computer system in your office is that people run it. And those people may also be your source to the Net. Any large system has system administrators, the folks who keep your network up and running. Chances are your employer pays the administrators a fair amount of money to watch the system so that they don't have to, and because your boss might not be a computer expert. Sysadmins are skilled in the ways of the Net, and are probably Internet gurus themselves, so this might buy you some slack.

It really depends on the atmosphere at your place of employment. No one wants to compromise their ethics, let alone risk their job to give you a Net feed, but with a little common sense on your part and a friendly enough office atmosphere, you might be able to talk a sysadmin into giving you a separate account on the work system that you can use for your personal Net account. It can't hurt to try; a beer now and then after work can go a long way. (Gurus are rumored to like Guinness Stout.)

## **Cooperative Access**

An Internet co-op is like any other co-op: a group of consumers with similar wants or needs obtains an otherwise unpurchasable commodity by splitting the cost among its users; you and the members of the co-op can all benefit. The idea behind co-op access to the Internet is so new that no co-ops are online yet, but efforts are springing up left and right.

The costs of starting an Internet dial-up service are mainly fixed costs. There is the cost of the initial hardware: a computer to log into, modems to connect to, phone lines, and the Internet connection itself. These costs can be split among members of the co-op, and then the monthly operating costs can also be

split. That way, members can all enjoy access without the expense of purchasing a connection individually.

The most well-known co-op forming is the Toronto Internet Co-operative. The membership in the co-op will be limited to 2000 members so that the cost of adding new local phone lines, modems, or disk drives will be finite, and can be paid for solely with membership fees. The membership fee for the co-op would be pricey at \$230, though that's not much more than PSI's start-up fee. Members of the co-op would then be entitled to unlimited access for \$7 a month, one of the most attractive opportunities around, and would be part-owners of their own Internet node.

The contacts for the Toronto co-op are:

Rocco Racioppo, P.Eng.	Dr. Art Stretton
Professional Engineers Ontario	Toronto Internet Co-operative
(416) 961-1288 ext 369	(905) 883-9555

You may also find that your area might be suitable for a co-op as well. Information is also available from the Toronto co-op on starting your own. It would be a lot of work to go to just to obtain personal Internet access, but it's not impossible to do.

## Alternative Nets

Computer networking is not a new concept, and the Internet is not the only computer network. There are other networks (such as FidoNet, discussed earlier) that are independent of the Internet and there are even more alternative networks available. The lure of the Internet is global access to a variety of information, but if you have limited information needs, there might be an alternative to the Internet that would suit your needs.

For example, if you were interested primarily in creating a wide area network out of your company's far-flung LANs, or if you were interested in receiving only specific information from a single database on a regular basis, one network that specializes in providing access to networked resources is Tymnet (1-800-937-2862). If you have specific business needs, and are looking for a service to provide market information and access to daily news or other items that might be relevant to business, the Dow Jones (1-800-832-1234) network offers those services to business through local dial-in as well.

There are also networks that have evolved around specific professions, which can usually be located by talking to others in your field. Elementary and secondary educators, for instance, often want access to the Internet because of the vast teaching resources it has to offer. Many local school systems have also recognized this and have begun to form networks of their own, designed to help educators contact each other. One such network is IDEAnet, sponsored by the Indiana state department of education. IDEAnet and other educational networks are generally sponsored by some other similar organization, and usually require professional affiliation in the field to gain access.

Another alternative network is PeaceNet, operated by the Institute for Global Communications (IGC) in San Francisco. The purpose of PeaceNet is to link peace workers and people interested in peace-related causes around the world. Many peace organizations are connected to PeaceNet, such as Amnesty International and Greenpeace. PeaceNet is a global network that offers international access in countries

such as Nicaragua, Brazil, Australia, the UK, Canada, Nigeria, Sweden, the former Soviet republics, and Germany.

Although a network like PeaceNet might not offer all the services of full Internet access, many of the services found on the Internet are duplicated on PeaceNet, such as discussion groups and even some USENET Newsgroups. E-mail is also available via PeaceNet, and can also be exchanged with Internet users.

PeaceNet's basic rates are similar to many online services (\$15 startup fee, \$10 monthly, and \$5-10/hr. connect-time charges), but if an alternative network like this suits your needs it might be a better deal. Contact them at:

PeaceNet  
18 De Boom Street  
San Francisco, CA 94107  
415-442-0220

Chances are, however, you want all of the services offered via the Internet. Although specialty networks might be great for specific causes, the appeal of the Internet is both the volume and variety of information and users it offers.

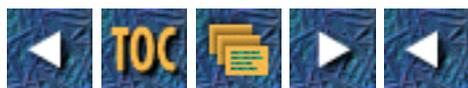
## Summary

If your goals for Internet access are modest—you want e-mail, but it doesn't have to be delivered instantaneously, or you want to peruse the Usenet—obtaining an account from a BBS can cost next to nothing. If you want to chat with others live around the world, or participate in multi-user simulations, or perhaps just get your e-mail right away instead of once a day, you'll need a real-time connection through a FreeNet, an online service provider, or an ISP.

Inevitably, you need to draw the bottom line. BBSs and FreeNets don't cost much, but their resources are limited. The costs of commercial service providers are dropping, but their services are increasing as rapidly as technology does, so they'll always cost something significant. If even a flat monthly rate can't convince you to shell out for a shell account, you might be tempted to try more drastic measures—skimming access from your local university or your employer.

Still daunted? It won't be long before you are able to make your desktop personal computer into its own Internet node. (The technology is already in use in the form of the Serial Line Internet Protocol (SLIP) and the Point-to-Point Protocol (PPP). These are protocols that fool the Internet into thinking that your home computer, connected temporarily via modem, is permanently connected to the Internet. Ask an Internet service provider if they offer a SLIP/PPP connection.) As the convergence of television, telephony, and computing continues unabated, you may soon be offered Internet access right from your house by a telephone company, a long-distance company, a cable company, or even a utility company!

Of course, the sooner you get onto the Internet, the sooner you can become a guru.





- [3](#)
- [Tricks of the Trade: Tools](#)
  - [--by Earl Fogel](#)
  - [File Conversion Tools](#)
    - [Getting Software via Anonymous FTP](#)
    - [Audio Files](#)
      - [PC Audio Software](#)
      - [Macintosh Audio Software](#)
      - [UNIX Audio Software](#)
    - [Image Files](#)
      - [PC Image Software](#)
      - [Macintosh Image Software](#)
      - [UNIX Image Software](#)
    - [Animation Files](#)
      - [PC Animation Software](#)
      - [Macintosh Animation Software](#)
      - [UNIX Animation Software](#)
    - [File Archiving, Encoding, and Compression](#)
      - [PC Software](#)
      - [Macintosh Software](#)
      - [UNIX Software](#)
    - [Automating File Conversions with MIME](#)
  - [Netfind](#)
    - [What Makes Netfind Unique](#)
    - [The Telnet Interface](#)
    - [The Gopher-to-Netfind Gateway](#)
    - [Using Netfind on the World Wide Web](#)
    - [How Netfind Works](#)
    - [Netfind Tips and Tricks \(Searching for Domains, Hosts, . . .\)](#)
    - [Where to Get Netfind Software for UNIX](#)
  - [Finger](#)
    - [History of Finger](#)
    - [Security Issues](#)
    - [Finger: How It Works](#)
    - [Neat Finger Sites](#)
    - [Customizing Your Personal Finger Information](#)

- [GNU Finger](#)
- [Where to Get Finger](#)
  - [UNIX Systems](#)
  - [Macintosh Systems](#)
  - [PC Systems](#)
- [Ping](#)
- [How Ping Works](#)
- [Where to Get Ping](#)

## 3

# Tricks of the Trade: Tools

--by [Earl Fogel](#)

## File Conversion Tools

After you become connected to the Internet, one of the first problems you are likely to run across is how to deal with the many different types of files that are out there. What is a .zip (or .gif, .hqx, .ps, .html, .uue, .tar.Z) file, and what do you do with it after you get it?

This was somewhat less of a problem in the early days of the Internet, because most of the things traveling around on the Net in those days were plain text files such as mail messages and Usenet news. When files were encoded, it was usually with uuencode, which was built-in on UNIX systems, and often unavailable on other types of systems. The "uu" in uuencode stands for UNIX-to-UNIX, and by and large that's all it was good for. Uuencode uses a very simple encoding scheme to convert 8-bit binary data into 7-bit ASCII (plain text) files. Due to an unfortunate design flaw, uuencoded files often become corrupted when they pass through a non-ASCII (for example, IBM mainframe) host. As a result, sending anything but plain text files was difficult enough that most people simply didn't bother.

Times have changed, however, and these days everyone is passing around graphics, audio files, movies and more. If you want to get in on this good stuff too, you'll need to know what the different file types are and you'll have to get your hands on a variety of file conversion utilities to deal with them.

A good source for this information—indeed, for any information—is the news.answers Usenet newsgroup, which contains periodic postings on almost any conceivable subject. Of particular interest to file conversion devotees are the FAQs (answers to Frequently Asked Questions) from the comp.compression, alt.binaries.pictures.d, and alt.binaries.sounds.d newsgroups.

Because the Usenet FAQs are periodic postings, the one you are looking for may not currently be available at your site. If that's the case, or if you simply don't have access to Usenet, you can retrieve any of the Usenet FAQs via anonymous FTP from rtfm.mit.edu, from the pub/usenet-by-hierarchy directory. World Wide Web devotees can also access the FAQs using the following Universal Resource Locator (URL):

<http://www.cis.ohio-state.edu/hypertext/faq/usenet/FAQ-List.html>

Another good place to look for this sort of information is in the comp.sys Usenet newsgroup hierarchy, where you will find groups devoted to most computer platforms, including Apple ][, Amiga, Atari, DEC, HP, IBM, Macintosh, NeXT, Sun, and many more.

Now, back to the original question. What is a .zip (or .gif, or. . .) file, and what can you do with it? In a moment, I'll list all of the common types of audio, image, video, encoded, archived, and compressed files, and suggest software you can use to handle them. But any such list is bound to be incomplete, so it's worth spending a few moments to consider what you can do when you find a file type that *isn't* listed here.

The first thing to do is to look in the FAQs for the comp.compression, alt.binaries.pictures.d, and alt.binaries.sounds.d Usenet newsgroups mentioned earlier. That's where you'll find the most complete listings of file types and their meanings, and it's also where you'll find the most up-to-date listings of software used to view, play, or otherwise manipulate these files.

If the information you are looking for isn't in one of those FAQs, the next best approach is to ask for help on one of the relevant Usenet newsgroups. Almost certainly, somebody out there knows the answer to your question, and would be more than willing to help.

## Getting Software via Anonymous FTP

This section has been included for those readers who don't know how to use anonymous FTP or who don't understand the cryptic notation I use for FTP retrieval information. If you already know this stuff, just skip this section—or read it over anyway, just in case. If you don't learn anything new, you may still enjoy the warm fuzzy feeling you get when something confirms just how much you already know.

All of the public domain and shareware software described in this chapter may be retrieved over the Internet via anonymous FTP. In most cases, the packages are available from many different FTP sites, but to save space, I only list one. The retrieval information given here is correct as I write this, but some of it will almost certainly be out of date by the time you read it. If you can't find one of these packages using the exact retrieval information given here, you might want to check for newer versions of the same package at the same location. You can also use Archie to find alternative sources for any of this software.

The retrieval information is given in the following format:

```
sitename.domain:path/file
```

For example, the retrieval information for WHAM (a Windows audio player) is given as:

```
ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/wham131.zip
```

To retrieve this file, use FTP to connect to the machine named ftp.ncsa.uiuc.edu, login as anonymous, give your e-mail address as your password, go to the /PC/Mosaic/Windows/viewers directory, and retrieve the file wham131.zip in binary mode.

If you happen to be using NcFTP, you can use the retrieval information exactly as given, by putting it on the NcFTP command line. For example, to retrieve WHAM using NcFTP on a UNIX host, you would type:

```
% ncftp ftp.ncsa.uiuc.edu:/PC/Mosaic/viewers/wham131.zip
```



**Tip:** You can also retrieve these files using a World Wide Web browser, by massaging the retrieval information into Universal Resource Locator format. Instead of using *sitename.domain:path/file*, you would use <ftp://sitename.domain/path/file>. For example, to retrieve WHAM using the Windows version of Mosaic, select "Open URL" from the File menu, and type the following text into the dialog box that appears:

```
ftp://ftp.ncsa.uiuc.edu/Web/Mosaic/Windows/viewers/wham131.zip
```

## Audio Files

Audio data comes in many different formats. Some files are monaural, others are in stereo. Many different combinations of sampling rate, sample size, and number of channels are in use, and audio file formats differ considerably from one system to another.

To convert sounds into digital format, they are sampled many times per second. The more samples that are taken per second, and the more bits used to store each sample, the better the results. Standard telephone-quality sound can be achieved with 8000 samples per second, using 8 bits to store each sample. Sound reproduction equivalent in quality to that of an audio CD requires 44,000 16-bit samples per second.

Fortunately, you don't have to worry about this very often. In most cases, your audio software can figure out what type of file you are feeding it, and make any necessary conversions to get it to play on your system.

Here is a list of some of the most common audio file types:

**Table 3.1. Common audio file types.**

<i>File Extension</i>	<i>Description</i>
.au	NeXT or Sun audio file
.snd	Mac, NeXT, PC or Sun sound file
.aif, .aiff, .aifc	Apple or SGI sound file
.voc	Sound Blaster sound file
.wav	Microsoft (windows) sound file
.iff, .mod, .nst	Amiga sound file

These days, the most common types of sound files you will find on the Internet are Sun audio files and Macintosh sound files.

The hardware you need to play audio files comes as standard equipment on all Macintoshes, most Sun workstations, and on many other personal computer and workstation brands. The hardware required to play sounds is not present, however, on many PCs. To get reasonably good quality sound from a PC, you will need to invest in a sound card and external speakers.

---

 **Note:** All of the current Sun workstation offerings (except for the older SPARC Classic) include 16-bit audio. Sun's internal speakers are capable of AM-radio quality (unlike PC speakers). External speakers are required for CD-quality sound.

---

Many software packages, both public domain and commercial, are available to convert and play audio files on various platforms. There are too many packages to list them all, so I will mention just a few for each of the major platforms. I'll try to pick packages that are commonly used, easily available, and reasonably powerful, but don't be surprised if I've left your favorite off the list. If the package I've chosen doesn't suit you, read the Audio File Formats FAQ from the alt.binaries.sounds.d Usenet newsgroup and pick one that is more to your taste. You should also consult the FAQ if you have an audio-capable system that I don't mention here, such as a DEC VAXstation 4000.

## PC Audio Software

To get reasonable quality sound out of a PC, you will need to invest in a sound card and external speakers. If you want to listen to Sun or Macintosh sound files on a PC, you will need to find some audio conversion software, such as SOX, WHAM, or wplany.

SOX (SOund eXchange) converts between most common audio file types. The PC version of SOX is available from:

<ftp.cwi.nl:/pub/audio/sox5dos.zip>

WHAM (*Waveform Hold and Modify*) is a Windows 3.1 application for manipulating and playing sound files.

<ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/wham131.zip>

Wplany is an audio player that works with the built-in PC speaker.

<ftp.ncsa.uiuc.edu:/PC/Mosaic/viewers/wplny09b.zip>

## Macintosh Audio Software

Support for sound is built into all Macintoshes, and if you are running System 7, you can play Mac sound files by double-clicking on them in the Finder. For other types of sound files, however, you'll need to get a sound conversion utility such as SoundMachine or SoundExtractor.

SoundMachine reads and plays most common sound formats.

`sumex-aim.stanford.edu:/info-mac/snd/util/sound-machine-21.hqx`

SoundExtractor reads, plays, and converts between most common sound formats.

`sumex-aim.stanford.edu:/info-mac/snd/util/sound-extractor-131.hqx`

## UNIX Audio Software

Sun Workstations include sound support, although the mechanism for playing Sun sound files varies from one release of SunOS or Solaris to another. On some systems, the built-in sound command is "play," on others it is "audioplay" or "audiotool," and on a few Suns, you may have to resort to the rather ugly `cat > /dev/audio`.

Sun's Audio Tool is shown in Figure 3.1.

### Figure 3.1. Sun's Audio Tool.

SOX (SOund eXchange) runs on most UNIX systems and converts between most common audio file types. SOX source code was posted in eight parts to the alt.sources Usenet newsgroup, and is available from:

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.12.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.13.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.14.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.15.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.16.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.17.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.18.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.19.2`

`ftp.sterling.com:/usenet/alt.sources/volume93/Jul/930726.20.2`

## Image Files

Table 3.2 shows some of the most common image file types.

**Table 3.2. Common image file types.**

<i>File Extension</i>	<i>Description</i>
.bmp	Windows or OS/2 bit-mapped picture file
.gif	CompuServe's Graphics Interchange Format
.im8, .img	Sun Image file
.jpe, .jpg, .jpeg	Joint Photography Experts Group file
.pcx	PC Paintbrush file
.pict	Macintosh QuickDraw PICTURE file
.eps, .ps	(Encapsulated) PostScript file
.tif, .tiff	Tagged Image Format File
.xbm, .xwd	X Window BitMap, X Window Dump files
.pbm, .ppm, .pgm	Portable BitMap, PixMap and GreyMap formats

These days, GIF is the most popular image format on the Internet, with JPEG running a close second. Both GIF and JPEG images can be viewed on many different platforms, and they have built-in compression to reduce the size of the image files. GIF images can have a maximum of 256 colors.

While not as popular as GIFs, JPEG images are becoming more and more common on the Net. JPEG uses a *lossy* compression format, which means that small amounts of image data are sacrificed to offer much better compression. JPEG files are often much smaller than comparable GIF images, with negligible loss of quality. JPEG images can have up to 16.7 million different colors.

The next most common image file type you are likely to find on the Internet is PostScript. PostScript is a special-purpose language used to describe printed pages in a device- and resolution-independent manner. PostScript files can contain image data, as well as text and printer commands (such as a command to put a printer into duplex mode). Because of its complexity, only a handful of programs can successfully deal with PostScript. These include GhostScript, GhostView, Adobe Illustrator, and Abode's Display PostScript.

Both GIF and JPEG are *raster image formats*. Raster images are stored as a rectangular array of dots, or pixels (picture elements). *Vector images*, on the other hand, are stored as a sequence of drawing operations. Raster formats (sometimes called *paint files*) are more common than vector formats (sometimes called *draw files*). Converting from a vector format to a raster is fairly easy, as is converting from one type of raster image to another. It's much harder to convert a raster image into vector format.

Some images are monochrome, others are greyscale or 4-, 8-, 16- or 24-bit color. The nicest and largest files you are likely to come across are 24-bit color (sometimes called *true color*). More than likely you won't have the video hardware needed to display 24-bit color images, so your image display software will convert them down to 8-bit color. Image files are *big*, and most of the image file types support one or more compression schemes to reduce the files to a more manageable (but still large) size.

In monochrome images, each pixel is represented by a single bit of computer storage. That is just enough to represent two colors (for example, black and white). 4-, 8-, 16-, and 24-bit color images can contain 16, 256, 32 thousand, and 16 million colors, respectively. It takes just over 2300 bits to store a 48x48 dot monochrome picture. A full-screen (1152x900), full-color (8-bit) image on my Sun workstation takes up over 8 million bits (1MB) of storage. In compressed format, that image might fit into a 200KB file, but when it comes time to display it, the image will be expanded to its full size in memory, which is why good video boards include 1MB or more of graphics memory.

To view images on your own system, you need graphics hardware (often called a *frame buffer* or *video card*), graphics memory, and a bit-mapped display. Generally speaking, the faster the graphics hardware, and the more graphics memory you have, and the larger the monitor, the better for displaying images.

Many software packages, both public domain and commercial, are available to convert and display image files on the various platforms. As with audio files, there are too many image utilities to list them all, so I will just hit the highlights. For more information, read the FAQ from the alt.binaries.pictures Usenet newsgroup. Also consult the FAQ if you have an system that I don't mention here, such as an Atari ST or an Amiga.

## PC Image Software

Graphic Workshop for DOS converts and displays most major image formats. Figure 3.2 shows an example of graphics conversion with Graphic Workshop.

### [Figure 3.2. File conversion in Graphic Workshop.](#)

You can get Graphic Workshop from:

`nic.switch.ch:/mirror/msdos/graphics/grfwk70b.zip`

PaintShop for Windows also converts and displays most common image formats. The PaintShop Pro file conversion help screen is shown in Figure 3.3.

### [Figure 3.3. File conversion in PaintShop Pro.](#)

PaintShop Pro is available from

`oak.oakland.edu:/pub/msdos/windows3/pspro200.zip`

GhostScript and Ghostview are Postscript document viewers for Windows:

<ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/gs261exe.zip>

<ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/gsview10.zip>

## Macintosh Image Software

GIFConverter converts between most common image formats.

`Mac.archive.umich.edu://mac/graphics/graphicsutil/gifconverter2.37.hqx`

JPEGView can display JPEG and GIF images.

<ftp.ncsa.uiuc.edu:/Web/Mosaic/Mac/Helpers/jpeg-view-33.hqx>

## UNIX Image Software

XV is a very nice image viewer for X11. It views and converts between most common image types. The more recent (shareware) versions of XV also perform many image processing operations. See Figure 3.4 for an example of a screen showing file conversion with XV. XV also runs on VMS systems. You can get the source code for XV from

<ftp.cis.upenn.edu:/pub/xv/xv-3.00a.tar.Z>

### [Figure 3.4. File conversion with XV.](#)

NetPBM (as well as its predecessor PBMPlus) can convert just about any image format into any other format, and it runs on many platforms (UNIX, VMS, DOS, OS/2, Amiga). It also can perform a variety of image-manipulation

operations. However, it is command-line oriented, and it doesn't actually display images—it just converts them. NetPBM is available from many locations, including:

[sunsite.unc.edu:/pub/X11/contrib/netpbm-1mar1994.tar.gz](http://sunsite.unc.edu:/pub/X11/contrib/netpbm-1mar1994.tar.gz)

GhostScript is a Postscript previewer available on most platforms, including UNIX/X11.

[ftp.ncsa.uiuc.edu:/Web/Unix/viewers/ghostscript-2.6.tar.z](http://ftp.ncsa.uiuc.edu:/Web/Unix/viewers/ghostscript-2.6.tar.z)

## Animation Files

Animation files contain a series of still-frame images that are displayed in sequence on your computer. If the images are small, and your hardware is fast, the sequence of images will be displayed quickly enough that it appears as live action motion.

Here is a list of some common animation file types:

**Table 3.3. Common animation file types.**

<i>File Extension</i>	<i>Description</i>
.mpeg, .mpg, .mpe	Moving Picture Experts Group movie file
.qt mov	Macintosh Quicktime movie
.movie	Silicon Graphics movie file
.dl, .gl	Animated picture file
.flc, .fli	Animated picture file

These days, the most common types of animation files you will find on the Internet are MPEG and Quicktime. Like JPEG, MPEG is a new format that is rapidly gaining in popularity and can be displayed on many different platforms.

Some animation files literally contain sequences of images, but others save space by storing only the differences between successive frames. Some animation files (such as Quicktime, for example) include audio information as well, so that the animation can be accompanied by a soundtrack.

Because animations are essentially just sequences of images, they have hardware requirements similar to those of image files. As with displaying images, the faster your graphics hardware, and the more graphics memory you have, and the larger the monitor, the better.

Many software packages, both public domain and commercial, are available to display animation files on the various platforms. As with audio and image files, there are numerous animation utilities, so I will just highlight some of the popular ones. For more information, consult the FAQ from the alt.binaries.pictures.d Usenet newsgroup. Also consult the FAQ if you have an system that I don't mention here, such as an Atari ST or an Amiga.

## PC Animation Software

Lview is a free GIF/JPEG viewer:

[ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/lview31.zip](http://ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/lview31.zip)

MPEGPLAY is a shareware mpeg viewer for Windows:

[ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/mpegw32e.zip](http://ftp.ncsa.uiuc.edu:/Web/Mosaic/Windows/viewers/mpegw32e.zip)

MPEGXing is a free Windows MPEG viewer from Xing Technology:

<A HREF="S2K-ftp.cs.berkeley.edu:/pub/multimedia/mpeg/Ports/xing/xing2.0.tar.Z">  
S2K-ftp.cs.berkeley.edu:/pub/multimedia/mpeg/Ports/xing/xing2.0.tar.Z

## Macintosh Animation Software

Simple Player is a Quicktime movie player from Apple. It is included with most versions of Quicktime. Simple Player is shown in Figure 3.5.

### Figure 3.5. Playing movies with Simple Player.

Sparkle is an MPEG movie player for the Mac.

<ftp://ftp.ncsa.uiuc.edu:/Web/Mosaic/Mac/Helpers/sparkle-213.hqx>

## UNIX Animation Software

Mpeg\_play is a free MPEG viewer for UNIX/X11 systems.

[s2k-ftp.cs.berkeley.edu:/pub/multimedia/mpeg/mpeg\\_play-2.0.tar.Z](ftp://s2k-ftp.cs.berkeley.edu:/pub/multimedia/mpeg/mpeg_play-2.0.tar.Z)

Xanim plays most other animation files on UNIX/X11 systems, including .dl, .fli, .gif, and .iff. Available from many locations including:

[syr.edu:/software/X/xanim229.tar.Z](ftp://syr.edu:/software/X/xanim229.tar.Z).

## File Archiving, Encoding, and Compression

Why do people encode, compress, or archive files?

Binary files must be encoded into an ASCII format in order to transmit them through text-only media such as electronic mail or Usenet news.

Macintosh files must be encoded in order to preserve their resource forks and file-type and creator information when they are transferred to other systems. Similarly, some VMS files must be encoded to preserve file-type and record-structure information.

Encoding a binary file typically increases its size by 30 or 35 percent. To compensate for this, people usually compress files before encoding them. There's no point in compressing a file after it's encoded, of course, because that would undo the benefits of the encoding—the result would no longer be a plain-text file.

Groups of related files are often archived (combined into one file). This ensures that none of the parts gets lost or mislaid in transit, and also simplifies the downloading process.

Most archiving utilities (other than the venerable UNIX tar and shar programs) also compress files. Because tar doesn't, most tar archives are compressed or gzipped as well, and you see them on the Net as .tar.Z or .tar.gz files.

By and large, audio, image, and video files are quite portable, and, given the right hardware, can be displayed on practically anything.

Archived, encoded, and compressed files, though, tend to be much less portable. It's not that they are intrinsically less portable, but that they are more often intended for use on a single type of machine. With the exception of Zip and occasionally Zoo archives, these files are intended for use on a particular type of system, and unpacking them elsewhere is usually more trouble than it's worth.

Even if you manage to find software that can unpack a UNIX tar archive on a Macintosh, chances are it won't do you any good; most tar archives contain UNIX source code, which won't compile on a Mac. Similarly, unpacking a Macintosh Compactor Pro archive won't do you much good on a PC; more than likely, the archive contains Macintosh software, which won't run on your PC.

Table 3.4 lists the most common encoded, compressed and archive file types.

**Table 3.4. Common encoded, compressed, and archive file types.**

<i>File Extension</i>	<i>Description</i>	<i>Capabilities</i>
.arc	The old DOS archive standard	Archiving and compression
.cpt	Compact Pro for Mac	Archiving and compression
.gz	GNU gzip	Compression only
.hqx	Macintosh BinHex	Encodes Mac files preserving resource fork, file type and creator
.lha, .lzh	LHarc	Archiving and compression
.sea	Macintosh	Self-extracting archive
.shar	UNIX shell archive	Archiving only
.sit	Stuffit for Macintosh	Compression and archiving
.tar	UNIX Tape Archive	Archiving only
.uue, .uu	UNIX UUEncoding	Simple binary-to-ASCII encoding
.z	pack or gzip (see .gz)	Compression only
.Z	UNIX compress	Compression only
.zip	Phil Katz's pkzip	Archiving and compression
.zoo	Rahul Dhesi's zoo	Archiving and compression

## PC Software

Phil Katz's pkunzip is the original DOS unzip utility. Find it at

`oak.oakland.edu:/msdos/zip/pkz204g.exe`

Rahul Dhesi's zoo creates and unarchives DOS zoo files. Find it at

`oak.oakland.edu:/msdos/zoo/zoo210.exe.`

The Gnu zip package is available for DOS systems, and it is useful for uncompressing .gz files found on the Internet.

`labrea.stanford.edu:/pub/gnu/gzip-1.2.2.msdos.exe`

Uuexe, a UNIX-compatible uuencode/uudecode utility for DOS, enables PC users to decode uuencoded files found on the Internet. Uuexe is available from:

`oak.oakland.edu:/pub/msdos/decode/uuexe525.zip`

## Macintosh Software

BinHex decodes Macintosh BinHex files. Find it at

`sumex-aim.stanford.edu:/info-mac/util/binhex-40.hqx.`

StuffIt Expander decompresses Macintosh StuffIt files. Find it at

`sumex-aim.stanford.edu:/info-mac/cmp/stuffit-expander-351.hqx.`

Info-zip's portable unzip program is available for the Macintosh, enabling Mac users to unarchive .zip files found on the Internet.

Unzip for the Macintosh is available from

`sumex-aim.stanford.edu:/info-mac/util/unzip*-* .hqx`

Uulite, a UNIX-compatible uuencode/uudecode utility for the Macintosh, is available from `mac.archive.umich.edu:/mac/util/compression/uulite1.6.cpt.hqx`

## UNIX Software

Tar, compress, uuencode and uudecode are built into almost all UNIX systems.

Info-zip's portable unzip program is available for UNIX systems.

Figure 3.6 shows an example of the use of unzip.

### Figure 3.6. Unpacking an archive with unzip.

The source code for unzip is available from

`oak.oakland.edu:/pub/misc/unix/unzip512.tar.Z`

The Gnu gunzip program (also known as zcat) uncompresses gzipped files. It is available from

`labrea.stanford.edu:/pub/gnu/gzip-1.2.2.tar`

Mcvvert is a UNIX utility that decodes Macintosh BinHex files. Find it at

`sumex-aim.stanford.edu:/info-mac/cmp/mcvvert-215.shar`

## Automating File Conversions with MIME

MIME (the Multi-part Internet Mail Exchange protocol) is the Internet standard mechanism for sending nontext documents through electronic mail. MIME provides a general mechanism for attaching text and nontext files to mail messages, for specifying various types of documents, and for automatically encoding, decoding, and viewing different document types.

For each document, MIME distinguishes between its "content type" (what it is) and its "encoding type" (how it is stored, or how it has been encoded for transmission via e-mail). Through its use of a standard set of content and encoding types, MIME makes it possible for applications to automatically encode, decode, and display many different types of files.

For example, suppose I have an image of myself stored in the file `earl.gif`. If I have a properly configured, MIME-compliant mail program, I can send that file to you with a simple attach command, and let my mail program worry about whether (and how) to encode the file for transmission through the mail.

If you also have a MIME-compliant mail program, then when you receive the file your mail software would read the MIME content type and encoding type from the message header and automatically decode the file. This is shown in Figure 3.7.

### Figure 3.7. Receiving a MIME mail message.

MIME is increasingly being used in nonmail applications—such as Gopher+ and the World Wide Web—as well, so that they too can automatically encode, decode, and view many different types of documents.



**Tip:** Although the MIME types are standardized on all systems, the mechanism for specifying them varies from one platform to another. NCSA Mosaic for Windows, for example, stores this information in the `mosaic.ini` file in your Windows directory. On UNIX, MIME information is stored in two separate files: `mime.types` and `mailcap`.

---

The `mime.types` file provides a mapping between the filename extensions and their corresponding MIME types. Here is

a sample mime.types file:

```
# Sample mime.types file

application/postscript      eps ps
audio/basic                 au snd
image/gif                   gif
image/tiff                  tiff tif
text/html                   html
text/plain                  txt c cc h
video/mpeg                  mpeg mpg mpe
video/quicktime             qt mov
```

The mime.types file in this example says that filenames ending in .eps or .ps are PostScript files, filenames ending in .au or .snd are basic audio files, and so on.

The companion file to mime.types is mailcap. The mailcap file specifies how each document type should be displayed. Here is a sample mailcap file:

```
# Sample mailcap file

image/*; xv %s
application/postscript ; ghostview %s
video/mpeg; mpeg_play %s
video/*; xanim %s
```

The mailcap file in this example directs MIME-compliant software such as Lynx and NCSA Mosaic to display images with xv, to display PostScript files with ghostview, to display mpeg videos with mpeg\_play, and to display all other types of videos with xanim.

If you set up the list of MIME types and viewers properly, and if you install all the right viewer applications, then whenever you use a World Wide Web or Gopher+ client or a MIME-compliant mail program, all the required file conversions will happen automatically for you—like magic.

## Netfind

Netfind is a wonderful tool for finding people on the Internet, and it is an indispensable part of any Internet guru's toolkit.

Technically, Netfind is an Internet White Pages directory facility. That is, you can use Netfind to find people on the Internet in much the same manner as you use the White Pages in a telephone book to find people in your city.

Netfind was created by Mike Schwartz at the University of Colorado in Boulder, and while it's not the only tool you'll need to find people on the Net, it is one of the best.

The way that I look for people on the Internet is to look in the Phone Books-Other Institutions menu on the Notre Dame

University Gopher (gopher.nd.edu). This menu lists all known Whois, CSO and X.500 directory services, and it can be used to determine quickly whether a site maintains a searchable phone book of some description. If a site has a phone book, then using that phone book is probably the quickest and most reliable way to find people at that site. If there is no locally maintained phone book, or if the person I am looking for isn't listed, then I'll try Netfind, usually via the Gopher-Netfind gateway on the Internic Gopher (ds.internic.net, port 4320).

## What Makes Netfind Unique

Many organizations maintain electronic phone books or user directories of one form or another. Some sites have whois servers or CSO phone books, others use X.500 directory servers, and others use WAIS databases. Sometimes these directories are complete, accurate and up-to-date, and sometimes they are not.

Most tools for finding people on the Internet rely on one or another of these directories. If you are looking for my e-mail address, for example, and you happen to know that the University of Saskatchewan maintains a whois server, and you know its Internet address (whois.usask.ca), and the data on the server is up-to-date, and you have whois client software, then you can find me very quickly.

If a site doesn't have a directory server, or you can't find its address, or the information on the server is out-of-date, or you don't have the right tools, then most of the Internet White Pages tools will fail. Most, but not Netfind.

Given the name of a person on the Internet and a rough description of where that person is, Netfind attempts to locate telephone and e-mail information about them. It does this by going out over the network and interactively querying various hosts, looking for people based on the search keys you provide.

Because it performs an interactive search, rather than using an existing directory of users, Netfind can often find people when other methods fail. If a site does not have a user directory, or if the directory is out-of-date or incomplete, Netfind may be the only way to find people there.

On the other hand, as a consequence of its interactive nature, Netfind can be slow. Also, while Netfind may find *an* electronic mail address, it may not always find the *best* address. This is particularly true for people who have accounts on many different systems.

## The Telnet Interface

To use Netfind, telnet to any of the following sites and log in as netfind:

<b>North America:</b>	
bruno.cs.colorado.edu	(University of Colorado, Boulder)
ds.internic.net	(InterNIC Directory and DB Services, South Plainfield, New Jersey)
mudhoney.micro.umn.edu	(University of Minnesota, Minneapolis)
eis.calstate.edu	(California State University, Fullerton)
hto-e.usc.edu	(University of Southern California, Los Angeles)
netfind.ee.mcgill.ca	(McGill University, Montréal, Québec, Canada)
netfind.oc.com	(OpenConnect Systems, Dallas, Texas)
netfind.sjsu.edu	(San Jose State University, San Jose, California)
redmont.cis.uab.edu	(University of Alabama at Birmingham)

## South America:

dino.conicit.ve	(National Council for Technology and Scientific Research, Venezuela)
malloco.ing.puc.cl	(Catholic University of Chile, Santiago)
netfind.if.usp.br	(University of São Paulo, Sao Paulo, Brazil)
Europe:	
monolith.cc.ic.ac.uk	(Imperial College, London, England)
netfind.icm.edu.pl	(Warsaw University, Warsaw, Poland)
netfind.vslib.cz	(Liberec University of Technology, Czech Republic)
nic.uakom.sk	(Academy of Sciences, Banská Bystrica, Slovakia)

<b>Australia:</b>	
archie.au	(AARNet, Melbourne, Australia)
netfind.anu.edu.au	(Australian National University, Canberra)

<b>Asia:</b>	
krnic.net	(Korea Network Information Center, Taejon, Korea)
lincoln.technet.sg	(Technet Unit, Singapore)

When you connect to Netfind with telnet, you will see a screen similar to the following:

```
% telnet netfind.ee.mcgill.ca
```

```
Trying 132.206.64.2...
```

```
Connected to Excalibur.EE.McGill.CA.
```

```
Escape character is '^]'.
```

```
SunOS UNIX (excalibur)
```

```
login: netfind
```

```
password:
```

```
=====
```

```
Welcome to the Microelectronics and Computer Systems  
Laboratory Netfind Server.
```

```
=====
```

```
...
```

```
Top level choices:
```

1. Help
2. Search
3. Seed database lookup
4. Options
5. Quit (exit server)

>

After you are connected to Netfind, select option number 2 to perform a search. Then, when prompted, type the name of the person you are looking for, followed by a set of location keywords, such as

```
schwartz boulder colorado university
```

The name can be a first, last, or login name, but (in most cases) only one name should be specified. The keys describe where the person may be found, and they typically include information such as the name of the individual's institution or its location (city, state, country) or both.

After you've entered a name and location keys, one of three things may happen:

1. If you've specified too many keys, Netfind may not find any hosts to search, and you'll have to try again with fewer, or different, search keys. This is shown in Figure 3.8.

#### **Figure 3.8. A Netfind search with too many keys.**

1. If you've specified too few keys, there will be too many hosts to search, and Netfind refuses to proceed. If this happens, Netfind presents you with a list of Internet domains that match your query and asks you to narrow down the search by selecting a few of these domains for further consideration. This is shown in Figure 3.9.

#### **Figure 3.9. A Netfind search with too few keys.**

1. Finally, if your search keys are just right, Netfind can proceed, searching a small number of hosts (up to fifty) in the domains that match your query. This phase of the search may take anywhere from a few seconds to a few minutes. This is shown in Figure 3.10.

#### **Figure 3.10. A successful search with Netfind.**

As the search proceeds, Netfind keeps you informed of its progress by displaying information about the hosts it is searching and which search techniques it is using.

The search results are often several pages long and can quickly scroll off your screen; if you have some sort of session capture utility, it's a good idea to turn it on before running Netfind.

### **The Gopher-to-Netfind Gateway**

To use the Gopher-to-Netfind gateway, give your Gopher client the information from one of the following Gopher

bookmarks:

Type=1

Name=Netfind Searches

Path=netfind

Host=archie.au

Port=4320

Type=1

Name=Netfind - Network Wide E-mail Searches

Path=netfind

Host=gopher.vslib.cz

Port=4320

Type=1

Name=Netfind Gateway

Path=netfind

Host=ds.internic.net

Port=4320

Type=1

Name=Network Wide E-mail Searches

Path=netfind

Host=mudhoney.micro.umn.edu

Port=4324

When you select one of these bookmarks in Gopher, you will be presented with a short menu similar to that in Figure 3.11.

### **[Figure 3.11. Searching Netfind with Gopher.](#)**

Select option number 1, Search Netfind for E-mail Addresses, to perform a search; then, when prompted, type the name of the person you are looking for, followed by a set of location keywords, such as

```
lindner minneapolis minnesota
```

The name can be a first, last, or login name, but (in most cases) only one name should be specified. The keys describe

where the person may be found, and they typically include information such as the name of the individual's institution, its location (city, state, country), or both.

After you've entered a name and location keys and you submit the search, Netfind presents you with a Gopher menu of all the domains that match the location keys in your query, as shown in Figure 3.12.

### **Figure 3.12. First phase of a Gopher/Netfind search.**

You can now choose any of these domains for further consideration, as shown in Figure 3.13. After you select a domain, Netfind proceeds to search individual hosts within the domain you've selected. This phase of the search may take anywhere from a few seconds to a few minutes.

### **Figure 3.13. Second phase of a Gopher/Netfind search.**

When the search is complete, the results will be returned to you in the form of a Gopher document. Netfind includes information about the hosts it searched and the search techniques it used along with the search results, so this document may be several pages long.

## **Using Netfind on the World Wide Web**

Currently, there is no gateway linking the World Wide Web to Netfind. Most World Wide Web browsers can, however, communicate directly with Gopher servers, so you can use the Gopher-to-Netfind gateway from your favorite World Wide Web browser.

To use Netfind from a Web browser, connect to one of the following Universal Resource Locators (URLs):

```
gopher://archie.au:4320/1netfind
```

```
gopher://gopher.vslib.cz:4320/1netfind
```

```
gopher://ds.internic.net:4320/1netfind
```

```
gopher://mudhoney.micro.umn.edu:4324/1netfind
```

When you connect to one of these URLs, you are presented with a short menu similar to the one shown in Figure 3.14.

### **Figure 3.14. Searching Netfind with Mosaic.**

Select the first link, Search Netfind for E-mail Addresses, to perform a search and when prompted, type the name of the person you are looking for, followed by a set of location keywords, such as

```
berners-lee cern
```

The name can be a first, last, or login name, but (in most cases) only one name should be specified. The keys describe where the person may be found, and they typically include information such as the name of their institution, its location (city, state, country), or both.

After you've entered a name and location keys, Netfind presents you with a World Wide Web document listing all the domains that match the location keys in your query, as shown in Figure 3.15.

### **Figure 3.15. First phase of a World Wide Web/Netfind search.**

You can now choose any of these domains for further consideration, as shown in Figure 3.16. After you select a domain, Netfind proceeds to search individual hosts within the domain you've selected. This phase of the search may

take anywhere from a few seconds to a few minutes.

### Figure 3.16: Second phase of a World Wide Web /Netfind search.

As with the Gopher interface to Netfind, the search results will be accompanied by information about the hosts searched and the search techniques used, so the resulting document may be several pages long.

## How Netfind Works

A Netfind search proceeds in two phases.

The first phase is to narrow down the search using the location keys you provide. Netfind does this by looking up your keys in a *site database*, which contains information on Internet domains, the names of organizations, and their locations.

The site database is created by periodically scouring the Net, gathering and collating information from many different sources (including Internet domain name searches, UUCP (mail) maps, network traffic logs, Usenet message headers, and so on). Much of this work can be performed automatically, but some of it is still done by hand by Netfind's creator, Mike Schwartz at the University of Colorado in Boulder. Every few weeks, all of the other sites that run their own Netfind servers FTP a copy of the site database from the University of Colorado.

The location keys are used to narrow down the the scope of the search to a small number of hosts, usually in a single Internet domain. Using the site database, Netfind tries to find up to three Internet domains that match your query and that are worthy of further consideration.

The choice of search keys is crucial to the success of Netfind. Like Goldilocks with her porridge, Netfind will proceed with a search only if your keys are just right.

The keys describe where the person you are looking for may be found, and THEY typically include information such as the name of the individual's institution or its location (city, state, country) or both. If you know the institution's Internet domain name (cs.colorado.edu, for example) you can use it in your keys by specifying it without the dots (cs colorado edu, for example). You cannot, however, use host names as keywords. If you know of a machine named brazil.cs.colorado.edu, then cs, colorado, and edu might be good keys, but brazil would not.

Using more than one key in a query implies the logical AND of the keys. Specifying too many keys may result in a search that is too narrow, so that Netfind will not find any domains to search. If this happens, try specifying fewer keys.

Specifying too few keys may result in a search that is too broad. If this happens, there will be too many domains to search in a reasonable amount of time and effort, so Netfind will present you with a list of the domains that match your query, and ask you either to try again with different search keys or to select a few domains from the list for further consideration.

After the search has been restricted to just a few Internet domains, Netfind can proceed with the second phase of the search. At this point, it begins to interactively query various hosts in those domains, looking for the person whose name you've supplied. It does this using a variety of techniques, including querying domain name servers (DNS), X.500 directory servers, whois databases, CSO nameservers, Simple Mail Transfer Protocol (SMTP) servers, and by using Finger.

The name that you supply can be a first, last, or login name, but in most cases only one name should be specified. A few of the data sources Netfind consults allow more than one name. If you do wish to specify two names, put them in quotes, like this:

```
"michael schwartz" boulder colorado university
```

In order to help Netfind do a better job of finding people at their sites, some organizations keep information about the White Pages (user directory) services they provide in their Domain Name System (DNS) database. At these sites,

Netfind uses only the directory services listed in the DNS database to carry out its search.

If DNS White Pages information is not available for a site, Netfind switches to a more exploratory search strategy, using DNS, SMTP and Finger. First, DNS is used to identify several well-known hosts within the domain, then SMTP is used to examine mail forwarding relationships on these hosts, and Finger is used to examine login information. If the person you are looking for can be found using any of these techniques, the results will be returned to you.

### **Netfind Tips and Tricks (Searching for Domains, Hosts, . . .)**

Not all of the organizations listed in Netfind's site database are on the Internet. Often, organizations apply for a domain name months or years before their physical Internet connection is active. By keeping information on these sites in its database, Netfind is able to find people at these locations as soon as their Internet connection is complete.

As a result, you can sometimes use Netfind to learn about organizations and the scope of their connection to the Internet. For example, one way (although certainly not the best way) to see if the Internet extends into a particular part of the world is to search there with Netfind.

Suppose we use Netfind to search for "anyone zimbabwe," as shown in Figure 3.17.

#### **Figure 3.17. Using Netfind to explore Zimbabwe.**

This tells us that, at the time I tried this, Zimbabwe was not yet on the Internet. However, the University of Zimbabwe appears to be in the process of joining. When they do come online, their domain name will be uz.zw, and one of the first machines that they are likely to connect is named zimbix.uz.zw.

This is very much the same situation that Hungary was in a few years ago, when Netfind discovered three registered domains in Hungary, none of which were connected to the Internet. Today Hungary is on the Internet, and when I tried it, Netfind discovered over 100 Hungarian Internet domains.

You can use the same technique to see if a prospective employer is on the Net by, for example, searching for anyone xerox. Or, if you hate to be out of touch even when you are on vacation, you can use Netfind to discover Internet connections in the sun by searching for anyone palm beach.

### **Where to Get Netfind Software for UNIX**

You can retrieve the current version of Netfind via anonymous FTP from the University of Colorado:

[ftp.cs.colorado.edu:/pub/cs/distrib/netfind/netfind4.6.tar.Z](ftp://ftp.cs.colorado.edu/pub/cs/distrib/netfind/netfind4.6.tar.Z)

[ftp.cs.colorado.edu:/pub/cs/distrib/netfind/seeddb.tar.Z](ftp://ftp.cs.colorado.edu/pub/cs/distrib/netfind/seeddb.tar.Z)

The first of these files is the Netfind source code. The second is the seed (or site) database. If you choose to run your own copy of Netfind, you will probably want to retrieve a new copy of the seed database every few weeks.



**Note:** Netfind only runs on Sun workstations, running SunOS 4.0 or greater. If you do decide to install your own copy of Netfind, make sure you have enough disk space for the seed database, which is currently about 12MB in size.

---

## **Finger**

Finger provides a very simple way to find information about users on your own system or on a remote system anywhere on the Internet. It can tell you who's currently logged on to the system, and it can give more detailed information about particular users.

To use Finger on a UNIX system, for example, you simply type `finger` followed by a username. To Finger people on a remote system, append an "at" sign (@) and the remote host name to your query, as shown in the Figure 3.18.

### Figure 3.18. Fingering an individual.

To get a list of all the people who are currently logged on to a remote system, use Finger but leave out the username.

If, as in the following example, a query matches multiple users, many systems return only summary information about each user. To get the full information, you must enter a more specific query or use the `-l` option with Finger to ask for a long listing. Examples of both of these uses of Finger are shown in Figure 3.19.

### Figure 3.19. More Finger examples.

## History of Finger

Finger is one of the earliest, and simplest, of network information protocols, in which a user or program on one system (the client) can request information from another system (the server). The name *Finger* is derived from the phone company's old slogan, "Let your fingers do the walking."

By Internet standards, Finger is an old protocol. In fact, its use predates the creation of the Internet by about six years.

Much useful information about the Internet and its predecessor, the ARPAnet, is published in the form of RFCs (Request for Comment documents). The original Finger RFC (RFC 742) was published in December 1977, at which time only three sites used Finger (SAIL, SRI, and MIT). In those days, the main concern people had about Finger was how to promote its use. These days, the main concern is network security.

## Security Issues

Finger's security problems are summed up neatly by this quotation from RFC 1288:



**Warning!!** Finger discloses information about users; moreover, such information may be considered sensitive. Security administrators should make explicit decisions about whether to run Finger and what information should be provided in responses. One existing implementation provides the time the user last logged in, the time he last read mail, whether unread mail was waiting for him, and who the most recent unread mail was from! This makes it possible to track conversations in progress and see where someone's attention was focused. Sites that are information-security conscious should not run Finger without an explicit understanding of how much information they are giving away.

---

A common procedure in many Internet break-in attempts is to compile a list of users on the target system, and then to systematically try to guess their passwords. As a result, one of the most elementary precautions that security-conscious system administrators adopt is to disable the broadcasting of user lists to the Net.

As a result, many sites refuse to answer Finger requests at all. Others will answer requests about specific users but refuse the more general list users form of Finger query.

One feature of the Finger protocol that was useful in the past has now become something of a security liability. This is the capability of relaying Finger requests through an intermediate host. The purpose of this feature was to enable hosts on two separate networks to finger each other through a gateway machine, but it has the unfortunate side effect of enabling crackers to disguise their trail by passing Finger requests through an intermediate system.

The Finger security issues mentioned so far are all of concern to people supplying Finger information, but there are security issues for people who ask for Finger information as well. For one thing, not all Finger clients filter out control characters, making it possible for unsavory individuals to embed escape sequences in their Finger information, messing

up the terminal settings of anyone who fingers them.

There is another potential problem for users of the X11 Windowing System. A few sites respond to Finger requests by using X11 to display graphical output on the system that initiated the Finger query. This is used, for example, when you

```
finger yourhostname:0@drink.csh.rit.edu
```

to draw a pretty picture of a soda machine on your screen. If, however, you set up your X11 security to enable arbitrary remote sites to display output on your screen, then users at those remote sites can do other things to your machine as well, including such tricks as recording all of your keystrokes.

## Finger: How It Works

Like Telnet, FTP, Gopher and the World Wide Web, Finger uses a client/server protocol, in which the client and server pass messages to one another using the Internet's underlying TCP/IP (Transmission Control Protocol/Internet Protocol).

When you finger someone on another machine on the Internet, your Finger client software makes a TCP connection to port 79 of the remote host, sends a one-line request, and waits for a response. At the other end, the remote host's Internet daemon process (inetd) waits for incoming TCP connections. When a request comes in on port 79, it starts up the Finger daemon (fingerd) to deal with it. On some systems, there is no inetd, and the Finger daemon does all the work itself.

The Finger daemon reads the request sent by your Finger client, processes it, and sends back the results, which your Finger client then displays on your screen.

To see how this works, let's consider a specific example, in which you try to finger me on one of our local UNIX hosts. When you type, for example, `finger fogel@willow.usask.ca`, your Finger client makes a connection to willow.usask.ca on TCP port 79, sends the line `fogel<cr><lf>` (the word fogel, followed by a carriage return/linefeed combination), and waits for a response.

You can try this yourself, bypassing the Finger client completely, by using Telnet to send a message directly to a remote Finger daemon, as shown in Figure 3.20.

If your local system does not support the `-l` option with Finger, you may be able to get the same effect by prefacing your query with the two character sequence `/w`. This too is shown in Figure 3.20.

### Figure 3.20: Performing a Finger with Telnet.

## Neat Finger Sites

Finger was intended to provide basic information about a system's users, but people soon learned they could do a lot more with it than that. You can use Finger to access all kinds of information, from current weather conditions to public encryption keys.

Here are a few examples:

Fingering this	Gives you
aurora@xi.uleth.ca	Current northern lights information
forecast@typhoon.atmos.colostate.edu	Tropical storm info
dej@torfree.net	Toronto Freenet information

Scott Yanoff has produced two excellent online guides to Finger resources. Fingerinfo, shown in Figure 3.21, is a UNIX shell script that gives menu-driven access to some of the most interesting Finger sites.

**Figure 3.21. The Fingerinfo main menu.**

Yanoff's other guide, Special Internet Connections, shown in Figure 3.22, lists many different types of interesting Internet resources—Finger, Telnet, e-mail, Gopher, and the World Wide Web.

**Figure 3.22. Special Internet Connections.**

This list is available both as a plain text file (Finger yanoff@alpha2.csd.uwm.edu for instructions on how to get the latest version) and as a World Wide Web page:

<http://www.uwm.edu/Mirror/inet.services.html>

World Wide Web users may be interested to know that there is a gateway from World Wide Web to Finger. As shown in Figure 3.23, the gateway converts ordinary Finger output into hypertext by translating it into the Web's Hypertext Markup Language (HTML).

**Figure 3.23: The gateway from World Wide Web to Finger.****Customizing Your Personal Finger Information**

Most systems that support Finger also give users some control over what Finger says about them.

Some UNIX systems support the `chfn` (change Finger information) command, which enables individual users to modify their own Finger information. The exact nature of the information you can modify with `chfn` varies from system to system. On DEC Ultrix systems, for example, you can change the text that Finger displays for your full name, your office address, and your office and home phone numbers.

On most UNIX systems, if you create a file named `.plan` in your home directory, then the contents of that file will be displayed whenever someone fingers you, in addition to the usual Finger information. If you create a file named `.project`, the first line of that file will be displayed as well.

On some systems, you can do even more exotic things and have Finger run a program every time you are fingered. Some people use this mechanism to provide up-to-the-minute information about themselves, while others use it to keep track of how often they are being fingered.

On UNIX systems that allow it, this is done by turning your `.plan` file into a named pipe, setting up a program that writes to the pipe, and waiting for someone to finger you. Here's how to do it:

First create a pipe named `.plan` in your home directory:

```
% mknod .plan p
```

Then run a program that writes to the pipe:

```
% plan.sh &
```

Here's a sample `plan.sh` script, which will send a different `.plan` file to each of the first five people who finger you:

```
#!/bin/sh
```

```
#
```

```
# sample script that runs a command via finger
```

```
#
```

```
count=0
```

```

object="person has"

date=`/bin/date '+%r on %A %B %d, %Y'`

while ( test $count -lt 5 )

do

count=`expr $count + 1`

echo "$count $object fingered me since $date" > .plan

object="people have"

done

```

Finally, wait for some people to finger you, as shown in Figure 3.24.

### [Figure 3.24: Running a program via Finger.](#)

## GNU Finger

At most sites, the days of the large time-sharing central computer systems are gone, and this reduces the usefulness of Finger. While at one time you could track someone down by fingering one or two large hosts, you may now have to try hundreds or even thousands of smaller machines. And because people often have accounts on more than one system, even when you find them, you may not find the system they use most frequently.

NetFind offers one solution to this problem. GNU Finger offers another.

The GNU (Gnu's Not UNIX) version of Finger is a drop-in replacement for the standard UNIX Finger and fingerd programs, which provides a site-wide (rather than system-wide) Finger service.

At MIT, for example, you can finger gnu.ai.mit.edu to find people on any of fourteen machines in the MIT AI lab.

GNU Finger offers several other extensions to the standard Finger protocol as well, including the ability to include "faces" (bit-mapped images) of users in the Finger output, and the ability to ask for general site information, such as a list of machines that are currently idle.

Users can modify the information GNU Finger displays about them by creating an executable script named .fingerrc in their home directory. If you have a .fingerrc file, GNU Finger filters the normal Finger output through this script before passing it on to the requestor, enabling you to modify, or completely replace, the usual information.

## Where to Get Finger

As mentioned earlier, if you do not have Finger client software, you can use Telnet to send a message directly to a remote Finger daemon instead. To finger user@host.domain, telnet to host.domain, port 79, and type the name of the user you want to Finger, followed by a carriage return and linefeed.

## UNIX Systems

Both the Finger client and server are included with most UNIX systems. GNU Finger is available from many sources, including anonymous FTP, from

```
labrea.stanford.edu:/pub/gnu/finger-1.37.tar.gz
```

## Macintosh Systems

Finger client software for the Macintosh is available from many locations, including `archive.umich.edu:/mac/util/network/finger1.37.sit.hqx`

A Finger client is included as part of Stanford's MacIP package and as part of the Mailstrom mail program.

## PC Systems

Finger client software is included with several commercial TCP/IP packages for the PC, including Sun's PC-NFS.

A Windows Winsock Finger daemon is available from

`sunsite.unc.edu:/pub/micro/pc-stuff/ms-windows/winsock/apps/fingerd.zip`

## Ping

Ping is one of the most basic Internet tools. It checks to see whether another machine on the network is reachable from your own host by sending it a message and waiting for a reply.

The original BSD (Berkeley Standard Distribution) version of Ping is quite terse in its output, as can be seen in the examples shown in Figure 3.25.

### [Figure 3.25. A simple Ping.](#)

Later, and more sophisticated, versions of Ping keep trying until you ask them to stop, and they return more information, such as the packet size, round-trip travel time, and the order in which the return packets arrive (`icmp_seq`), as shown in Figure 3.26.

### [Figure 3.26. A fancier Ping.](#)

## How Ping Works

Ping uses the Internet Control Message Protocol (ICMP) to send messages across the network.

Most of the Internet tools (Telnet, FTP, Finger, Gopher, World Wide Web, and so on) communicate across the network using the Internet's Transmission Control Protocol (TCP). TCP is a complex protocol that breaks messages up into packets, sends them over the network, and puts them back together again at the other end. During transmission, packets may be lost, duplicated, or delayed, and TCP also includes mechanisms for detecting and dealing with these conditions.

Some applications, however, don't require all the complexity of TCP, so the Internet supports two other, simpler protocols: User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP).

Ping requests, for example, always fit into a single packet, and it is no big deal if that packet is lost, because the originating application can always try again a few seconds later if no response is received the first time.

Ping checks to see if another host is reachable by sending it a series of ICMP ECHO\_REQUEST messages, and listening for the responses. All Internet hosts are required to respond to ICMP ECHO\_REQUEST messages, so if you can't get through to a host with Ping, then it is pretty much guaranteed that none of the other Internet applications (such as Telnet) will work either.

## Where to Get Ping

Ping is built into UNIX systems.

On DOS systems, you can get a packet driver version of Ping from:

`omnigate.clarkson.edu:/pub/cutcp/v2.2-E/ping.exe`

Windows users can get a Winsock version of Ping from:

`winftp.cica.indiana.edu:/pub/pc/win3/winsock/ws_ping.zip`

Finally, a Macintosh version of Ping is available from:

`ftp.germany.eu.net:/pub/comp/macintosh/comm/ping-11.hqx.gz`





- [4](#)

- [Where's the Complaint Department? Or, What to Do When Things Don't Work](#)
  - [--by Kevin Mullet](#)
  - [How to Approach Internet Problem-Solving](#)
    - [Good Troubleshooting Practices](#)
  - [What Are Your Expectations?](#)
  - [Has It Ever Worked?](#)
  - [Has It Failed Before? What Fixed It Then?](#)
  - [What Is the Simplest Way of Consistently Reproducing This Problem?](#)
  - [What Pieces Must Go Together for This to Work?](#)
  - [Do I Have Accurate Assumptions about a Possible Cause and Solution?](#)
  - [KISS: Sometimes There's No Grassy Knoll](#)
  - ["Can We Talk?" Or, Getting Clients and Servers to Talk to Each Other](#)
    - [Common E-Mail Problems](#)
    - ["The Check's in the Mail": Trace Your Mail on its Travels](#)
    - [Usenet News](#)
    - [Gopher](#)
    - [The World Wide Web and Associated Browsers](#)
    - ["And the Web Was without Forms and Void—"](#)
    - [ICMP: The Immovable Object Meets the Unreachable Destination](#)
    - [How About Those TCP/IP Settings?](#)
      - [IP Address Problems](#)
      - [The Subnet Mask](#)
      - [Internet Routing](#)
      - [The Default Route or Gateway](#)
      - [The Domain Name Server](#)
  - [A Word About Hardware](#)
  - [The Graceful Art of Finger-Pointing](#)
    - [How Do You Get There from Here?](#)
    - [Who Supports What?](#)
    - [Regional Contact References](#)
    - [Getting Down to "Brass Tacts"](#)
  - [Getting Around the Problem](#)
  - [Summary](#)

# 4

## Where's the Complaint Department? Or, What to Do When Things Don't Work

--by Kevin Mullet

### How to Approach Internet Problem-Solving

"Why on earth," you may ask, "should I learn anything about Internet troubleshooting?" Simple. No matter how good your network support structure is, a bit of self-reliance is always a good thing on the Internet. Naturally, unless your job is computer or network support, you've got other ways to spend your time than peering into the abyss of network protocol analysis and service availability. Just as you're better off knowing how to change the oil and fill up the tank in your automobile, some knowledge about how things talk to each other on the Net and what it's like when they go down goes a long way.

This information is directed at everyday Internet users who don't have an arsenal of network management tools at their beck and call, but still want to be able to find out more about why things on the Internet sometimes don't work—and if there's anything they can do to fix it, or to at least make problem resolution easier for those who can.

### Good Troubleshooting Practices

The most important ideas in this chapter have nothing to do with the Internet. The essentials of troubleshooting are valuable to just about every discipline. It's surprising how many professional troubleshooters regard systematic troubleshooting as an unnecessary chore once they acquire a bit of confidence and experience. Ordinary Internet users armed with less experience but a better troubleshooting practice can stand toe-to-toe with the experts in their ability to diagnose a majority of Internet service availability problems.

Each time you encounter a problem on the Internet, it helps to run through a few essential questions to get a clear idea of exactly what the problem is.

### What Are Your Expectations?

It's quite common to begin troubleshooting with nothing more substantial than a general feeling of non-productiveness, and no specific ideas about what your eventual expectations are from your troubleshooting efforts. The first order of business when troubleshooting is to boil your troubles down to a simple statement of what you want that you're not getting, and what requirements must be met by your troubleshooting efforts. A poor problem definition would be "Telnet is broke" followed by an equally poor resolution requirement of "Fix it." Something along the lines of "When I try to telnet to zig.foo from zag.foo, I connect but never get a login prompt" and "Make sure I get a login prompt when I telnet to zag.foo" leads to a much more self-contained troubleshooting agenda than something abysmal like "The system is down; make it work."

## Has It Ever Worked?

Sometimes system performance expectations are more a function of urban legends than actual experience. Some of the world's most humble software can be imbued with the most amazing characteristics. It's worthwhile to separate the wheat from the chaff and determine if the task before you can actually be done. Does telnet really have a video-conferencing mode?

## Has It Failed Before? What Fixed It Then?

Most troubleshooting would pretty much be a waste of time unless we retained the information we learned. Keep records of what fails and when, and how it was fixed. Most problems aren't unique—they've been solved before. Finding an earlier solution that works for you is usually preferable to finding a new one. The extra time it takes to keep adequate records is almost certainly less time that it would take to repeat the entire troubleshooting process the next time the problem happens.

## What Is the Simplest Way of Consistently Reproducing This Problem?

The first thing you'll need to do if you or someone else tries to fix a problem is to reproduce it. Problem reproduction is necessary both for further characterization and for testing possible solutions. Unless it would be damaging to do so, find a way to repeat the problem, then remove variables until you have the simplest circumstances under which you can reproduce the problem.

For instance, if you use an MS-Windows client to send e-mail with MIME attachments to numerous recipients and it always bounces back, you might start by closing out the other applications you're running at the same time. You can then reduce the number of recipients to one and see if it still bounces, maybe trying a different recipient as well. You can also try sending a very simple message with no MIME attachments and see if that makes a difference.

A good way of finding the minimum level of complexity that still exhibits your problem is the "binary search," so-called because you're forever splitting the number of possibilities in two.

Let's say you've a 10BaseT hub with thirty computers attached to it, and one of them is using the same IP address as the mainframe that's on the same subnet—which means that it's practically impossible to get work done on the mainframe. This condition has the curious side-effect of putting all your VM programmers in a really foul mood. You've got to find the offending computer and reconfigure it to have a unique IP address. Using a binary search, you start by taking half the users off the hub and see if the problem is still there. If it's not, that means the problem is probably with one of the 15 computers you just pulled off the Net. Next, you put half of the disconnected computers back on the Net. If the problem shows up again, that means that one of the seven or eight computers you just replaced is the one you're after. You follow this procedure a few more times until you narrow the possible choices to the machine belonging to the individual with the unruly mob gathering outside his or her office—at which point you fix the problem and blame it on whoever was the most recent person to quit.

## What Pieces Must Go Together for This to Work?

An understanding of what talks what on the path from your computer or terminal to the end service can be as invaluable as a good map on a road trip. Troubleshooting usually lends itself well to one of two approaches: a *binary search* or a *linear search*. Linear searches, in which you begin at one end of the route and verify components in a line until you get to the other end, require you to know at any given point on the network what the next item is on the path to the other end.

Regional IP routing problems often lend themselves well to a linear approach—especially when there are dissimilar mediums between routers. If, for instance, you use a dial-up facility with a terminal server in one city that connects via the Internet to hosts in another city—going through numerous local Ethernets, T1 lines, and a microwave link—it would be difficult to troubleshoot connectivity or service quality problems without the knowledge of where the traffic went when it traveled from the terminal server to the hosts. Frequently, there is a method for finding such information dynamically on an IP network. We'll get to that a little later on.

## **Do I Have Accurate Assumptions about a Possible Cause and Solution?**

An inexperienced troubleshooter often has an important advantage over more experienced ones: modesty. A humble troubleshooter is less likely to make rash assumptions about the state of both the Net and their own capabilities.

Accurate assumptions about causes and solutions should be based on information such as what kind of error messages coincided with the problem, whether the service has worked since any related system changes or upgrades, and whether anyone else using the same resource is having the same problem.

## **KISS: Sometimes There's No Grassy Knoll**

One of the axioms of troubleshooting is known as KISS: Keep It Simple Stupid. It's an important component to any problem solving technique. If, once you eliminate the impossible, there are numerous conceivable causes for a particular problem, it's more likely to be the uncomplicated, obvious choice than the one involving international conspiracy and a gunman on a grassy knoll.

## **"Can We Talk?" Or, Getting Clients and Servers to Talk to Each Other**

To find the source of a problem, you occasionally need to look no further than your local application or operating system. In a way, problems like this are much better than errors out on some remote weigh station on the information highway—there's a greater possibility you can fix these and move on.

Sometimes, though, the problem is with the server for whatever service you're trying to access. When there's a problem with a server, it is frequently easy to narrow the number of possible causes, but the responsibility for resolving the problem may ultimately fall to the local MIS or Computing Center personnel.

The sheer size and complexity of the Internet demands that there's always something down somewhere. This isn't to say that the Internet should be written off as unreliable, but that the system is so immense that even if a tiny percentage of it is down at any given time, it might just be that tiny percentage that's critical to something you want to do right now. Troubleshooting end-to-end connectivity isn't rocket science, though, and there are a few easy steps that, if they won't fix the problem, will at least give you more information about where the cause is and who to talk to about it.

In the spirit of keeping it simple, it's usually a good idea to check a few rudimentary areas before you drag out the big guns:

Has your software been able to do what you want it to do since the most recent configuration change in the application, network driver, or operating system?

Did you specify the correct path?

For PC users, and to a lesser extent for Mac users, do you have the correct network drivers loaded?

Take a couple of minutes to walk through the process you're trying to accomplish and ask yourself why you're

doing each step. The oddest things can look acceptable if you're tired, hurried, or otherwise stressed. Lots of things can be quickly resolved with a slow, sane walk-through. Although we don't often like to admit it, sometimes the problem really is the nut behind the keyboard.

## Common E-Mail Problems

After eliminating rather obvious things like availability of local disk space for temporary files and confirming that the configuration for whatever e-mail package you're using is actually being used, there are a few problems that seem to be endemic to e-mail packages in general.

Many e-mail packages typically use one or two of three main e-mail protocols: Internet Message Access Protocol (IMAP), Post Office Protocol (POP), and Simple Message Transport Protocol (SMTP). IMAP and POP retrieve and manipulate messages on a remote mail server. SMTP is used for sending Internet mail. Of the e-mail clients that don't use these protocols, many of them use proprietary methods for sending, manipulating, and retrieving messages from a central mail server. Such a proprietary mail server, however, frequently either supports SMTP or communicates with a server that does if it's Internet accessible.

It's important that any IMAP-, POP- or SMTP-compatible e-mail client be configured to use the appropriate servers. The first thing you'll need to do to ensure you're pointed at the appropriate servers is to check your application and see which standards, if any, it adheres to. Just because you can send Internet mail doesn't mean you necessarily use SMTP. You might be using a proprietary e-mail client that all but precludes standardized protocol debugging. Let's assume you're not, though, and that you've determined by either asking your neighborhood guru or (if you are the guru) by reading the documentation on what protocol your e-mail client uses, and that it's some combination of SMTP with either POP or IMAP.

To test basic connectivity to an SMTP client, you can telnet to it at tcp port 25, (see the following example) and issue a noop (no operation or null operation) command. If this works, you will have confirmed two things:

End-to-end connectivity with your SMTP host.

Your SMTP server is "listening" for SMTP connections, and at least has enough resources to start receiving SMTP requests.

You will not have confirmed:

That you are talking to the correct SMTP server. That's an administrative decision, not a technical one.

That the SMTP server is able to process much more than the null command you sent it.

Generally, if you can connect at all with a server in this manner, it's running fine. E-mail server failures tend to be catastrophic and, in such situations, the solution is usually just to restart the server software.

Here's a basic telnet test to the SMTP port. The example is shown from a UNIX csh prompt, but it can just as easily be done from any telnet client that lets you telnet to tcp ports other than 23, the one officially assigned to the Telnet application. You would type what's in **bold**, and replace your own server name with the one in the example:

```
%telnet foo.any.net 25
```

```
Trying 127.1.2.3 ...
```

```
Connected to foo.any.net.
```

```
Escape character is '^]'
```

220 foo.any.net SMTP ready at Mon, 1 Aug 94 10:41:43 CDT.

noop

200 OK

quit

221 foo.any.net closing connection

Connection closed by foreign host.

%

If your client uses POP, you can do an equivalent test with tcp port 109 of your POP server to establish similar connectivity assumptions:

```
%telnet pop.bogus.net 109
```

Trying 127.1.2.3 ...

Connected to pop.bogus.net.

Escape character is '^]'.  
+ POP2 pop Server (Version 1.001 February, 1989)

```
quit
```

```
quit
```

```
+ Bye POP2 pop Server (Version 1.001) shutdown
```

Connection closed by foreign host.

%

As listed in RFC937, POP2 uses a default port of tcp/109; as listed in RFC1081, POP3 uses a default port of tcp/110.

And finally, here's a test you can do with port tcp/143 of your IMAP server to establish basic connectivity. The part you type may look a little strange to you, but IMAP requires that incoming commands be serialized; that's why there's an arbitrary code in front of each command you type.

```
%telnet imap.bogus.net 143
```

Trying 127.1.2.3 ...

Connected to imap.bogus.net.

Escape character is '^]'.  
\* OK imap.bogus.net IMAP2bis Service 7.0(52) at Mon, 1 Aug 1994

```
A1 noop
```

```
A1 OK NOOP completed
```

```
A2 logout
```

```
* BYE imap.bogus.net IMAP2bis server terminating connection
```

```
A2 OK LOGOUT completed
```

```
Connection closed by foreign host.
```

```
%
```

If all goes well, you will get replies from your SMTP, POP, or IMAP server similar to the preceding ones. If not, that's also good because it gives you information you can use to solve your problem. It's important to remember that there's no such thing as a failed experiment—only one that didn't necessarily go as planned. Here are some things that may have happened in the preceding tests and what they may indicate:

```
%telnet server.bogus.net 25
```

```
Trying 127.1.2.3 ...
```

```
telnet: connect: Connection timed out
```

```
telnet>
```

There are a variety of error messages that amount to "you can't get there from here." This may indicate loss of end-to-end connectivity between the client and the server. More about that later.

```
%telnet speigal.hou.tx.us
```

```
speigal.hou.tx.us: unknown host
```

```
telnet>
```

An unknown host message may indicate a couple of things:

- If it truly is an unknown host, you may just have the wrong name. One way to check is to use the IP address to connect to it. If that fails as well, and depending on just how it failed, you may have end-to-end connectivity problems.
- If you know you have the correct name, and you are able to connect successfully to the IP address but not to the "Fully-Qualified Domain Name" or FQDN, you may have a problem with your Domain Name Service. DNS is the system that associates FQDNs that are meaningful to people, and IP addresses that are meaningful to computers. Before DNS, host managers had to keep local host files with a list of all hosts and all IP addresses on each host. With over 20 million users now on the Internet, the existence of DNS is great news for system managers—probably lukewarm news for disk manufacturers.

If your connection is refused:

```
%telnet server.bogus.net 25
```

```
Trying 127.5.4.3 ...
```

```
telnet: connect: Connection refused
```

```
telnet>
```

it means that the machine you want to talk to is there, but it isn't accepting connections for the service you want. This can be caused by numerous things, including:

- You're trying to talk to the wrong machine—another machine is actually the server you want. Double-check your information.
- The server software on the machine isn't working correctly. See if you can get whoever runs the machine to take a look at it.
- There's a security gotcha in place. Talk to the folks responsible for network security of whatever network you're using and see if the particular tcp service port you're trying to use has been blocked between your source machine and the destination server. This is appropriate as a last resort, in case nothing else pans out. If there is a security block in place, it's probably there for a good reason.

## "The Check's in the Mail": Trace Your Mail on its Travels

If you have established that you have basic e-mail connectivity, and you're using an e-mail program that lets you add your own user-defined header lines, you can attempt to send mail with a return receipt.

If you don't get a receipt, it doesn't mean you have a problem—it may mean that one or more of the mail servers your mail has to travel through to get to its destination doesn't support return receipts. Return receipts are great if you can get them, but they don't mean much if you don't—unless you're previously confirmed that a given mail server does support receipts.

The method or possibility of adding a header line to a mail message varies from client-to-client, but essentially, you should add a line like this to your message, replacing the e-mail address with your own.

```
Return-Receipt-To: cuss@blankety.blank.com
```

After you send such a message, a mail server that supports return receipts on the e-mail route between you and your addressee will send you a separate message informing you that it received your message.

## Usenet News

Making sure you're talking to the right Usenet news server isn't all that much different than confirming basic e-mail connectivity. The transport protocol for most Usenet news, Network News Transport Protocol (NNTP), is a TCP-based service, as are SMTP, POP, and IMAP. As you may suspect, this means that there's a simple telnet test to make sure that there's end-to-end connectivity to the server to which you believe you're supposed to connect to with your news reader.

After you've ruled out obvious problems like a configuration file in the wrong subdirectory or lack of local disk space, you may want to try the following test on your designated news server's NNTP port, tcp/119.



**Note:** If you use UUCP, IP Multicasting, or another method to get your news, this method won't do you any good.

---

```
%telnet news.clench.org 119
```

```
Trying 127.1.5.5 ...
```

```
Connected to news.clench.org.
```

```
Escape character is '^]'.  
200 news NNTP server ENEWS 1.4 22-Dec-93 ready (posting ok).  
quit  
205  
Connection closed by foreign host.  
%
```

Aside from the basic connectivity issues covered earlier, you can also determine if you are permitted to originate or "post" new Usenet articles through this server. On the greeting line that begins 200 news NNTP server ENEWS 1.4, look at the end of the line. You'll see a statement of whether you can post or not. If an NNTP server doesn't permit you to post, you may either be using the wrong server, or have essentially read-only access to Usenet.

## Gopher

Gopher is another TCP-based service, which means (as you may suspect) you can give it a cursory check by telnetting to its service port. Although it's not a hard-and-fast rule, about half of all Gophers may be found on TCP port 70. If a Gopher doesn't respond on port 70, it's not necessarily down—it may be installed on a different TCP port. This test, therefore, can reliably confirm the health of a Gopher server, but it can't reliably confirm that Gopher server is down *unless* you already know the Gopher is installed on a particular port and isn't responding.

As you can see from the example, this is a pretty straightforward test. You should connect to the server and get no welcome message or other output until you press Return. Once you hit RETURN, you'll get a listing of the top-level directory entries for that particular Gopher server, followed by a period (.) on a line by itself before the connection is closed from the remote end.

```
%telnet rs.internic.net 70
```

```
Trying 198.41.0.5 ...
```

```
Connected to rs.internic.net.
```

```
Escape character is '^]'.  
(press RETURN here)
```

```
1Information about the InterNIC 1/aboutinternic is.internic.net 70
```

```
1InterNIC Information Services 1/infoguide is.internic.net 70
```

```
1InterNIC Registration Services 1 /rs rs.internic.net 70
```

```
1InterNIC Directory Services 1/.ds gopher.internic.net 70
```

```
.  
Connection closed by foreign host.
```

```
%
```

If the Gopher doesn't connect at the appropriate port, it may have lost its ability to listen for new connections. The system administrator for the Gopher machine can probably fix this using a solution from the Gopher FAQ at [URL](#)

```
gopher://mudhoney.micro.umn.edu/00/Gopher.FAQ
```

## The World Wide Web and Associated Browsers

The telnet test for a WWW server (actually, a HyperText Transport Protocol or HTTP server) involves connecting to TCP port 80 and doing a simple command to verify operation. Like their cousins the Gophers, HTTP servers may also be installed on alternate service ports, so be sure to double-check which port your HTTP server ought to be using before you begin testing.

The server can be tested by telnetting to the HTTP service port of a particular machine, and requesting the root document with a GET / command. The server should spit out a document writing in the hypertext markup language (HTML). If it doesn't, this doesn't necessarily mean the server doesn't work—it just means there isn't a root document. To test servers without a root document, you have to know the path to a document available on the server and give that instead of /.

```
%telnet info.cern.ch 80
```

```
Trying 128.141.201.214 ...
```

```
Connected to info.cern.ch.
```

```
Escape character is '^]'.  
GET /
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>World-Wide Web Home</TITLE>
```

```
(lots of text deleted)
```

&lt;/BODY&gt;

&lt;/HTML&gt;

Connection closed by foreign host.

%

WWW Browsers are unique critters in that they not only serve up access to their own native protocol, HTTP, but they provide access to other information-server technologies as well, including FTP, Gopher, WAIS, and numerous others. You may start out troubleshooting a WWW service failure but soon realize that it is actually an FTP problem.

As a matter of fact, FTP is occasionally problematic when used with file:// or ftp:// URLs in World Wide Web browsers. Web browsers do an anonymous FTP login before and logout after every operation requested by the person using the browser, whether it's a directory listing or a single file retrieval. For all but the most resource-poor or busiest FTP sites, this presents no problem. It's likely, however, that the busiest sites will also be frequented by Web-browser users. The problem lies in the fact that many FTP sites place a ceiling on the number of users that can be doing anonymous FTP at one time. Any users attempting to connect after such a threshold has been reached will be refused access. Such sites frequently stay at their ceiling constantly during peak periods of usage. During such times, Web browsers may perform poorly as FTP clients, and you would be well-advised to use a conventional FTP client instead.

## **"And the Web Was without Forms and Void—"**

Well, it's not really that bad, but surfing the World Wide Web these days without a browser capable of HTML+ forms is like having a television with no cable—quaint idea, but you're really missing a lot. Many of the most interesting and cutting-edge things on the Internet are implemented with HTML+ Forms front ends that deliver everything from databases or flowers to your doorstep to prerecorded video. If you find that Web pages that are supposed to have forms don't display the fields, or display unintelligible text, you may want to upgrade your client to one that can handle forms.

## **ICMP: The Immovable Object Meets the Unreachable Destination**

Internet Control Message Protocol. Yeah, right. But wait! It's really not all that complex. ICMP is not some kind of rocket science network mysticism that requires a pointy hat, magic wand, and a membership card in a space-cadet club to figure out. Briefly stated, ICMP is a protocol for explaining why you can't do what you want to on the Net, and a little bit about why. ICMP doesn't cover all possible problems—far from it—but it does describe a great deal of the problems you can encounter that involve failed end-to-end connectivity.

The reason you would want to know anything at all about ICMP messages is that the language of ICMP errors is usually the same language in which the applications that receive them tell you about network problems. When your Web browser, telnet app, or e-mail program comes back with a message like destination unreachable, host unknown, or port unreachable, it's quoting chapter and verse from ICMP. The first step to understanding the nuances of Internet end-to-end connectivity is a familiarity with ICMP and how it relates to the Net.

The ICMP errors you may want to be familiar with are network unreachable, host unreachable, port unreachable, destination network unknown, destination host unknown, and a handful of errors reflecting an administrative prohibition to a particular host, network, or service.

A network unreachable message means that the network may exist, but a critical link between two or more sites

is down on the route between you and your destination. The reason for this is usually a failure of the communications equipment that provides the link between the two sites, or a circumstance known in Net parlance as a *route flap*, in which the physical route may be perfectly healthy but the configurations of two or more pieces of communications equipment may be locked in a type of deadly embrace, each fighting for its own view of how the network link ought to be configured. Legitimate physical and electronic problems, of course, can also cause route flaps—but it's also likely that the routing-layer flap may be caused by failure of adjacent Internet providers to adequately work together to provide optimum connectivity. The hallmark of a configuration-caused route flap is one that goes up and down with a rhythmic regularity. If you test connectivity to a remote site and you notice that, without fail, it goes down for 45 seconds, comes up for 45 seconds, goes back down for 45 seconds ad nauseam, hang on, because you might be riding the roller-coaster of an interregional border dispute.

A host unreachable message happens when the route between your site and the destination site is in working order, but the ultimate destination for your service is down. This usually occurs when a computer is either down or partially disabled due to a difficulty specific to a small number of hosts.

A port unreachable error indicates that while there is successful routing between source and destination sites, and while the hosts on each end can talk to each other, the service you want from the remote host is not available. This doesn't necessarily mean that it shouldn't be available. Sometimes the software providing a particular service can "give up the ghost" without affecting other services on the machine. Actually, this is one of the most frequent critical errors you may experience.

Destination network unknown and destination host unknown messages happen when information about the location of your destination network or host has expired, or was never there to begin with. The reason for this may be that the host or network was unreachable for so long—that its route expired in the constituent routers that provide a path between you and the destination. It may also be the case that the network or host name or address is just wrong.

Administrative prohibitions may be put in place to intentionally prevent access to services (ports), hosts, or entire networks. In fact, the more commercialized and less provincial the Internet becomes, the more valuables there are to protect. It stands to reason, therefore, that the number of people in this global community installing locks on their front doors will only increase over time.

## **How About Those TCP/IP Settings?**

Often, network problems can be solved as easily as correcting the local network client configuration. TCP/IP, the protocol suite that forms the basis for the global Internet, requires that each client either be preconfigured or remotely configured with a number of parameters that are critical to the healthy operation of any host.

## **IP Address Problems**

If one machine is configured with the same IP address as another machine on the same network, neither of the machines will be able to function on the network while both are running, but both will be able to run just fine individually—that is, if enough time has elapsed since the appearance of the other host that its address resolution entry has expired from the local hosts and routers. This problem is usually resolved by changing the IP address of the host on which it is most convenient to do so.

Often, if a host is moved from one location to another, it isn't reconfigured to have a new address appropriate for the new location. In this situation, the host will not function on the network at all. If a host—particularly a PC or a Mac, on which it is easy to overlook such things—ceases to function after it's been moved to a new location, you may want to check the IP address.

Another clue that a wrong IP address is the problem is if your machine is using more than one network protocol

"stack" and IP is the only thing that isn't working. If you can talk to DECNET or Novell just fine, but the Internet stops working for you after a move, this is almost certainly the problem.

## The Subnet Mask

The subnet mask tells your Internet software how much of your IP address describes which computer you are on your local-area network, and how much of the address describes which network you are attached to. For instance, a subnet mask of 255.255.255.0 (sometimes stated in C-language hexadecimal notation as 0xfffff00) on an address of 192.136.150.6 means that the computer is number six on a network that can be described as 192.136.150.0. If a computer is set up with an inappropriate subnet mask, that means that it will probably be able to talk to other computers on its local-area network, but that it might not be able to talk to hosts outside of it. To understand why, we need to take a brief pit stop and look at something called the IP Routing Algorithm.

## Internet Routing

With so many millions of hosts on the Internet, how does your host know whether the destination for a particular service is the computer in the next office or a mainframe in Canada? To a large extent, this kind of global localism is accomplished because TCP/IP software follows a standard set of routing "rules of the road." For any given IP address that it would contact:

- Is it visible on the local-area network to which it is attached? If so, route established—if not, go to next step.
- Is the route recorded locally, perhaps in a local host's file? If so, route established—if not, try the next step.
- Has the route been learned about through a dynamic routing broadcast protocol? If so, great—if not, we've still got one more thing to try.
- Has the software been configured with a *default route* or *default gateway*? If so, forget about trying to find out where the destination is and let the default gateway try this whole business. If there isn't a default gateway, on to the next step.
- Sorry, you've lost the routing game. destination host unknown or destination network unknown.

A lot of Internet software uses the subnet mask to help determine if a destination host is on the same network as it is. For instance, if the source host had the address of 198.64.55.100 and the destination host had the address of 198.64.56.101, it's a good bet that if they're using a subnet mask of 255.255.0.0, that they're on the same local network. If, however, they're using a subnet mask of 255.255.255.0, that means that one computer is on subnet 198.64.55 and the other one is on subnet 198.64.56—two different networks. In the first example, though, each computer is considered to be on the same network: network 198.64.0.0.

## The Default Route or Gateway

As we've seen earlier, the default route or gateway is critical to proper connection with hosts beyond your local-area network. If you can't talk to any computer outside your own local-area network, and you're using the correct subnet mask, your software might be pointing at the wrong gateway.

A feature of some TCP/IP software is that it requires the network portion of your gateway address to be the same as the network portion of your own IP address. This may seem to be an obvious thing to assume but, as in the

earlier example, computers 198.64.55.100 and 198.64.56.101 may actually be on the same local network, even if they have a subnet mask of 255.255.255.0. This may happen if two networks that are previously separate are consolidated, or if there are other reasons you may want to clearly distinguish the address of two or more groups of computers on a local-area network from each other.

In this situation, such software may require that the 198.64.56.101 computer use a gateway address beginning with 198.64.56, and that the 198.64.55.100 computer have a gateway address beginning with 198.64.55. There are two ways around this problem. One is to use different TCP/IP software on that network (often a support headache) and the other is to see if the caretaker of the local routers can add a supplementary address to the gateway you need to use. In this certain limited circumstance, one computer (the router) can respond to packets on the network with one of two or more IP addresses, even though packets that it sends out in return will only use the main IP address.

This is one of those lady-or-the-tiger choices: either a headache for the networking support people, who have to keep track of multiple addresses for a given gateway router on a single local-area network, or a headache for the software support folks, who may have to support an additional TCP/IP package for a particular network.

## The Domain Name Server

If all these numbers are confusing, don't worry—you should rarely have to deal with them except when troubleshooting. The *domain name server* is a protocol for associating more memorable names with the IP address numbers necessary for routing between hosts.

If DNS is working, you should be able to contact hosts by their proper names like rs.internic.net. If not, you may have to resort to using their IP addresses, such as 198.41.0.5. If, in fact, you are able to talk to a destination host by using the IP address but not the fully-qualified domain name, it's a good indicator that your software isn't configured with a correct domain name server, or the domain name server itself is nonfunctional.

A good way to test this if you're on a UNIX host is to use nslookup to compare what your host *thinks* is the correct resolution of a given address with the actual source of authority for that network.

To check what IP address the host brigadoon.hou.tx.us resolves to, for instance, try this from the UNIX shell prompt. You may have to specify the full path for nslookup. It's often kept in /usr/etc.

```
%nslookup brigadoon.hou.tx.us
```

```
Server: is.rice.edu
```

```
Address: 128.42.42.24
```

```
Non-authoritative answer:
```

```
Name: brigadoon.hou.tx.us
```

```
Address: 198.64.55.98
```

```
%
```

This exchange can tell you a couple of things. First, that the host uses domain name server is.rice.edu to get the information from the nslookup. That server says that it has a non-authoritative answer to the query, which is that brigadoon's IP address is 198.64.55.98. "Non-authoritative" refers to the fact that the server supplied the information from cache instead of getting it from the authoritative source. The server will not actually go to the

original source to get the answer to a DNS query unless it can't find it in cache, or the time to live on the cache entry has expired.

Here's an example. If nslookup queries the domain name server for something that isn't likely to be in its cache, it goes out to the authority for the entry (in this case, the source of authority for UNT.EDU) and returns an authoritative entry. The second time the same name server is queried for the same information, a nonauthoritative entry is returned from the local cache.

```
> nslookup loher.dialup.unt.edu
```

```
Server:  is.rice.edu
```

```
Address: 128.42.42.24
```

```
Name:    loher.dialup.unt.edu
```

```
Address: 129.120.35.13
```

```
> nslookup loher.dialup.unt.edu
```

```
Server:  is.rice.edu
```

```
Address: 128.42.42.24
```

```
Non-authoritative answer:
```

```
Name:    loher.dialup.unt.edu
```

```
Address: 129.120.35.13
```

```
>
```

Because of the tendency of DNS entries to not be updated until they expire, they are sometimes out-of-date with respect to changes at a remote location. If, soon after that DNS record was cached by is.rice.edu, that remote network was reconfigured and loher.dialup.unt.edu was given a new IP address, you would be unable to reach it by its domain name until your local cache entry expired some hours later.

There is a method for using nslookup for finding the formal source of authority for any given fully-qualified domain name. It involves setting the query type to SOA (source of authority), and querying successively smaller portions of the address until you hit on the site's domain name (like hou.tx.us as opposed to gotham.hou.tx.us). After you get a source of authority record, you can point nslookup at that server and get an authoritative reply. See the following example:

```
%nslookup
```

```
Default Server:  is.rice.edu
```

```
Address: 128.42.42.24
```

```
> narnia.hou.tx.us
```

```
Server:  is.rice.edu
```

Address: 128.42.42.24

Non-authoritative answer:

Name: narnia.hou.tx.us

Address: 198.64.55.97

> set type=soa

> narnia.hou.tx.us

Server: is.rice.edu

Address: 128.42.42.24

\*\*\* No start of authority zone information is available for narnia.hou.tx.us

> hou.tx.us

Server: is.rice.edu

Address: 128.42.42.24

Non-authoritative answer:

hou.tx.us origin = sesqui.net

mail addr = hostmaster.sesqui.net

serial=19930236, refresh=10800, retry=900, expire=3600000, min=86400

Authoritative answers can be found from:

ns.sesqui.net inet address = 128.241.0.84

moe.rice.edu inet address = 128.42.5.4

> server ns.sesqui.net

Default Server: ns.sesqui.net

Address: 128.241.0.84

> set type=a

> narnia.hou.tx.us

Server: ns.sesqui.net

Address: 128.241.0.84

Name: narnia.hou.tx.us

Address: 198.64.55.97

>

%

Here's what the user did in the preceding example:

- Run nslookup.
- Request an IP address for the host narnia.hou.tx.us.
- Receive a nonauthoritative answer from is.rice.edu, the default name server for the current host.
- Tell nslookup to switch to requesting source-of-authority records instead of the default address (a) type records.
- Request a source of authority for narnia.hou.tx.us. That fails because narnia is an individual host, not a domain.
- Try requesting a source of authority record for the domain hou.tx.us, which succeeds, because hou.tx.us is a full domain. If that failed, a good next try would have been tx.us ,and so on.
- The copious SOA reply lists two authoritative sources of DNS for the domain hou.tx.us: ns.sesqui.net and moe.rice.edu.
- Direct nslookup to use the domain name server ns.sesqui.net.
- Change the query type back to address (type A) from source of authority (type SOA).
- Ask for the address of narnia.hou.tx.us.
- All done. Wasn't that fun?

You now know that the IP address for the host narnia.hou.tx.us in your local DNS cache agrees with the actual entry out on the authoritative source for that domain, ns.sesqui.net.

## A Word About Hardware

Believe it or not, faulty local hardware can sometimes prevent you from cruising the Internet. Nailing down hardware problems can get a bit hairy. Usually, you're doing good if you can isolate the problem far enough to know who to call. It's important when you start looking at the hardware to have fairly confident knowledge of just what is plugging into what.

In this day of "brouters" and Ethernet switches, it's less easy to pigeon-hole your work-a-day network box than it used to be, but three of the basic categories are still worth knowing: *repeaters*, *bridges*, and *routers*.

Repeaters are probably the most humble network box. They have two or more network interfaces and simply regenerate the electrical signal from any given interface to all the others—warts and all. Repeaters are usually used to increase the legal limit on length or number of nodes possible on a local-area network.

Bridges offer a bit more protection from propagating electrical problems than do repeaters. A bridge "understands" the lowest-layer "Media Access Control" packets on the network, albeit Ethernet, token-ring, or whatever. A bridge will not ordinarily permit electrical signal problems to spread between its two or more network interfaces. What it will permit, though, is for higher-level, packet-storm type problems to spread. To ensure against this type of problem, a more complex type of machine is needed: a router.

Routers "understand" the higher-layer "network" protocols such as IP, IPX, DECNET, and so on. The greater degree of complexity affords a greater range of success—a router will both permit you to set up global networks that are all insulated from each other, and give you more than enough rope with which to hang yourself.

Each level of complexity afforded by repeaters, bridges, and routers represents an increase in security, but an increase in price and complexity of skill set required to maintain them as well. For this and other reasons, there are few tools available to diagnose and troubleshoot repeaters, only a handful more available to troubleshoot bridges, and an entire industry that earns its livelihood from router troubleshooting tools.

If your Internet problems are shared by other persons in the same office or building, you might want to consider the possibility that some hardware may be at fault. Do you share an Ethernet segment with the other persons having the problem? Do you share a repeater or bridge?

Later in this chapter, a few methods are presented for finding problems on specific routers on complex IP-based wide-area networks like the Internet, where we all share routers varying in complexity from a UNIX host running routed or gated, to custom-designed IBM Nodal Switching Systems.

## The Graceful Art of Finger-Pointing

By far, the best way to assure good quality and consistency in your Internet experience is to look beyond the products of each FTP retrieval or Web surf to the *process* of using the Internet. If you look at the big picture—all the pieces that come into play when you use the Internet—it is much easier to deal with the inconsistencies and other problems that will inevitably haunt your online sessions when you can least afford them. By tuning the process instead of patching or reworking the result, you will not only increase your own expertise, but create an environment that produces less problems and is much more enjoyable to use.

## How Do You Get There from Here?

Eventually, you'll encounter an Internet problem that's caused by loss of service somewhere beyond your local site. Two invaluable tools for tracking down this kind of problem are PING and Traceroute.

PING performs the simplest of network tests. It bounces a packet against the remote site and waits for it to return.

Command-line options on different versions of PING permit you to expand the test to note the total route trip time in milliseconds, or to do a continuous stream of PINGs, noting the amount of loss, and more.

In its simplest form, a PING test would be performed like this:

```
%ping mit.edu
```

```
mit.edu is alive
```

```
%
```

A more complex test could be used to get some information about the quality of a connection out to a particular site. The following test tells PING to send five packets of 100 bytes to a remote host and note the route trip time.

```
%ping -s mit.edu 100 5
```

```
PING mit.edu: 100 data bytes
```

```
108 bytes from MIT.MIT.EDU (18.72.2.1): icmp_seq=0. time=53. ms
```

```
108 bytes from MIT.MIT.EDU (18.72.2.1): icmp_seq=1. time=51. ms
```

```
108 bytes from MIT.MIT.EDU (18.72.2.1): icmp_seq=2. time=54. ms
```

```
108 bytes from MIT.MIT.EDU (18.72.2.1): icmp_seq=3. time=51. ms
```

```
108 bytes from MIT.MIT.EDU (18.72.2.1): icmp_seq=4. time=53. ms
```

```
----MIT.MIT.EDU PING Statistics----
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip (ms) min/avg/max = 51/52/54
```

```
%
```

Traceroute works in a similar manner, but it also lists each of the routers between your local site and the remote side:

```
%traceroute mit.edu
```

```
traceroute to mit.edu (18.72.2.1), 30 hops max, 40 byte packets
```

```
1 bruno-p5.rice.edu (128.42.42.1) 2 ms 3 ms 2 ms
```

```
2 verge.rice.edu (128.42.209.42) 4 ms 4 ms 2 ms
```

```
3 ENSS-F0.SESQUI.NET (192.67.13.94) 3 ms 3 ms 4 ms
```

```
4 t3-3.cnss64.Houston.t3.ans.net (140.222.64.4) 6 ms 5 ms 2 ms
```

```
5 t3-0.cnss80.St-Louis.t3.ans.net (140.222.80.1) 19 ms 25 ms 21 ms
```

```

6  t3-1.cnss25.Chicago.t3.ans.net (140.222.25.2)  24 ms  25 ms  26 ms
7  t3-0.cnss40.Cleveland.t3.ans.net (140.222.40.1)  32 ms  33 ms  34 ms
8  t3-1.cnss48.Hartford.t3.ans.net (140.222.48.2)  51 ms  46 ms  47 ms
9  t3-0.enss134.t3.ans.net (140.222.134.1)  51 ms  50 ms  49 ms
10 w91-rtr-external-fddi.mit.edu (192.233.33.1)  51 ms  52 ms  51 ms
11 E40-RTR-FDDI.MIT.EDU (18.168.0.2)  52 ms  52 ms  64 ms
12 MIT.MIT.EDU (18.72.2.1)  56 ms  53 ms  *

```

%

In this example, the entire route is operative. Each router between the local site and mit.edu responds. If one of the routers is down, however, each line after it has a line of astrisks like this:

```
%tracert gotham.hou.tx.us
```

```
tracert to gotham.hou.tx.us (198.64.55.96), 30 hops max, 40 byte packets
```

```

1  bruno-p5.rice.edu (128.42.42.1)  2 ms  2 ms  2 ms
2  verge.rice.edu (128.42.209.42)  3 ms  3 ms  2 ms
3  RICE3-F0.SESQUI.NET (192.67.13.86)  4 ms  4 ms  4 ms
4  LCIS.SESQUI.NET (192.136.149.9)  8 ms  4 ms  3 ms
5  192.136.150.6 (192.136.150.6)  243 ms  220 ms  213 ms
6  192.136.150.6 (192.136.150.6)  228 ms  *  *
7  *  *  *
8  *  *  *
9  *  *  *
10 *  *  *

```

(additional text deleted)

That could mean one of two things. Either the router after hop 6 doesn't support the feature of IP routing that Traceroute exploits, or there is an actual outage preventing contact with gotham.hou.tx.us. For this reason, *a PING test should always be done before a Traceroute test.*

If the PING succeeds and the Traceroute fails, it's likely that the problem is in using Traceroute to test that

particular route, not an actual end-to-end connectivity failure.

Let's assume, however, that PING and Traceroute have both indicated that there is an end-to-end connectivity failure. What next? Well, you could set about trying to find out who's responsible for the router on the last hop and see if they have any clues as to why the route to your destination is unavailable.

## Who Supports What?

Finding out who to call is a sticky business on the Internet. Regional Internet providers have network operation centers that vary in size and formality, from a couple of college students who only work from nine to five, to a full-fledged, mission-control-style operations center with three shifts of full-time staff.

When you find out who the technical contact for a remote site is, consider whether or not it's really worthwhile to call them. Perhaps you might want to get someone else to do it. If so, you can call the InterNIC Information Services Reference Desk. They may not only be able to help you sort out what kind of connectivity problems you might be having, but they may be able to direct you to the appropriate regional provider to get your problem resolved. The folks at InterNIC, however, are understandably loaded with requests concerning everything from how to get on the Internet, to inquiries about why things aren't working correctly.

The InterNIC Services Reference Desk is the Network Information Center of first and last resort. As the NIC of first resort, they try and direct Internet users to the appropriate folks to field their service requests and complaints. As the NIC of last resort, they will lend an ear to anyone who encounters problems because of the chaotic nature of the Internet, and the fact that after all is said and done, no one's really in charge.

Here's their contact information:

voice:	(619) 455-4600
fax:	(619) 455-4640
e-mail:	info@is.internic.net
Gopher:	gopher://rs.internic.net/
WWW:	<a href="http://www.internic.net/">http://www.internic.net/</a>

Try the online resources first—they're a virtual encyclopedic resource of connectivity information, with plenty of pointers to other sites.

If you're determined to try and get the regional contact information yourself, there are a few resources (discussed in the following section) you can use.

## Regional Contact References

The whois program enables you to have access to numerous remote databases. In this section, we discuss four in particular: the InterNIC Registration Services, the InternNIC Directory and Database Services, the MERIT Policy-Based Routing Database, and the MERIT Routing Registry Database.

The syntax for using each database varies slightly, but the following commands can be used to get help on each one:

The InterNIC Registration Services:

```
whois -h rs.internic.net
```

The InterNIC Directory and Database Services:

```
whois -h ds.internic.net help
```

The MERIT Policy-based Routing Database (being phased out soon):

```
whois -h prdb.merit.edu help
```

The MERIT Routing Registry Database (new system):

```
whois -h rrdb.merit.edu help
```

As already noted, the MERIT PRDB may be unavailable after April, 1995. The MERIT RRDB is a newer system that is still being loaded with the pertinent information.

To use the InterNIC Registrations Services to determine, for instance, to who's responsible for the router t3-1.cns25.Chicago.t3.ans.net found in the first Traceroute in the preceding example, we can do a search on the domain ans.net in the InterNIC Registration Services, as in this example:

```
%whois -h rs.internic.net ans.net
```

```
Advanced Networks & Services Inc. (ANS-DOM)
```

```
100 Clearbrook Road
```

```
Elmsford, NY 10523
```

```
Domain Name: ANS.NET
```

```
Administrative Contact:
```

```
Hershman, Ittai (IH4) ittai@ANS.NET
```

```
(914) 789-5337
```

```
Technical Contact, Zone Contact:
```

```
Wolfson, Claudia G. (CGW2) wolfson@ANS.NET
```

```
(914) 789-5369
```

```
Record last updated on 08-Mar-93.
```

```
Domain servers in listed order:
```

```
NIS.ANS.NET 147.225.1.2
```

```
NS.ANS.NET 192.103.63.100
```

The InterNIC Registration Services Host ONLY contains Internet Information (Networks, ASN's, Domains, and POC's).

Please use the whois server at nic.ddn.mil for MILNET Information.

As you see in the example, a query sent to rs.internic.net shows who the technical contact is for ans.net. In many cases, the information you retrieve through one of these databases will be a good starting point, but the person in the entry may not be the actual appropriate person with whom you should discuss connectivity problems.

## Getting Down to "Brass Tacts"

Your troubleshooting experience will be greatly enhanced if you remember that any "brass" you find in a whois database and call will probably already know about the problem; a little politeness will go a long way. It's unlikely that these numbers will be valid before or after their local working hours, so you should also restrict calls to conventional working hours for their time zone.

Occasionally, though, the technician at the remote site may be glad to have the opportunity to chat with a remote user and get some additional information about a current downage. The Internet technical community is every bit as diverse as the Internet itself, and the reaction from anyone you call about a technical problem is very much a pot-luck affair.

## Getting Around the Problem

If a problem is particularly resistant to troubleshooting efforts, it may be wise to consider a couple of alternatives.

If you've lost end-to-end connectivity, you might want to see if you can arrange Internet connectivity through an alternate provider. The resources listed earlier for the InterNic will also be productive in finding a suitable provider in your area for the type of service you want.

If the problem isn't end-to-end connectivity but loss of service, and you still have electronic mail access, look around on Gopher and World Wide Web sites for information about sites that offer e-mail access to services such as Gopher, WWW, Archie, FTP and other services. If all else fails, there are numerous commercial online services such as America Online and CompuServe that offer electronic mail access to the Internet, from which you can indirectly access many of the other services on the Net.

## Summary

When all is said and done, there's not a great deal you can do about many things that can go wrong on the Internet. You can get lots of information useful to yourself and others, but ultimately it's the responsibility of the service provider to ensure that their service delivers consistent and high quality to the end user.

It is unfortunate that occurrences on the Internet can make such troubleshooting skills a highly prized item; but in a way, it's better than other networks like the telephone voice network and the cable television networks, in which things go wrong and all you can do is wait for the problem to be resolved. If nothing else, at least these troubleshooting skills give you something interesting to do while you're waiting for the Net to start working again.



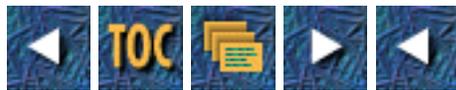


- [Part II](#)
  - [Everyday Guru Facilities](#)

## Part II

# Everyday Guru Facilities





- [5](#)

- [E-Mail](#)

- [Manipulate Your Mailer](#)
  - [Mail Options No Guru Does Without](#)
  - [Taking Advantage of Simple Mail Transport Protocol \(SMTP\)](#)
- [Exploiting Various Headers](#)
  - [Header Elements to Be Aware of](#)
    - [Received:](#)
    - [Reply-to:](#)
    - [BCC:](#)
- [Headers and Rules-Based Mail](#)
  - [Concepts of E-mail Rules Facilities](#)
    - [Pegasus Mail](#)
    - [Filter](#)
    - [Procmail](#)
- [Nathaniel Borenstein](#)
  - [by Tod Foley](#)
- [Anonymous E-Mail and Privacy Issues \(PEM, PGP, and So on\)](#)
  - [The Problem of E-Mail Security](#)
  - [Some E-Mail Privacy Solutions](#)
    - [The PEM Standard](#)
    - [Introducing RIPEM](#)
    - [Obtaining RIPEM](#)
  - [An Introduction to Pretty Good Privacy \(PGP\)](#)
    - [Advantages of PGP](#)
    - [Where to Get PGP](#)
    - [How to Use PGP](#)
    - [Caveats: Disadvantages of Using PGP](#)
  - [Finding Out More about PGP and PEM](#)
- [Subscribing and Staying Current on Listservs and Mailing Lists \(How a Guru Is Able to Track So Many Groups\)](#)
  - [Five Major Guru Rules for Reading All That Mail](#)

- [1. Don't Read It All](#)
- [2. Read Your Mail Every Day; Suspend Subscriptions When Away](#)
- [3. Be Selective in Your Subscriptions](#)
- [4. When Possible, Read News Rather Than Mail](#)
- [5. Organize Your Reading Process](#)

## 5

# E-Mail

[--by Philip Baczewski](#)

E-mail is the staple of any Internet habitué. The effective use of e-mail is particularly important for anyone wishing to take advantage of the Internet for interpersonal communication or information gathering. Experience is probably the best teacher in this regard, and any knowledge of an e-mail program on a particular system will be enhanced by knowledge of the editor employed within the e-mail program and by experience manipulating files on that particular system. There are some e-mail topics that go beyond the basic level of knowledge. Among these are using programs to preprocess and organize your mail, and making the content of your mail messages secure. The following discussion covers both of these areas, along with a number of other tidbits that you might find handy when working with electronic mail from the Internet.

## Manipulate Your Mailer

The more you know about your mail environment, the more efficient you can be in using electronic mail. The more you can control your mail environment, the more powerful you can be using electronic mail. Here are some items that may help on both accounts.

## Mail Options No Guru Does Without

No guru does without the options that enable the most efficient possible configuration of mail—unless, of course, those options aren't available. In general, I find the following to be important in the mail programs I use. Sorting—preferably multiple levels of sorting—should always be configurable in a good mail program. Forwarding is another important e-mail function that will be useful or required at some point in your Internet career. Another option to be aware of in your mail package is that of acknowledgements. The level of acknowledgement available from various mail programs varies. As you will see below, however, you can often generate your own acknowledgements via an e-mail preprocessor program. The most powerful automated manipulation of e-mail can be done with the preprocessing filter programs described in this chapter. An overall knowledge of your e-mail environment and a good idea of how you can best organize your incoming mail will ultimately provide you with the basis for using the various tools

described below.

## Taking Advantage of Simple Mail Transport Protocol (SMTP)

The openness of Internet mail exchange can make it easier to identify problems or find information about a particular Internet mail address. A telnet session to port 25 of an SMTP-capable system should get you that system's SMTP interface, which will enable you to perform several handy functions. It is important, however, that this capability be used judiciously. Abuse of any open system will only lead to a more closed future environment. The SMTP verify (VRFY) command can tell you that a user does or doesn't exist at a particular e-mail location (useful for solving delivery problems), and can also provide information about an alternate or forwarding address for that user. The expand (EXPN) command can confirm that an address is a mailing list and provide a membership list. Since VRFY and EXPN sometimes can return a false positive, these commands should be used to confirm other evidence of an address. More information about SMTP can be found in numerous references, including the original specification document, Internet Request for Comment (RFC) 821 (see the following for more information about obtaining RFCs).

## Exploiting Various Headers

If you haven't done so already and you want to know all about Internet mail headers, find a copy of RFC 822 (just do an Archie search on the string rfc822 and you are sure to find it). If you don't know, RFC 822 is the set of specifications that defines the way information is structured in an Internet e-mail header. You are probably familiar with most of the common elements of a header—From:, To:, Subject:, and so on. If you haven't managed an Internet mailer or had to solve mail-routing problems, you might be unaware of the usefulness of some of the other header elements.

### Header Elements to Be Aware of

The following are comments on the general usefulness of headers and may be a review or an overview for you. For the technical specifications of these fields, see RFC 822.

#### ***Received:***

The Received: field enables you to track a message through mail handlers and gateways. One Received: field is present for each delivery transaction. You can examine these to pinpoint gateways as well as to solve delivery problems. You might want to use a mail filter to process mail delivered from a specific site instead of from a specific person.

#### ***Reply-to:***

Mailing lists usually set the Reply-to: field to the list address so that replies will automatically be redistributed to all subscribers. This stimulates discussion on the list, but as you have probably seen, it also results in messages that were meant for an individual being inadvertently sent to the list. You can usually use your mail program to set a Reply-to: field that will supersede any From: field. Most mail programs also enable you to send your reply specifically to the From: field. When I forward mailing-list mail, though, I sometimes set the Reply-to: to my address to help prevent an accidental reply to the list.

#### ***BCC:***

**BCC:** helps ensure e-mail discretion as long as you are aware that you are the blind "copyee." "BCC" stands for "Blind Carbon Copy" and causes a copy of the message to be sent to the BCC address without that address showing up in the headers of the primary or carbon copies. (Since most computers do not use carbon paper, sometimes "CC" is referred to as a "Complementary Copy.") A word of advice: Be careful not to inadvertently or inappropriately forward mail that was intended to be discreet. The "reply all" function of your mail program can be dangerous in this situation as well.

## Headers and Rules-Based Mail

The contents of header fields such as From:, Resent-From:, Subject:, and others can serve as the basis for preprocessing and organizing your mail using a "rules" or filter facility. For example, if you wanted to discard automatically all messages whose subjects contain the word unsubscribe, a mail filter could scan for that word in mail subjects and perform a discard function based on such a match.

### Concepts of E-mail Rules Facilities

If you are willing to do some work, you can set up your filter to generate automated replies to selected mail messages or even trigger other automated processes. Most people will want to redirect messages to specific folders or forward mail to another location.

### Pegasus Mail

Several e-mail products in common use support filtering rules in one manner or another. For example, David Harris's Pegasus mail package, which runs on Novell file servers, has a built-in filter rules facility that enables you to perform actions based on the From:, CC:, To:, Subject:, or Reply-to: fields. Pegasus mail can check any of these fields for a specified string and can trigger twelve different functions based upon a match. Rules are defined by a fairly simple interface where you define the action fields, the string to match, the action and any parameters needed for the action. This is a simple, but effective, implementation of mail filter rules and illustrates the general concepts involved. (This concept has also reached commercial products like Lotus' cc:Mail and Novell's GroupWise 4.1, formerly WordPerfect Office mail.)

The following represents a sample filter rule definition in Pegasus mail.

```
If this field   : Cc, To
Contains       : [netmanl
                ]
Then do this   : Move
                (Using)   Network Managers List
```

### Filter

In the UNIX environment, some even more sophisticated tools are available for mail filtering. If you have the Elm mail package, it probably came with a program named Filter, which performs rules-based mail filtering in conjunction with that mail reader. To make use of the program, you must create a rules file, named filter-rules, in your .elm directory. Rules are rather simple in their construction. The general syntax is as follows:

```
if (condition) then action
```

*Condition* consists of a field, a relation, and a value. The fields include subject, sender, from, to, lines, and contains. For lines, the relation can be any standard numeric equivalence or comparison (<, >, =, <=, and so on). For other fields, contains can be used as a relation. Values are quoted strings for all fields (except lines) where the value is an integer. Nonnumeric values can also be regular expressions enclosed in quotes. Actions include delete, save *foldername*, savecopy *foldername*, forward *address* forwardc *address*, execute *command*, executec *command* leave. Most of these are self-explanatory. Forwardc and executec are commands that leave a copy of the message in your mailbox. Leave is the explicit command to leave the message in your mailbox.

A rule, for example, to delete all of those pesky subscribe and unsubscribe messages that are sent to mailing lists might look like this:

```
if (subject contains "subscribe") then delete
```

To log messages to a particular folder, you might try the following:

```
if (to contains "MACAV-L" and lines > 15) then save "~/mail/Macintosh"
```

Notice that in the preceding example two conditions are tested and linked via a logical AND operator. To test for the negative of a condition, the field value can be prefixed with NOT. A logical OR condition is not supported in Filter (at least not in Elm version 2.4). Comments can be placed in your filter-rules by starting the line with a pound sign (#).

Filter is invoked by placing a line in your .forward or equivalent file, which also means that your UNIX system must be running sendmail or some equally functional program. A typical entry in the .forward file might look like the following:

```
" | /usr/local/bin/filter -o ~/mail/FilterErrors"
```

Because the filter may be run while you are not logged on, Filter includes the option of logging any errors to a file, as specified by the -o option. (You might also log the errors to tmp or scratch space.)

Whenever mail forwarding is involved, it's best to observe some caution, lest you accidentally create e-mail loops or route all your mail to a null file. Your rules will be processed in the order that they appear in the file, so be sure to evaluate carefully whether the operations performed will be done in a logical and desirable sequence. You can check what rules are in place by using the command filter -r. Before using Filter, read *The Elm Filter System Guide*, which is distributed with the program. Documentation is also found in the Filter man page distributed with Elm.

## Procmail

Procmail is another UNIX program that can be used to preprocess your incoming mail based upon the content of the mail message. Procmail is a bit more powerful than Filter, since it can scan both the header and the body of your message. It also enables checking of multiple conditions within each rule definition (or in Procmail parlance, within each "recipe"). Procmail, like Filter, can be invoked via the .forward file, but Procmail can also be installed by the system administrator as the primary mail delivery agent. The latter case makes the configuration process a bit easier for the average user.

Procmail rules are defined in a file named .procmailrc, usually located in your home directory. The .procmailrc file contains environment variable assignments (some of which are significant to Procmail) and

recipes, and both can be freely interspersed within the file. Comments can be placed in the file by starting the line with a pound sign (#).

Procmail recipes begin with a line starting with a colon (:). Following the colon is an optional number that indicates how many conditions are to be checked within the recipe. Following the number are one or more optional flags that indicate how a recipe will be processed. The last item on a recipe's first line is a colon followed optionally by the name of a local lock file. Specifying the colon by itself will cause Procmail to use a lock file name formed from default values. If several Procmail programs are delivering mail at once, the existence of a lock file will prevent multiple attempts to deliver mail to a mail folder. Following the first line of a Procmail recipe are zero or more conditions, with each condition occupying its own line. A single action line follows any conditions.

The flags available are as follows:

<b>Flag</b>	<b>Function</b>
H	Searches the header of the message (the default)
B	Searches the body of the message
D	Distinguishes between case (ignore case is the default)
A	Recipe depends on the completion of the preceding recipe
a	Recipe depends on the successful completion of the preceding recipe
h	Processes (that is, forwards, saves, and so on) the header (default)
b	Processes the body (default)
f	Considers the processing to be a filter
c	Continues processing other recipes even if this recipe matches
w	Waits for processing to finish and checks a return code
W	Waits for processing to finish, checks the return code, but suppresses any failure message
i	Ignores any write errors for this recipe

Since Procmail uses egrep to search the specified portion of a message, conditions are written as egrep regular expressions. If a condition starts with an exclamation point (!), then the condition is inverted. Greater than (>) or less than (<) signs, followed by a decimal integer, can be used to make a condition true only if the size of the mail is greater than or less than the specified number (in bytes). Regular expressions starting with the value ^TO will be replaced with the expression ^(To|Cc|Apparently-To):.\* in order to find all destination specifications. The following condition would match any message sent to the Macintosh AV mailing list:

```
^TOMacav-1
```

A match of the string subscribe in the subject field would be found by the following:

```
^Subject:. *subscribe
```

An action can be as simple as specifying a folder name into which a message will be filed. Beginning an action with an exclamation point (!) will cause the message to be forwarded to an immediately following

e-mail address. A pipe (|) can be used to initiate some other action regarding the message. The following is an example of a .procmailrc file.

```
#Example .procmailrc file
```

```
#See man procmailex for more examples
```

```
LOGFILE=$HOME/mail/.proclog
```

```
MAILDIR=$HOME/mail
```

```
#Save any messages to "staff" in staff folder
```

```
:w:
```

```
^TOstaff.*
```

```
staff
```

```
#Save any message from the Macintosh AV list in the
```

```
#macintosh folder
```

```
:w:
```

```
^TOMacav-l
```

```
macintosh
```

```
#Save all correspondance with "Bob"
```

```
:w:
```

```
(^From|^To:|^Cc:).*Bob.*
```

```
BobMail
```

```
#Forward any mainframe-related mail
```

```
:2w:
```

```
^TO(mainframe&manager)
```

```
^Subject:.*mainframe.*
```

```
!myname@mainframe.here.edu
```

```
#Add all mail about horses to the horses archive and
```

```
#forward a copy to Betty
```

```
:wc:
```

```
^Subject:.*horse.*
```

```
|uncompress horses.Z;cat>>horses;compress horses
```

```
:A
```

```
!Betty@stable.com
```

```
#End of .procmailrc example
```

As is the case when dealing with e-mail, great care should be taken in writing your forwarding rules. If you are interested in using Procmail, it would be best to check with your system administrator as to whether it is installed globally or whether it will work with your particular system mailer. For more information on Procmail, see `man procmail`, `man procmailrc`, and `man procmailex`.

## Nathaniel Borenstein

**by Tod Foley**

Chief Scientist at First Virtual Holdings, Inc., Nathaniel Borenstein is a world-rekknowned doctor of computer science and a veritable legend of the online world. His programming work focuses on e-mail encoding/decoding and transport, and it places special emphasis on interoperability issues and the problems associated with multimedia on the Net. Nathaniel's programming credits include such applications as Andrew (a pioneering multimedia messaging application), metemail (a portable MIME implementation used by millions worldwide), and the ATOMICMAIL and Safe-Tcl languages (for safe and portable interactive electronic mail messages); but his most famous work is undoubtably MIME (the Internet standard format for interoperable multimedia data). He is also the founder of the experimental electronic magazine *Electric Eclectic*, and the author of *Programming As If People Mattered: Friendly Programs, Software Engineering, and Other Noble Delusions*. Now, with his partners at First Virtual, he continues to forge ahead, developing new answers to the problems of networked information delivery. In a recent e-mail interview, Nathaniel explained the groundbreaking MIME format, and took a look at the future of the pan-platform multimedia world.

**TF:** These days, many people are familiar with MIME—at least in terms of seeing the word in their e-mail headers—but few people seem to grasp the concept. What is MIME *really*, and how does it extend the e-mail capabilities of users?

**NB:** I can tell you what *I* think MIME really is. This isn't what I thought when we were designing it, but it's what I think now:

I think that the best one-phrase summary of MIME is "ASCII version two." By this, I mean that it is a data representation format that is totally machine-independent and can be used anywhere, like ASCII. In fact, it

is represented entirely *in* ASCII, so you could view it as a backwards-compatible extension to ASCII. What it comes down to is a well-defined, totally portable way to represent *any* kind of data inside ASCII. This explains why MIME is rapidly being adopted for so many non-e-mail uses (netnews, WWW, Gopher, and even for X11 cut & paste in the COSE proposals). It turns out that there are *lots* of applications for a totally platform- and vendor-neutral way to represent any kind of data in the world and encapsulate it in an ASCII byte stream. That's what MIME's turning out to be.

**TF:** So it's a *transparent protocol*, not a tool or service. . . .

**NB:** Well, what MIME becomes is the *lingua franca* for communication between multimedia applications, even though users never need to know they're using MIME. Today, if two users exchange mail, and one of them is using, say, Eudora, and the other is using Lotus Notes, and the mail being exchanged between them includes a picture (which is something both systems support), then one of two things will happen:

1. The message will be garbled.
2. The message will work fine, because they're using a recent version of Eudora and a recent Notes-to-SMTP gateway, both of which support MIME.

Note what this looks like from the end user's perspective: either "the system" works, or it doesn't. Evolutionarily, it used to be always case #1, and is slowly turning into case #2 more often. Even a user who is alert enough to notice the improvement will just think "they've improved the software," but such users never have to hear the word "MIME" or know anything about it at all.

**TF:** That's a great description•97Ämakes things a *lot* clearer. Of course the primary source for information on MIME would be RFC 1521, but are there any other good sources for users to turn to? Third-party literature?

**NB:** Well. . . that depends on your definition of "third-party." I'd refer you to Marshall Rose's book *The Internet Message*, published by Prentice Hall, which has a whole chapter on MIME. There's also the MIME FAQ, which is itself third-party, and which contains lots of pointers to other information.

**TF:** How does one use MIME to encode multimedia e-mail—assuming the mailer has that capability?

**NB:** That depends on the mail tool you're using. In some systems, you have to type an arcane command, followed by the filename, and you even have to type in the MIME content-type (for example, image/gif, or audio/basic, and so on) by hand. Some systems have an "include attachment" menu or button, and may or may not use file type information to deduce the content-type. Other systems have particular buttons or menus for "include image" or "include sound," or even more specialized ones such as "include screen dump" or "record & include sound." The best systems (including Andrew, about which I am a shockingly immodest author <g> ) will even let you compose and edit pictures, sounds, and so on, right in the middle of your mail message as you compose it. But as long as these systems all produce MIME as their data format, their users can all interoperate. MIME simply increases the infrastructure available for vendors of ever-better mail tools.

**TF:** Are MIME messages transportable across both SMTP and UUCP networks?

**NB:** Absolutely. In fact, we designed MIME to be transportable across every mail system in the known world. If properly encoded, MIME messages can pass through gateways that connect not only SMTP and UUCP, but also Bitnet, FidoNet, X.400, SNADS, PROFS, and everything else (including some really odd

stuff) that was known to the hundred-odd mail gurus involved in the protocol design. A MIME message can pass through any or all such gateways, and as long as the headers aren't stripped off, if the recipient has MIME-smart software, the message will be seen in its full multimedia glory. Einar Stefferud, one of the great visionaries and grand old men of the Internet, likes to speak of MIME as generalized "tunnelling technology," by which arbitrary data can be safely transported through some of the strangest transport gateways in the world. He also calls it "teflon-coated e-mail armor," or something like that, but my favorite description is still "ASCII version two."

**TF:** Metamail—the program that performs the actual MIME encoding/decoding and display—is described as "a portable MIME implementation" which is called by the mailer. I would imagine that Andrew calls metamail; and it's fairly common knowledge that Pine and Eudora also support its use—what other mailers currently allow for calls to metamail?

**NB:** Well, actually, this is not quite right. Andrew, Pine, and Eudora all support MIME, but the implementation in Eudora is totally independent of metamail. I'm not sure about Pine, but I think it started out as mostly metamail but is adding stuff. Andrew uses metamail as a fallback when it hits a MIME content-type for which it doesn't have built-in support; but it has built-in support for the most common types, such as image and audio data. So the story is considerably more complicated.

Basically, metamail was designed to be a program that you could easily plug into any mail reader to let it display MIME. This strategy was a very successful way to add *minimal* support for MIME to existing mail readers, but the result was not what you'd call a seamless or well-integrated user interface. I always saw it as primarily being a transitional tool, to aid in the transition from pre-MIME to post-MIME mail readers. It turns out to be a bit more than that, because nowadays a fair number of MIME-smart mail readers are being designed to use metamail as a fallback for unrecognized data types, the way Andrew does. This is a nice strategy because metamail is so easily configured to handle new content-types by running separate external viewing programs.

Anyway, to get back to your original question, it turns out that I don't really know all the tools that use metamail. I do know that not all of them are mail readers—there are news readers and various other tools that use metamail to handle multimedia "attachments" in MIME format. I know that I have personally patched or received patches for Berkeley Mail and its many variants; Xmail, Mailtool, MH, XMH, MH-E, Rmail, VM, the four Andrew mail-reading tools; Elm, Mush, MM, Msgs, UUPC, RN, TRN, and various lesser-known packages. It has been fairly amazing to watch the way it has taken off. It was 22 days from the first public release of metamail on SunOS to my receipt of the first MS-DOS port, which is when I was really sure I had a hit on my hands.

**TF:** What about offline applications? Is the day coming when my paint program will have the option to save directly into MIME format instead of saving bit maps or .GIFs?

**NB:** That's a particularly interesting question, I think, because it goes to the heart of what the world really needs for multimedia interoperability. There are still competing approaches in this regard, such as Bento, and I'm not sure I can predict the future. The cut & paste situation in X11 is the best example of why MIME would be useful for these "offline applications," as you call them—you need either MIME or something just like it if you're ever going to be able to cut & paste compound multimedia stuff including text, audio, and video between independently-developed applications. I think that day probably is coming on UNIX systems. It's harder to predict in non-UNIX environments because there's more of a chance for vendor-defined standards. For example, cut & paste already works fairly well on the Mac, because Apple

managed to control the standards quite nicely by anticipating the need, at least at the level of things like bit maps. (Cut & paste actually breaks down at a more structured level of data, but it remains to be seen if the market really requires the cutting & pasting of active spreadsheets, for example.) Microsoft could have probably set the standards equally well for Windows, but they may have missed the boat by not doing it right sooner—I don't know.

I'd certainly like to think that the day you mention is coming, but if it is I think it will be customer-driven. It will happen if enough users demand the ability to get really high-level interoperability between independent multivendor applications. And that will only happen if consumers are educated enough to realize that this is even possible.

You can obtain more information on MIME and related topics in the following places:

The MIME FAQ, currently maintained by Jerry Sweet, can be found via anonymous FTP at a number of sites; simply ask Archie to find mime-faq, or FTP to any of these sites:

[rtfm.mit.edu/pub/usenet-by-group/news.answers/mail/mime-faq/\\*](ftp://rtfm.mit.edu/pub/usenet-by-group/news.answers/mail/mime-faq/)

[ftp.uu.net/usenet/news.answers/mail/mime-faq/\\*](ftp://ftp.uu.net/usenet/news.answers/mail/mime-faq/)

[ftp.ics.uci.edu/mh/contrib/multimedia/mime-faq.txt.gz](ftp://ftp.ics.uci.edu/mh/contrib/multimedia/mime-faq.txt.gz)

To retrieve the FAQ by FTPmail, send e-mail to

[archive-server@ics.uci.edu](mailto:archive-server@ics.uci.edu)

with the text body:

mimesend /mh/contrib/multimedia mime-faq.txt.gz

The MIME FAQ is also posted monthly to the Usenet newsgroup comp.mail.mime.

mime.txt, a good primer on MIME and how it works, can be found at [ftp.netcom.com](ftp://ftp.netcom.com/pub/mdg/mime.txt), in the directory /pub/mdg/mime.txt.

MIME, metainfo, and related programs and sample files are available via anonymous FTP from [thumper.bellcore.com](ftp://thumper.bellcore.com/pub/nsb), in the /pub/nsb directory.

## **Anonymous E-Mail and Privacy Issues (PEM, PGP, and So on)**

In my capacity as Bitnet support person, I once ran across the classic bit of forged e-mail shown in the following screen dump. This was a message posted to the Bitnet LINKFAIL mailing list—undoubtedly as a parody of the usual kinds of notices posted there. While it provided an amusing aside to the usual monitoring of downed Bitnet links (a critical issue at that time, since Bitnet was still primarily a store-and-forward network), it also drove home the fact that not all e-mail was authentic. Today, while a number of safeguards have been put in place by system managers, forged e-mail is still a possibility. The following Bitnet Listserv mail message was a funny, if obvious, forgery.

IPFRCVM Iowa Pig Farm Research Center will be down tomorrow from

20:0023:00 for system maintenance. Since we are an end node, nobody will be affected except for us.

It turns out that one of our sows got in through a hole in the wall and had her litter of piglets under our raised floor. The operator on duty got quite a scare when he heard a number of squeals. He assumed we had some massive head crashes and powered down the CPU. Since the squeals continued we traced it to a corner under our raised floors. We will be off the air tonight so that we can power down again and get the sow and her piglets out from under the floor.

Zeke System Grunt IPFRC

Anonymous access to the Internet is counter to the usage guidelines of most Internet service providers, who require that news posts and mail be traceable to a specific computer or human user. In spite of this requirement, however, standard Internet mail is not guaranteed to be legitimate, secure, or authentic. It's still possible to forge SMTP mail, but these days it's harder to do so without getting caught. While mail is, for the most part, provided a degree of legitimacy by the security of the computing systems attached to the Internet, standard Internet mail has to date not been secure enough to support things like monetary transactions, business negotiations, and other truly private transactions.

## **The Problem of E-Mail Security**

It is the intent of most system managers to keep their systems secure and to guard user privacy. Such concern for security and privacy usually extends to electronic mail as well. After mail leaves a host machine, control over that security and privacy is out of the hands of the sender or the system's manager. While it is counterproductive to question the motives and integrity of every system manager other than your own, most of us who have spent a good deal of time on the Internet live by the tenet that e-mail is everlasting and ubiquitous. Never commit something to an e-mail message that you would not be able to defend or laugh off at a later date.

To quell some tinges of paranoia, it is unlikely that some secret agency is routinely gathering all Internet e-mail messages ever sent. They would probably like to, but the volume of mail is just too great. The problem is far more practical and mundane. For example, messages sometimes pass through *gateways*—computers that use disk space to store messages for later delivery. Disk drives get backed up on a regular basis. In other words, your e-mail message may exist on someone's backup tape. Although it may be the case that no one knows that your e-mail message is there, that does not prevent that message from being accessed at some later time. The way systems communicate on the Internet is also part of the problem. Internet protocols are open and well documented. While it's usually software programs that do so,

anyone can connect to an SMTP mail agent and begin to enter commands. because most e-mail messages exist in clear text, there is no guarantee that it will be read only by the individual for whom it is intended.

## **Some E-Mail Privacy Solutions**

Since e-mail exists in clear text, the answer to security must lie in encryption, right? Well, the answer is both no and yes. Standard encryption methods require that some "key" value be used to encode and decode the text. If you are sending e-mail, transmitting a key value is problematic. It is possible to do so via some other secured channel of communication, but it certainly would not be convenient, especially on a regular basis. The answer may lie in the dual key model of encryption proposed by the RSA (Rivest, Shamir, and Adleman) standard, published (much to the NSA's chagrin, apparently) in 1977.

The RSA algorithm makes use of a public and private key to ensure both privacy and authenticity. The private key remains the secret possession of an individual, while an associated public key can be published to others on the network. To put it simply, text encrypted using the public key can only be decrypted using the corresponding private key, thereby ensuring that only the recipient (and private key holder) can access the text, thus preserving the privacy of the message. Likewise, text encrypted with the private key can be decrypted only using the associated public key, thereby ensuring that the text is indeed originating from the sender (and private key holder), thus providing authentication.

## **The PEM Standard**

Implementation of RSA security for Internet mail has progressed far beyond the proposal stage, but it has not yet reached the practical level of other implemented and officially accepted Internet standards. Any mail program that wants to operate on the Internet must be capable of "speaking" SMTP (or at least be on speaking terms with another program that is). Very few programs are yet "speaking" PEM.

PEM stands for Privacy Enhanced Mail, and is documented in RFC numbers 1421, 1422, 1423, and 1424. The PEM standard is intended to implement the RSA encryption algorithm in a manner that can support the exchange of secure electronic mail via Internet protocols. The RFCs mentioned above outline all aspects of managing such a protocol on the Internet, including encryption algorithms used for management of public and private keys. The standard proposes a centralized hierarchical structure for the issuance of public key certificates, and outlines how public keys could be integrated with directory services. The overall scheme is ambitious and has not been fully implemented or officially adopted at this writing.

One problem with relying on a dual-key encryption method is the inability to ensure that published public keys are actually associated with the person whom they purport to represent. If an impostor publishes a public key for a person, then the impostor can decode messages intended for that person if unsuspecting people use the bogus key to encrypt messages. The PEM standard proposes a hierarchy of certification authorities who would be charged with the task of providing presumably secure "key certificates." Some people, however, are uncomfortable with entrusting this task to a central authority and advocate a "grass-roots" network of authentication, with one's personal authorities serving as the basis for deciding that a public key is valid. The debates on this topic and on topics such as the U.S. Government-proposed "clipper chip" show that this can be an emotional and complicated issue.

## **Introducing RIPEM**

A version of PEM is available from a company name Trusted Information Systems. However, another

version named RIPEM seems to be more readily available via the Internet. RIPEM is a program written by Mark Riordin that implements many aspects of the PEM standard but does not implement a hierarchical certification scheme.

RIPEM makes use of a package of encryption algorithms named the RSAREF Toolkit, which is available from RSA Data Security Inc. at the anonymous FTP site [rsa.com](http://rsa.com). RSA licenses several patented encryption algorithms (including the RSA algorithm) from a company named Public Key Partners. Public Key Partners owns the exclusive licensing rights to those algorithms. The RSAREF Toolkit—actually a package of C libraries—is freely available subject to export limitations and licensing restrictions. Because of U.S. Government rules, only citizens of the U.S. and Canada are eligible to acquire and use the RSAREF package. This is also true of RIPEM and other RSA encryption packages. The following copy of the RSA license agreement illuminates the various restrictions in relation to this package.

RSA LABORATORIES

PROGRAM LICENSE AGREEMENT

January 5, 1993

RSA LABORATORIES, A DIVISION OF RSA DATA SECURITY, INC. ("RSA")

GRANTS YOU A LICENSE AS FOLLOWS TO THE "RSAREF" PROGRAM:

1. LICENSE. RSA grants you a nonexclusive, nontransferable, perpetual (subject to the conditions of section 8) license for the "RSAREF" program (the "Program") and its associated documentation, subject to all of the following terms and conditions:
  - a. to use the Program on any computer in your possession;
  - b. to make copies of the Program for backup purposes;
  - c. to modify the Program in any manner for porting or performance improvement purposes (subject to Section 2) or to incorporate the Program into other computer programs for your own personal or internal use, provided that you provide RSA with a copy of any such modification or Application Program by electronic mail, and grant RSA

a perpetual, royaltyfree license to use and distribute such modifications and Application Programs on the terms set forth in this Agreement.

- d. to copy and distribute the Program and Application Programs in accordance with the limitations set forth in Section 2.

"Application Programs" are programs which incorporate all or any portion of the Program in any form. The restrictions imposed on Application Programs in this Agreement shall not apply to any software which, through the mere aggregation on distribution media, is colocated or stored with the Program.

## 2. LIMITATIONS ON LICENSE.

- a. RSA owns the Program and its associated documentation and all copyrights therein. You may only use, copy, modify and distribute the Program as expressly provided for in this Agreement. You must reproduce and include this Agreement, RSA's copyright notices and disclaimer of warranty on any copy and its associated documentation.
- b. The Program and all Application Programs are to be used only for noncommercial purposes. However, media costs associated with the distribution of the Program or Application Programs may be recovered.
- c. The Program, if modified, must carry prominent notices stating that changes have been made, and the dates of any

such changes.

d. Prior permission from RSA is required for any modifications that access the Program through ways other than the published Program interface or for modifications to the Program interface. RSA will grant all reasonable requests for permission to make such modifications.

3. NO RSA OBLIGATION. You are solely responsible for all of your costs and expenses incurred in connection with the distribution of the Program or any Application Program hereunder, and RSA shall have no liability, obligation or responsibility therefor. RSA shall have no obligation to provide maintenance, support, upgrades or new releases to you or to any distributee of the Program or any Application Program.
4. NO WARRANTY OF PERFORMANCE. THE PROGRAM AND ITS ASSOCIATED DOCUMENTATION ARE LICENSED "AS IS" WITHOUT WARRANTY AS TO THEIR PERFORMANCE, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE PROGRAM IS ASSUMED BY YOU AND YOUR DISTRIBUTEES. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU AND YOUR DISTRIBUTEES (AND NOT RSA) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
5. LIMITATION OF LIABILITY. EXCEPT AS EXPRESSLY PROVIDED FOR IN

SECTION 6 HEREINUNDER, NEITHER RSA NOR ANY OTHER PERSON WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE PROGRAM SHALL BE LIABLE TO YOU OR TO ANY OTHER PERSON FOR ANY DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF RSA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

6. PATENT INFRINGEMENT OBLIGATION. Subject to the limitations set forth below, RSA, at its own expense, shall: (i) defend, or at its option settle, any claim, suit or proceeding against you on the basis of infringement of any United States patent in the field of cryptography by the unmodified Program; and (ii) pay any final judgment or settlement entered against you on such issue in any such suit or proceeding defended by RSA. The obligations of RSA under this Section 6 are subject to:
- (i) RSA's having sole control of the defense of any such claim, suit or proceeding;
  - (ii) your notifying RSA promptly in writing of each such claim, suit or proceeding and giving RSA authority to proceed as stated in this Section 6; and
  - (iii) your giving RSA all information known to you relating to such claim, suit or proceeding and cooperating with RSA to defend any such claim, suit or proceeding. RSA shall have no obligation under this Section 6 with respect to any claim to the extent it is based upon (A) use of the Program as modified by any person other than RSA or use of any Application Program, where use

of the unmodified Program would not constitute an infringement, or (B) use of the Program in a manner other than that permitted by this Agreement. THIS SECTION 6 SETS FORTH RSA'S ENTIRE OBLIGATION AND YOUR EXCLUSIVE REMEDIES CONCERNING CLAIMS FOR PROPRIETARY RIGHTS INFRINGEMENT.

NOTE: Portions of the Program practice methods described in and subject to U.S. Patents Nos. 4,200,770, 4,218,582 and 4,405,829, and all foreign counterparts and equivalents, issued to Leland Stanford Jr. University and to Massachusetts Institute of Technology. Such patents are licensed to RSA by Public Key Partners of Sunnyvale, California, the holder of exclusive licensing rights. This Agreement does not grant or convey any interest whatsoever in such patents.

7. RSAREF is a noncommercial publication of cryptographic techniques. Portions of RSAREF have been published in the International Security Handbook and the August 1992 issue of Dr. Dobb's Journal. Privacy applications developed with RSAREF may be subject to export controls. If you are located in the United States and develop such applications, you are advised to consult with the State Department's Office of Defense Trade Controls.
8. TERM. The license granted hereunder is effective until terminated. You may terminate it at anytime by destroying the Program and its associated documentation. The termination

of your license will not result in the termination of the licenses of any distributees who have received rights to the Program through you so long as they are in compliance with the provisions of this license.

## 9. GENERAL

- a. This Agreement shall be governed by the laws of the State of California.
- b. Address all correspondence regarding this license to RSA's electronic mail address <rsarefadministrator@rsa.com>, or to

RSA Laboratories

ATTN: RSAREF Administrator

100 Marine Parkway, Suite 500

Redwood City, CA 94065

RIPEM is available for several platforms including UNIX, Macintosh, and MS-DOS. It is a program that acts independently of your mail program, providing file encryption/decryption, and public and private key management. It is not without its limitations, however. As per the PEM standard, it can act upon text files only. It is limited to a 1023-character line length, and the encryptable message size is limited by the available size of your computer system's memory.

RIPEM and other similar programs use a two-step encryption process. The message text is first encrypted using the Data Encryption Standard (DES) algorithm and that encryption's key value is then encrypted using the RSA algorithm. This process takes less time, since RSA encryption generally is more complex than DES. The text remains secure because the DES key is guarded by the RSA encryption. RIPEM can also ensure that a file has not been altered. It does this by encrypting a hash value with a signed file. The whole process is transparent to the user.

### **Obtaining RIPEM**

The official distribution site for RIPEM is [ripem.msu.edu](http://ripem.msu.edu), and information about the program can be found in the /pub/crypt/ripem directory. The files ripemusr.txt and ripem.man (the man page) are particularly informative. Since RIPEM and programs like it are export-restricted, the actual executables are only

distributable upon request. The RIPEM source code has been placed in the public domain, but because of RSAERF restrictions commercial use of the program is apparently forbidden.

To gain access, you can send an e-mail message to ripem@ripem.msu.edu with several pieces of information: you must provide your citizenship (must be U.S. or Canadian to be authorized to receive RIPEM), you must state your willingness to comply with relevant export laws and software licenses, and you must provide the fully-qualified domain name of your Internet host and the country in which it resides. Assuming you qualify, you will receive a message with a special FTP name and password usable only from your specified host. RIPEM is also available from rsa.com with similar restrictions. The following README file on that host gives the instructions for accessing both RSAREF and RIPEM and gives you a good idea of the restrictions involved.

Please note: RSAREF and the associated programs distributed in this directory are EXPORT CONTROLLED and are available via anonymous FTP to U.S. persons only. Canadian citizens may obtain RSAREF via electronic mail by sending a message acknowledging the four points below, changing the country as necessary, to <rsarefadministrator@rsa.com>. The RSAREF license is in the file 'license.txt'. Please read it. If you acknowledge that:

1. You have read the RSAREF Program License Agreement and understand and agree to be bound by its terms and conditions, including without limitation its restrictions on foreign reshipment of the Program and information related to the Program;
2. The computer to which you are requesting that the program be transmitted is located in the United States of America;
3. You are a United States citizen or a permanent resident of the United States; and
4. The RSAREF Program License Agreement is the complete and exclusive agreement between RSA Laboratories and you relating

to the Program, and supersedes any proposal or prior agreement, oral or written, and any other communications between RSA Laboratories and you relating to the Program,

then change to the following directory:

```
dist/U.S.onlyf1dd43
```

This is a timedependent directory. Check this file again if you can't find the directory.

RSAREF is in the files 'rsaref.tar.Z' and 'rsaref.zip'. On the first you would run the commands:

```
uncompress rsaref.tar.Z      # produces rsaref.tar
tar xvf rsaref.tar
```

On the second you would run:

```
pkunzip rsaref.zip
```

RIPEM, Mark Riordan's RSAREFbased privacyenhanced mail program, is also distributed in that directory. 'ripem\*.tar.Z' includes RSAREF, so you only need to get one file. 'ripem.el' contains an Emacs command interface to RIPEM. 'ripemmacintosh\*.sit.hqx' is Raymond Lau's version of RIPEM for the Macintosh(R), archived and compressed with StuffIt and encoded in ASCII with BinHex.

A collection of RIPEM public keys is in the file 'rpubkeys'. Those keys are uncertified, so use them AT YOUR OWN RISK. We just add whatever people send to us. To have your RIPEM public keys added, send electronic mail to <ripemkeys@rsa.com>.

If you have questions on RSAREF software, licenses, export restrictions, or other RSA Laboratories offerings, send electronic mail to <rsarefadministrator@rsa.com>.

Revised December 1, 1993

## **An Introduction to Pretty Good Privacy (PGP)**

Pretty Good Privacy, or PGP, has the potential to be the Kermit of the 1990s. Kermit, the public-domain, file-transfer protocol developed at Columbia University, is—while not necessarily the fastest or most elegant—certainly the most widespread solution. It achieved its ubiquitous status because it was freely available and implementable on many diverse computer systems. PGP has yet to attain the same status, but like Kermit it is freely available for noncommercial use (but not in the public domain), and because the source code is available, it has the potential to be implemented on diverse systems.

PGP is an implementation of dual-key encryption for use with Internet mail and for general data security. PGP was written by Philip Zimmerman, and is credited to his company, Phil's Pretty Good Software. PGP also has a rather storied history, since it has been in the center of a controversy related to whether posting on an anonymous FTP site constitutes export, and it has drawn attention to the sensitivity about cryptography technology in some government circles. Early versions were subject to some RSA license questions as well. The current versions of PGP have been developed using the RSAREF Toolkit, a move that has apparently quelled any ambiguities about licensing.

## **Advantages of PGP**

PGP is very explicitly a publicly available security solution—RSA for the "little guy." Its author is quite open and eloquent about the various issues of encryption security within the electronic society of the Internet. PGP's attractiveness goes beyond this availability, however. Unlike RIPEM and other PEM-based programs, PGP goes beyond simple encryption and includes built-in compression and ASCII encoding as well. PGP can perform the compression functions of zip-like programs and do ASCII encoding (similarly to uuencode) in one single operation and provide encryption security on top of it all. It can also produce clear text messages with encrypted signatures attached. PGP is very simple to use, but it is highly recommended that you read all the accompanying documentation to fully understand the concepts of its use as well as the program's limitations.

## **Where to Get PGP**

The official distribution point for PGP in the U.S. is at the Massachusetts Institute of Technology (MIT), on the node net-dist.mit.edu, in the directory /pub/PGP. The README file in that directory will provide an explanation of the procedure for acquiring PGP from that site. An Archie search for PGP will turn up numerous other sites. Care should always be taken, however, to obtain the package from the most authoritative source available. A commercial version is available from a company named ViaCrypt (2014 West Peoria Avenue, Phoenix, Arizona 85029; tel. 602-944-0773).

## **How to Use PGP**

After PGP is acquired, installed, and configured, it is fairly straightforward to use. The `pgp` command is used in combination with a number of subcommand options to perform various functions. Encryption is accomplished by using the `-e` option and decryption is accomplished by using the command `pgp filename`, where your filename usually has a `.pgp` extension. Another possible option is `-a`, which produces an ASCII-encoded output (radix-64) suitable for mailing over the Internet. For example, if you wanted to create a file encrypted with someone's public key and signed with your private key and suitable for sending via an e-mail message, you might use a command like the following:

```
pgp -esa testfile.txt Bob Jones
```

The following is a key-generation dialog resulting from the command `pgp -kg`.

```
pgp -kg
```

```
Pretty Good Privacy(tm) 2.6  Publickey encryption for the masses.
```

```
(c) 19901994 Philip Zimmermann, Phil's Pretty Good Software. 23 May 94
```

```
Distributed by the Massachusetts Institute of Technology. Uses RSAREF.
```

```
Export of this software may be restricted by the U.S. government.
```

```
Current time: 1994/06/07 15:56 GMT
```

```
Pick your RSA key size:
```

- 1) 512 bits Low commercial grade, fast but less secure
- 2) 768 bits High commercial grade, medium speed, good security
- 3) 1024 bits "Military" grade, slow, highest security

```
Choose 1, 2, or 3, or enter desired number of bits: 2
```

```
Generating an RSA key with a 768bit modulus.
```

```
You need a user ID for your public key. The desired form for this  
user ID is your name, followed by your Email address enclosed in  
<angle brackets>, if you have an Email address.
```

```
For example: John Q. Smith <12345.6789@compuserve.com>
```

```
Enter a user ID for your public key:
```

```
John R. Smith <smith@somewhere.edu>
```

You need a pass phrase to protect your RSA secret key.

Your pass phrase can be any sentence or phrase and may have many words, spaces, punctuation, or any other printable characters.

Enter pass phrase:

Enter same pass phrase again:

We need to generate 385 random bits. This is done by measuring the time intervals between your keystrokes. Please enter some random text on your keyboard until you hear the beep:

0 \* Enough, thank you.

.....++++ .....++++

Key generation completed.

The following is an example of a simple encryption/decryption process.

C:\PGP:pgp e c:\testfile.txt o testfile.pgp

Pretty Good Privacy(tm) 2.6 Publickey encryption for the masses.

(c) 19901994 Philip Zimmermann, Phil's Pretty Good Software. 23 May 94

Distributed by the Massachusetts Institute of Technology. Uses RSAREF.

Export of this software may be restricted by the U.S. government.

Current time: 1994/06/07 14:42 GMT

Recipients' public key(s) will be used to encrypt.

A user ID is required to select the recipient's public key.

Enter the recipient's user ID: John R. Smith

Key for user ID: John R. Smith

768bit key, Key ID 0E922109, created 1994/06/07

.

Ciphertext file: testfile.pgp

C:\PGP:pgp testfile.pgp

Pretty Good Privacy(tm) 2.6 Publickey encryption for the masses.

(c) 19901994 Philip Zimmermann, Phil's Pretty Good Software. 23 May 94

Distributed by the Massachusetts Institute of Technology. Uses RSAREF.

Export of this software may be restricted by the U.S. government.

Current time: 1994/06/07 14:49 GMT

File is encrypted. Secret key is required to read it.

Key for user ID: John R. Smith

768bit key, Key ID 0E922109, created 1994/06/07

You need a pass phrase to unlock your RSA secret key.

Enter pass phrase: Pass phrase is good. Just a moment.....

Plaintext filename: testfile

C:\PGP:

## **Caveats: Disadvantages of Using PGP**

One disadvantage to relying upon PGP or other programs such as RIPEM is that your intended recipient must also have a copy of the program or of a program that supports the same implementation of the encryption algorithm. As a standard develops, it will be increasingly possible to have some assurance that your message can be decrypted and verified by the recipient. For example, the same is currently true of the MIME standard for mail attachments. Even though MIME is fairly well established, it will become a useful standard after most e-mail programs have support built in.

Unlike most MIME support, PGP and other similar programs are currently implemented independently of mail programs and require you to act upon a file before sending it via e-mail. The PEM standard seems to support integration of security into mail applications, but at the time of this writing, no major mail programs are supporting built-in PEM. Such independence will not necessarily prevent a standard from being established. Uuencode/uudecode and the various "zip" programs have achieved de facto standard status and are commonly available on many platforms and in many locations. A standard becomes useful when a "critical mass" of people or computers employ it, and PGP has as much chance of catching on as many other programs at this point.

Verifying the security of the security software and signature certificates can be somewhat problematic with

PGP. Because no central authority provides or certifies the public keys for PGP, you must rely on "personal" authorities to verify signatures. PGP even supports this interpersonal certification method, enabling you to certify public keys on your key ring for later distribution to others and enabling you to specify your confidence level in a particular key based upon who else has certified it. MIT serves as the central authority for distribution of the PGP software, and also maintains some public keys.

Export controls and other licensing issues related to products using RSA security can be somewhat complex and sometimes confusing. The fact that U.S. export restrictions for this software are in place currently reduces the likelihood of it becoming a global standard. (While there is an international version of PGP named PGPui, establishment of a global standard is still somewhat inhibited by U.S. restrictions.) An overview of the political debate on this subject is not appropriate here, but if you research these programs you can't help but get a sense of the nature of the debate. The PGP documentation provides a good overview of some of the legal issues involved for this program and others like it.

## Finding Out More about PGP and PEM

If you are new to RSA security and want to learn more about the general concepts and some of the technical issues involved, the PGP documentation files (pgtdoc1.txt and pgpdoc2.txt) are great places to start. As already mentioned, they provide a good overview of the technology involved, as well as an overview of the legal and licensing issues. The RIPEM documentation (ripemusr.txt) is also helpful in this regard. Other sources for information are the news groups alt.security.pgp, alt.security.ripen, sci.crypt, and info.pem-dev.

## Subscribing and Staying Current on Listservs and Mailing Lists (How a Guru Is Able to Track So Many Groups)

### [ALEPH: The Rise and Fall of a Virtual Community --by Tod Foley](#)

---



The Internet is touted as a place where people of diverse types and scattered localities can come together to work, play, and socialize in the spirit of the common good, congregating in all manner of self-organizing groups often known as *virtual communities*. As many Net gurus will tell you, e-mail lists present an excellent environment for the construction of such experimental communities: distribution is handled automatically, list subscription can be strictly controlled (if desired), and the asynchronous nature of the medium enables subscribers to participate at their own pace, in their own unique way.

To many people (including myself), this sort of structure represents total creative freedom, and perhaps the best environment yet designed for shared human labor and social expression. Last year, I was a key participant in one of these working environments—a progressive hothouse of furious metaphysical speculation called aleph@pyramid.com—where I learned that such freedom is not without its cost.

### The Ambistucture of Aleph

One way of looking at an e-list is to liken the entire structure to a single massive mind, speaking to itself in a holographic and self-reflexive way. If this model were applied to Aleph, the diagnosis would have

revealed a schizophrenic physicist/mystic, compulsively obsessed with the idea of codifying the twin universes of mind and matter into a single massive body of semiformal systems theory. The early days of Aleph, like those of most creative projects, were marked by feverish bouts of prolific brainstorming, and by an overly-broad definition of our interests and activities. There were no hierarchies and no official titles, just various "electronic citizens" adding the contents of their minds and hearts to this strange portable public forum. In general, Mitch Porter provided the basis of our scientific research, while I dug up the arcane works of mystics and Heath Rezabek ("free agent .rez") rallied the creative forces of Alephian text-artists.

Among the various tasks Aleph set upon were the Fixion project (an exercise in cooperative writing and creative role-playing, devised by .rez), the study of Applied Memetics (the means by which ideas spread from mind to mind), the creation of The Alephian Lexicon (a huge collection of useful terms and redefinitions from the realms of philosophy, science, and mysticism), Alpha/Omega (Mitch's idea for a global event heralding a huge worldwide networked Millennial Climax), Gaia 2000 (an experiment in the creation of real-life organizations based upon our fictional structures), and the formation of the HyperQabalah (an interactive Qabalistic Tree of Life). Ultimately, the Alephians assumed, this massive body of knowledge would result in the creation of what we called the AAleph—a relativistic online model of the universe. Unrealistic? Perhaps. Insane? Probably. But none of this stopped us from working. Aleph was unleashed upon the unsuspecting Net in May of 1993; by the end of July, our archives contained over three megabytes—800 pages—of dense and self-reflexive text, loosely referred to as The Alephian Current.

## Weird Sister

In the first week of August, the list began receiving essays and letters from someone who called herself "Doctress Neutopia." She immediately proved herself to be a prolific writer, highly imaginative and not prone to holding back her opinion. The writings of The Doctress dealt consistently with issues of sexism and the oppression of females by males, and were generally of tremendous length and great personal investment. In shorter e-mailed notes, she spoke of adding a love saga to the rapidly-developing Fixion project. She also told us she was looking for "the A/O (Alpha/Omega) Man," who would join her in rule over the Earth when the Lovolution came. (In an early letter to me she asked if *I* was that man; a little stunned, I bounced back Mitch's address, figuring she was looking for the author of A/O.) Day after day, her writings poured in, screenful after screenful. She seemed to care little for our already-developing lines of work, preferring instead to stick to her own campaign and agenda. For a week or so, her words seemed illuminating in a unique way; her over-the-top, egocentric, dyslexic metaphor seemed intentionally postmodern, and her research skills were undeniable (even if her prose tended toward hyperbole). The Doctress was unconditionally welcomed to the list. Relatively speaking; Aleph had found another weird sister.

Upon acceptance, her tone changed almost immediately. As I soon found out, she was not one to submit herself to the communal-editor approach we fostered; the slightest criticism of her words would send her into a rant, and heaven forbid if that criticism come from a *male*! Beneath her hard-edged rhetoric, it soon became apparent that another thread existed: as she often revealed in her own poetic works and letters to the list, The Doctress was actually an extremely sensitive person who had been imprinted by various unfortunate incidents throughout her life. She was very badly wounded, very much in pain, and quite possibly out of touch with reality. As her words rolled in by the bufferload, she began painting for us her feverish visions of a "Neutopian Society" with herself at its apex; a world in which males were subservient

to females. The issue of subservience was key—there had always been Alephians of both sexes, but sexism *per se* had never been an issue. Some Alephians became outraged by her continued presence, and several of them unsubbed from the list. Others supported her views but tried to convince her to adopt a more sensitive viewpoint. Some of us ignored her posts altogether, and, in their own naive neohumanistic ways, some of us assumed they could help her.

## War in Cyberspace

Alone or in groups, the Alephians tried every possible means of communication to rationalize with "The Doctress." Her writings were analyzed, edited, ignored, attacked, defended, spoofed and satirized; she was contacted in private e-mail as well as in our public forum, and even spoke to a few of us live, via IRC channels and telephone. In synchronous dialogue—such as the five-hour telephone conversation we once had—she was quite personable, even *engaging*. In realtime, it was possible to nip potential misunderstandings in the bud, saying, "No no, I meant *this*," or "Why did you use *that* word?" But back online, driven by the juggernaut of her cybernetic persona, she again assumed her autocratic and ignominious tone.

Publicly, diehard Alephians tried to ignore her posts. In private e-mail, the same group worried about the future of the list. Certainly, we could have removed her "forcibly," but we didn't want to do that. Moderating is a tricky job, requiring a strange new social skillset (one which I'm not really sure I've developed completely), but the general consensus was: (a) if we kicked her off the list we would be acting like the "fascists" she often accused us of being, and (b) if *anyone* could actually *help* this woman, it would be us. While we brainstormed and deliberated, further arousing Neutopia's temper, the Alephian Current was becoming a socio-sexual flamewar. Things move very quickly in Cyberspace, and a reputation for a low signal-to-noise ratio can be a real subscription-killer.

On August 20th, Neutopia played out her gambit. I logged on that morning to find another e-comm from The Doctress in my .inbox, ominously entitled, "Urgency—THE COUP OF ALEPH." I opened the message and read. A sick feeling crept over me as I took in The Doctress' words; *she was attempting to take over the list!* In paragraph after ranting paragraph, Doctress Neutopia was dictating what the list was about, what our goals had become, and what our individual functions were, where we would strike, whom we would contact, and—of course—eliciting her call for the deific A/O Man who would sit beside her on the planetary throne. Her massive agenda called for investigations of NASA and military science institutes, microbiological research projects on the state of Gaia, phreaking and cracking maneuvers intended to "free" the Internet, public lectures and performance events, reports on extraterrestrial contacts, video documentaries, book summaries, activist mobilizations, recruitment projects, press releases and publicity campaigns, all aimed at the Neutopian goal of taking over the world.

For the next few weeks, what traffic remained on Aleph stumbled and halted. The Fixion project did its best to continue, but our dwindling population was making it difficult to maintain our original velocity. A month after the "coup," Neutopia was still arguing with nonsupporters, and our ongoing research projects had ceased entirely. Our subscription base was next to nil. Having spent hundreds of dollars in Aleph-related phone calls over the previous few weeks, Mitch and I were both seriously indebted to our respective phone companies and were having difficulty getting online. Soon after, free agent .rez and the Fixioneers moved on to another site (fixion-list@netcom.com), and the Alephian Current, for all intents and purposes, came to an end.

## Alephtermath

The Doctress hung around for a little while after the dissolution but apparently found the lack of company disenchanting; she eventually discovered other elists, and recently began posting to Usenet. The funny thing is; sometimes I miss her.

Aleph still exists today, although lately it's held little more than a slow-motion conversation on "quantum consciousness" and a number of press releases reflected there from other lists. Mitch and I are back online regularly again, although neither of us is racking up linetime like it's going out of style (quite the opposite, in fact). As we enter the summer of 1994, a number of old and new friends have come around; there are signs that the Current may be unceremoniously returning to life, but cautiously, and in a totally new form (the cliché might be "sadder but wiser"). We haven't yet determined our "policy" regarding subscription denials, killfiles, and electronic coups ;-), but like the futuristic telecommunications commercials constantly remind us: "We will." The moral of our story? Draw your own. Perhaps simply, "Caveat Listor."

*To subscribe to the Aleph Elist, send e-mail to:*

Majordomo@gossip.pyramid.com

*Text body:* subscribe aleph

*To subscribe to the Fixion Elist, send e-mail to:*

fixion-list-request@netcom.com

*Subject:* subscribe fixion-list

---

After years of subscribing to many active mailing lists and reading very active newsgroups, in self-defense I have developed a set of personal rules to guide my own activities in this regard and to keep my head above e-mail water.

## Five Major Guru Rules for Reading All That Mail

Everyone's rules may be different and may reflect their own personality, but it's been my observation that people I know who read and track a lot of information in newsgroups and mailing lists seem to have several of these rules in common. They are presented here in order of importance (in my opinion). In some cases these may seem like common sense, but I've seen too many competent computing professionals bogged down by their e-mail reading to believe that even the most commonsensical rules are easily adopted.

### 1. Don't Read It All

There is an overwhelming urge, when presented with so much information, to try to absorb it all. A colleague of mine once decided that he should try to read all of Usenet news (back when there were less than 1500 groups) and even he, who was notorious for such exploits, gave up early in the process. The bottom line is that you don't have to read it all to be reasonably informed. Even the most professional of discussion groups has e-mail "noise" mixed in with the informative messages. Even some informative

messages will not be of interest to you. I scan subject lines fairly thoroughly and delete many mail messages from my mailbox before reading them. You can use e-mail rules facilities to automatically discard messages such as those with the notorious unsubscribe in the subject field. The same is true for news readers that support kill files. By being very selective in what you read, you can cover a lot of territory without devoting your life to e-mail. Remember that in most cases, you can search a mailing list archive for a particular topic after the fact if you do miss something important the first time around.

## **2. Read Your Mail Every Day; Suspend Subscriptions When Away**

If you are like me, with subscriptions to more than several active mailing lists, not reading mail for even one day will result in an uncomfortably large number of messages waiting for you on the next day. The best way to keep up with mail is to read it every day. I know several people who are continually catching up on their e-mail because they let one or more days go by without their attention. It's a simple rule, but it takes a lot of discipline for some people to follow it. One very nice feature of e-mail is that it is out-of-time communication, so it does not usually matter when or how often you read it. Sneaking in four fifteen-minute segments in one day is sometimes more productive than spending a whole hour at once. Also, my general rule is that if you are going to be gone a week or more, it's more productive to suspend delivery of your most active mailing lists than to weed through the volumes of messages when you get back. If you find an interesting thread occurring, you can usually retrieve list archives to catch up on the start of the "conversation."

## **3. Be Selective in Your Subscriptions**

Find the mailing lists that provide the best information for your needs and stick with them. Most mailing list beginners, enthusiastic about the medium, have a tendency to over-subscribe, and even veterans sometimes have trouble letting go of a list that has previously been useful but is no longer relevant. Mailing lists can become part of your life and sometimes it's just hard to say goodbye. There are many good reference lists that can serve as pointers to information on a wide range of topics (new-list and net-happenings are a couple of examples). Sometimes you can fulfill an information need by searching a list's archives or by posting to a list of which you are not a member and asking for replies directly to your address.

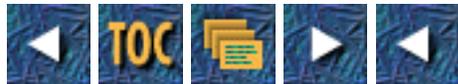
## **4. When Possible, Read News Rather Than Mail**

Mailing-list mail usually comes whether you want it to or not. Usenet news is there, waiting to be read. It doesn't count against your space quota if you skip a few days of reading, and you can easily read as much or as little as you wish. Many news readers have fairly sophisticated mechanisms for managing messages, making news reading more efficient than e-mail access. Many mailing lists are echoed to news and vice versa. The bit.listserv hierarchy is dedicated to echoing Bitnet Listserv mailing lists, and you can participate in the discussions via news without having to subscribe and then sign off later. This makes it much easier to visit lists on a temporary basis.

## **5. Organize Your Reading Process**

Even if you are not by nature an organized person, you should try to organize your e-mail reading. I have my mail options set so that mail is automatically sorted by the From: field and the Date: field. This groups messages by sender or mailing list and places them in chronological order. I further reduce my view of

messages to read only one group at a time. This can be accomplished in some mail programs with a **SELECT** command or by using a mail preprocessor or rules facility to file the mail into different folders based on the header fields. If you can further select messages based upon subject thread, you can further increase the efficiency of the process—especially for groups or lists that have frequent ongoing discussions on a particular topic. I find the thread feature of my news readers to be one of my favorite features. However you do it, an organized approach to reading news and mail will cut down on the amount of time needed and increase the amount of information you can track.





- [6](#)
  - [Faxing from the Internet](#)
    - [Free Faxing Services](#)
      - [TPC.INT Remote Printing](#)
      - [The University of Minnesota Fax Service](#)
      - [rabbit.rgm Sacramento Fax Service](#)
      - [Swedish University Network](#)
    - [Commercial Services](#)
      - [FAXiNET](#)
      - [InterFax](#)
      - [RadioMail](#)
      - [Unigate](#)
      - [FaxLinq](#)

## 6

# Faxing from the Internet

--by [Kevin Savetz](#)

The fax machine, like electronic mail, has helped revolutionize how we work. Businesses large and small send and receive faxes (short for facsimiles) any time a copy of a document is needed quickly on the other side of town, or the other side of the planet. Faxing has some advantages over electronic mail—you can easily send images and, more notably, your recipient doesn't need to be wired to the Internet to receive a fax. A phone line and an inexpensive machine or fax modem is all it takes.

Fax machines are devices that can send a copy of a printed page over a telephone line to another fax machine. Standard faxes work by "scanning" the page to transmit for light and dark areas. The machine converts the tonal values to sounds and sends them over a phone line to a recipient machine that converts the sounds back into light and dark tones and prints them on paper.

Although fax machines have been in popular use since the 1970s, the technology for turning the printed page into sound was invented hundreds of years ago. Not until the invention of the telephone and the proliferation of computers and microchips, did fax machines become a practical method of information exchange.

Here's the best part: because you're on the Internet, you don't even need a fax machine to send faxes. That's right—you can send a fax without using a fax machine or one of those fancy New-Age fax modems. There are several services for sending a fax via electronic mail. So, if you need to send a note to a friend or an associate who's not online, you can do so without logging out of your Net account.

---



**Note:** Although you may think fax machines are just a staple for big business, small offices and home-based businesses have also discovered their usefulness. If the current trend continues, by 1995, 30 percent of all U.S. homes will have a fax machine.

---

Some of these services are free, run as experiments or by hobbyists. Others are commercial services that charge a fee for every fax you send. As you might expect, the quality and coverage of these services can vary considerably. The free services all have limited coverage areas: most span only a city or a few area codes. The commercial services often cover wider areas, but are (of necessity) more expensive.

So, how do you send a fax via e-mail? Read on. The following information discusses eight services that will let you do just that. And, since this is e-mail, you can send your fax to multiple destinations, or even a combination of fax machines and regular e-mail recipients. Be sure to notice that the procedure is not standardized because each service operates a little differently from the rest.

All of these services let you send plain old ASCII text. Some of them can even handle PostScript documents, should you want to send a fax with fancy formatting or graphics. But unlike a "real" fax machine, all of these services can only send information that is stored in your computer: that hand-written letter from Aunt Zelda and your kid's finger-paint artwork just can't be sent in this manner (unless, of course, you scan it in to your computer and convert the graphic to a PostScript document—I don't even want to *think* about that.). If you just want to send unformatted documents created with your word processor, you've got it easy.

## Free Faxing Services

You may find that the following e-mail-to-fax services are a little funky, sometimes unreliable, or downright unusable. But the upside is, they're free! (To be honest, you'll probably find that they all work pretty well, but I do hear the occasional complaint about unreliable services. I just thought I'd warn you.)

### TPC.INT Remote Printing

The most (in)famous fax-from-the-Internet service is the brainchild of Carl Malamud (the creator of Internet Talk Radio) and Marshall Rose. They're doing research on how to integrate special-purpose devices, like facsimile printers, into the fabric of the Internet.

The official name for this project is "an experiment in remote printing", or TPC.INT. The experiment is a good hack, enabling Internet users to easily send faxes via electronic mail. It works simply enough—send electronic mail to a special address, and soon after (if your recipient's fax machine is in a covered area) out comes a freshly-minted fax. After the deed is done, you will receive electronic mail telling you if your fax was successfully sent or not.

---



**Note:** What does TPC stand for? Go rent the film *The President's Analyst*, Paramount Pictures, 1967.

---

A variety of companies, institutions and citizens linked to the Internet have joined the experiment by linking a computer and fax modem to the Net. When an organization joins the remote-fax service, it specifies which areas it is willing to send faxes to. In most cases, an organization will allow faxes to be sent to any machine that is a local call from its fax server.

As a result, with just a smattering of organizations participating, you can't send a fax just anywhere. When you send an e-mail fax message, you (naturally) must include the phone number of the recipient's fax machine. The TPC.INT server looks at the phone number and decides if any participating fax machines cover the area to which you want to send a fax. If so, your message is routed to the appropriate machine for "faxation." Otherwise, you will receive electronic mail with the disappointing news that your fax couldn't be delivered.

Although the service is free, the service's creators are investigating ways to help the organizations recover the cost of sending faxes. So far, this has been primarily through little advertisements—sponsorship messages touting the good name of the organization operating the fax node—on the title page of the resulting fax.

To send a fax by e-mail, send this message:

To: `remote-printer.info@phone_number.iddd.tpc.int`

where *info* contains information for the cover page. In *info*, / is turned into a line break and \_ is turned into a space. For example, the address:

To: `remote-printer.Arlo_Cats/Room_123@12025551212.iddd.tpc.int`

Would send a fax to +1-202-555-1212 with the cover page:

Please deliver this facsimile to:

Arlo Cats

Room 123

---



**Note:** There's another way to address faxes which works more reliably sometimes. In this format, the phone number is backwards and the numbers are separated by periods. It looks ugly, but it works:

To: `remote-printer.Arlo_Cats/Room_123@2.1.2.1.5.5.5.2.0.2.1.tpc.int`

---

For a FAQ about the TPC.INT service, send electronic mail (any subject line and message body will do) to `tpc-faq@town.hall.org`. For a list of the areas where faxing is currently available with the service, e-mail `tpc-coverage@town.hall.org`.

There is also a mailing list for discussion of the service. To subscribe, send e-mail to

tpc-rp-request@aarnet.edu.au. Administrative questions about the project should be directed to the following:

tpc-admin@town.hall.org

---



**Note:** There is also a World Wide Web page that serves as a central location for information about the TPC.INT fax service. If your Web browser can display forms, you can even fill out a form and send a fax interactively. Use your favorite Web browser to connect to

<http://linux1.balliol.ox.ac.uk/fax/faxsend.html>

If you want to know the gory technical and implementation details of the TPC.INT experiment, there are three RFCs you might want to read:

RFC 1530 covers the general principles and policies of the experiment.

RFC 1529 covers administrative policies

RFC 1528 covers technical procedures for implementation

Here's an abbreviated list (as of this writing) of the places you can fax using the TPC.INT service. Many times, the entire country or area code is not covered, so check the current coverage list for a complete breakdown. You can get a complete and current list by sending e-mail (any subject line, any message body) to the following:

tpc-coverage@town.hall.org

### **Outside the United States and Canada**

Australia (+61)

Denmark (+45)

Germany (+49)

Japan (+81)

Korea (+82)

New Zealand (+64)

Portugal (+351): Lisbon (+351-1)

Sweden (+46)

United Kingdom (+44):

### **Canada and the United States (+1)**

+1-202: Washington, D.C. (except Congress)

+1-212: New York

+1-301: Maryland

+1-310: California

+1-313: Michigan

+1-317: Indiana  
+1-408: Sunnyvale, California  
+1-410: Maryland  
+1-412: Pittsburgh, Pennsylvania  
+1-415: San Francisco, California  
+1-416: Toronto  
+1-508: Massachusetts  
+1-510: Central California  
+1-516: New York  
+1-608: Wisconsin  
+1-613: Massachusetts  
+1-617: Massachusetts  
+1-703; Virginia  
+1-714: Irvine, California  
+1-718: New York  
+1-800: Toll-free calls  
+1-813: Tampa  
+1-818: California  
+1-819: Quebec, Canada  
+1-905: Markham, Canada  
+1-909: California  
+1-917: New York, New York  
+1-919: North Carolina

---

## **The University of Minnesota Fax Service**

The University of Minnesota operates a fax gateway which allows students and staff to send faxes anywhere. Even if you don't go to UMinn, you may use the service to send faxes to folks at the University and exchanges local to the campus.

The structure for e-mailing a fax is:

To: /pn=John.Doe/dd.fax=234-5678/@fax.tc.umn.edu

Put your recipient's name, with a period between the first and last names, after pn= and put the seven digit fax number after the characters dd.fax=. This name will be printed in the To: field on the fax cover page. The area code for the University of Minnesota is +1-612, so you don't need to supply an area code.

For more information, send e-mail to [ccs@maroon.tc.umn.edu](mailto:ccs@maroon.tc.umn.edu).

## ***rabbit.rgm* Sacramento Fax Service**

This service is a feature of a Sacramento, California-based bulletin board system. You can use it to send faxes to areas that are a local call from Sacramento (that's area code +1-916) including the California State Legislature. This service is run as a hobby and is connected to the Internet via UUCP, so it can take from 12 to 24 hours for your fax to be delivered or for the help files to reach you. It does not support multiple addressing—only one fax number per message. It also does not send a cover page, so be sure to

start your message with a note directing it to someone's attention. It will truncate faxes longer than two pages (that's 132 lines). To use this fax service, send e-mail:

To: [faxline@rabbit.rgm.com](mailto:faxline@rabbit.rgm.com)

Subject: local (7 digit) phone number, without area code

Body: <text of fax>

For complete usage information, send e-mail

To: [request@rabbit.rgm.com](mailto:request@rabbit.rgm.com)

Subject: 052

For a list of some legislators' fax numbers in the Sacramento area, send e-mail:

To: [request@rabbit.rgm.com](mailto:request@rabbit.rgm.com)

Subject: 050

## Swedish University Network

The Swedish University Computer Network (called sunet) operates a national fax service that can be used by anyone at no cost. Users in Sweden can use it to send faxes all over the world, but users outside Sweden can use it for telephone numbers anywhere within Sweden.

To send a fax to Arlo Cats (wow, this cat gets around, doesn't he?) at +46-87654321 (that's international notation for Sweden, phone number 08/765 43 21) send e-mail:

To: [Arlo\\_Cats@F087654321.fax.sunet.se](mailto:Arlo_Cats@F087654321.fax.sunet.se)

Note that you will always have to preface the phone number with the letter F, for *fax*. For more information, e-mail

[faxmaster@fax.sunet.se](mailto:faxmaster@fax.sunet.se)

## Commercial Services

The following services charge for use, but you may find they're easier to use because they cover much larger areas than any of the free services. You can pick one service and use it as a sort of "one-stop-shop" for all your e-mail-to-fax needs, without worrying whether your fax's destination is in a covered city or area code.

### FAXiNET

Another fax-by-e-mail service is FAXiNET, which lets you send ASCII text or PostScript documents to fax machines worldwide. FAXiNET can currently send faxes to more than 50 countries and plans to add more. The company can also receive faxes for you, which will be delivered to you via electronic mail.

Accounts for individuals cost 75 cents per page, plus a one-time \$20 activation fee. Additional services,

including adding your custom logo and signature to your faxes, are available at extra cost. Corporate accounts are also available. More information is available from AnyWare Associates, FAXiNET, 32 Woodland Road, Boston, MA 02130. (617) 522-8102. E-mail: sales@awa.com

## **InterFax**

InterFax enables you to send faxes via e-mail within the US or internationally. InterFax is a fee-based service (billed to your credit card) but, unlike the services already listed, InterFax lets you send faxes anywhere, not just select locations. As of this writing, InterFax costs \$5 per month, which includes your first five fax pages. Additional pages cost 50 cents each. There is a one-time sign-up charge of \$25. For further information, send e-mail to faxmaster@pan.com, or contact InterFax at PO Box 162, Skippack, PA 19474 USA. Phone: (215) 584-0300. Fax: (215)584-1038.

## **RadioMail**

This e-mail-to-fax service lets you send faxes to just about any country you can name. Prices vary accordingly—it's much less expensive to send a fax to a "well-connected" country (like Canada and Sweden) than less-connected places like Laos and the Solomon Islands.

All fax messages are charged on a per-page basis. Faxes to the U.S. and Canada are billed at the rate of 99 cents per page. The rate for international faxes depends on the destination country: \$0.99, \$1.99, \$3.99, or \$4.99 per page.

RadioMail stands out because the service doesn't require that you have access to a telephone line—it can work using a wireless modem. To become a RadioMail subscriber, users need a wireless modem and the special RadioMail software. According to the company, "RadioMail supports DOS computers, Macintosh computers and HP Palmtops in addition to providing one-way communications on the Newton." For more information, send e-mail to radiohelp@radiomail.net.

## **Unigate**

Unigate is another pay-for-use service that enables you to send faxes to Russia and the Commonwealth of Independent States. (It offers other interesting services as well, including the ability to send e-mail that turns into postal mail and the ability to receive faxes as electronic mail.)

Most of us probably don't need to send faxes to Russia, but if you do this is probably cheaper than however you're doing it now. Fax service from USA to Russia (or back) is \$1.59 per page; from Russia to Canada is 1.79 per page. For more information, send e-mail to:

yuri@atmos.washington.edu

## **FaxLinq**

This service doesn't quite fit into the e-mail-to-fax arena, but it's close enough to mention here. With FaxLinq, you may receive facsimile messages as e-mail. Your correspondents send a fax to the service's machine, which is converted to a TIFF file and sent to you in a MIME-compliant e-mail message. (You must be able to handle MIME e-mail and view TIFF files—you can't receive faxes as ASCII text.) FaxLinq is a one-way ticket: it doesn't handle e-mail-to-fax transmissions.

There is an annual subscription fee of \$39, which includes 10 pages of facsimile transmission. Additional pages received cost \$1 per page. FaxLinq uses *deposit* accounts: you must pay in advance for any faxes you expect to receive. Should a fax be received for you when your account balance is not sufficient to cover the number of pages received, you will be notified by e-mail.

For more information, e-mail [info@antigone.com](mailto:info@antigone.com) or write: Antigone Press, 1310 Clayton Street, Suite 15, San Francisco CA 94114.

**Note:** For updates on how to send faxes via e-mail, check out my FAQ called (appropriately enough), "How can I send a fax from the Internet?" This file is posted twice monthly to the Usenet newsgroups [alt.internet.services](mailto:alt.internet.services), [alt.online-service](mailto:alt.online-service), [alt.bbs.internet](mailto:alt.bbs.internet), [alt.answers](mailto:alt.answers) and [news.answers](mailto:news.answers). You can also receive it via electronic mail by sending e-mail in the following manner:

To: [mail-server@rtfm.mit.edu](mailto:mail-server@rtfm.mit.edu)

Subject: *subject line is ignored*

Body: send /usenet/news.answers/internet-services/fax-faq

By the way, George Pajari maintains another interesting fax FAQ list. That document is concerned with fax standards and computer-based faxing systems but does not focus on faxing from the Internet. Here are some of the topics it covers:

Can I use my \* data modem to send/receive faxes?

Can my fax modem transmit data?

How can I fax PostScript or PCL documents using computer-based fax?

How can I view incoming faxes on my computer?

How can I print incoming faxes on my computer?

Can fax modems also handle data or voice calls?

What resolution are fax images?

Can I take a fax file and edit it?

Is there a standard program interface (API) for fax communications?

How can I share my single phone line with voice, fax, data, and so on?

You can grab a copy from the Usenet groups [comp.dcom.fax](mailto:comp.dcom.fax), [alt.fax](mailto:alt.fax), or [news.answers](mailto:news.answers). It's also available via FTP at [rtfm.mit.edu/pub/usenet/news.answers/fax-faq/](ftp://rtfm.mit.edu/pub/usenet/news.answers/fax-faq/)\*





- [7](#)
  - [FTP](#)
    - [Those Pesky FTP Servers](#)
    - [Batching It](#)
      - [Batch FTP on VMS](#)
      - [Batch FTP on UNIX](#)
    - [Gopher FTP](#)
      - [Some Gopherable FTP Sites](#)
      - [A Final Word on Gopher FTP](#)
    - [FTP Through a WWW Browser](#)
    - [The File's in the Mail](#)
      - [BitFTP](#)
      - [FTPMail](#)
    - [NcFTP: An Alternate Unix Client](#)
    - [Wuarchive FTPD](#)
      - [The Future of FTP](#)

# 7

## FTP

--by [Philip Baczewski](#)

### Those Pesky FTP Servers

Anonymous FTP is one of the most useful Internet services, and at the same time it can be one of the most frustrating. Because some anonymous FTP servers are so comprehensive, they are also very popular. As you may have already found out the hard way, sites like wuarchive.wustl.edu often are inaccessible during daylight hours, even though some support as many as 200 simultaneous connections. Even mirror sites are often quite busy, and because the organizational structure may be different, it's not always easy to find your way around. Fortunately, there are several solutions for retrieving files from popular sites, or even obscure ones. The simplest solution is to stay up late to do your FTP sessions. However, for those who do sleep, some other ideas are offered here.



**Stupid UNIX FTP Tricks:** The following commands, which will work on most UNIX systems, let you enjoy some shortcuts and time-savers during an FTP session:

1. Display text files to the terminal.

It's very handy sometimes to be able to display a short readme file directly to your terminal, rather than downloading it and then invoking a file viewer. The following command will get a file and transfer it to the console rather than to the directory

space:

```
ftp> get readme.txt -
```

This command is a normal ASCII transfer, but the hyphen represents the console as the file's destination.

## 2. Browse text files within FTP.

"Ah yes," you say, "but what if I want to examine a longer text file and I don't want it to scroll off my screen?" The following command enables you to pipe a transferred file to the more program, which enables you to page up, page down, and search within the file.

```
ftp> get readme.txt |more
```

There is no space between the pipe symbol and more.

## 3. Browse a directory within FTP.

If you have ever stumbled across a directory with hundreds of entries, you will understand the significance of the following command:

```
ftp> get . |more
```

This transfers the listing of the current directory (.) and pipes the output to more. Instead of watching those filenames scroll off your screen for a couple of minutes, you can search the listing for a particular file or scroll the file at your own pace.

---

## Batching It

One method for retrieving files from busy sites (while still getting some sleep) is to make use of the *batch*, or *timed-execution*, facility of your host computer system. This is possible on most UNIX and DEC VMS systems, but check with your system administrator (if that's not you) to be sure. The following examples are not necessarily the only or most elegant solutions, but they may give you some ideas to build upon.

### Batch FTP on VMS

On VMS, a DCL script can be used to submit and schedule a batch job to accomplish an FTP session. The following example uses a VMS DCL script to invoke FTP and a data file that contains the FTP commands you want to execute. As you can see, it doesn't take many lines to get the job done. This script was created on a system running Wollongong TCP/IP, so a version for your system might vary slightly depending upon the TCP/IP software that's installed.

The following statements should be saved in an executable file named FTP.COM. Comments begin with an exclamation point.

```
#!/FTP.COM - Batch FTP on a VMS System
```

```
#!/Change to the download directory
```

```
$ SET DEFAULT DUA0: [myid.kermit]
```

```
#!/FTP command input comes from a file in the ftpbat directory
```

```
$ DEFINE/USER SYSS$INPUT DUA0: [myid.ftpbat]FTP.DAT
```

```
#!/Set the path to the ftp application
```

```
$ ftp ::= $twg$tcp: [netapps.user]ftp ftp
```

```
#!Invoke the FTP command with a target address
```

```
#!This example is the server at Columbia University
```

```
$(ftp.cc.columbia.edu)
```

```
$ FTP 128.59.39.2
```

A typical data file with your FTP commands might be as follows. This file would be named FTP.DAT, as seen in the preceding script example. The QUIET and NOINTER commands turn off verbose messages and suppress prompting, respectively.

```
anonymous
```

```
myid@mynode.edu
```

```
CD kermit/b
```

```
QUIET
```

```
NOINTER
```

```
mget ikc*
```

```
QUIT
```

As you can see, the script simply sets the download path, points to the FTP command file, sets a logical variable for the FTP program, and executes a connection. The FTP.COM file can be scheduled for batch execution with a command like the following:

```
submit/noprint/log=DUA0:[myid.ftpbat]ftp.log/queue=batch/after="21:00" ftp
```

Because this command is rather long, you could create a .COM file for it as well, using a hyphen as a continuation character to the next line if necessary. The content of such a file might appear as follows:

```
$submit/noprint/log=DUA0:[myid.ftpbat]ftp.log/queue=batch/after="21:00"-
```

```
ftp
```

Any output from the FTP session will be stored in the file FTP.LOG, in this case located in the ftpbat directory. The FTP.COM file will be executed after 11:00 p.m. You will probably need to check with your VMS system administrator for the name and availability of a batch queue; however, it is likely that such a facility does exist on your VMS system.

## Batch FTP on UNIX

A facility similar to the preceding VMS example is available to you on UNIX when you schedule a shell script to do your FTP commands after business hours. The following example employs a script to initiate the FTP session and another file to hold the actual FTP commands. These two files combine to operate as a batch FTP session on UNIX systems.

```
#bftp -- Batch FTP script thanks to Amos Gouaux (amos@unt.edu) for
```

```
#his loan of this example batch ftp shell script.
```

```
#!/bin/sh
```

```
set e
```

```
case $# in
```

```

1)

    if [ ! f $1 ]

    then

        echo "$0: file \"$1\" not found."

        exit

    fi

    ;;

*)

    echo 'Usage: bftp site(:ext)'

    echo 'Whereas, "site" is a job file having the same name as the'

    echo 'ftp site and ":ext" is a filename extension that may be'

    echo 'appended to the end of "site".'

    exit

    ;;

esac

touch bftp.log

echo "Processing script \"$1\"." >> bftp.log

ftp dinv << _eof >> bftp.log 2>&1

    open `echo $1 | sed e 's/:.*$//'`

    user anonymous myid@mynode.edu

    `cat $1 | sed e '/^[^az]*\#.*$/d' | sed e 's/\#.*$//'`

    bye

_eof

echo >> bftp.log

```

The following is an example of a data file for the preceding script. The name of this file should be the same as the fully-qualified domain name of the FTP server you are accessing (in this example, ftp.cc.columbia.edu). Comments can be placed in this data file by prefixing the line with a pound sign (#).

```
cd kermit/b
```

```
#turn off verbose messages
```

```

verbose
#turn off prompting
prompt
#get all Amiga C-Kermit files
mget ikc*
QUIT

```

This script uses the name of the intended FTP site for the name of the file containing the FTP commands you want to execute at that site. This clever bit of programming (I wish I could say it were mine) saves some lines of code and also can help organize and easily identify command files that will be used on a regular basis (for regularly posted software updates, archive files, and so on).

On most UNIX systems, you can schedule this batch FTP script to execute at a specified time by using the at command. (Once again, you may need to check with the system administrator to gain authorization to use the at command.) A typical command might be as follows:

```
at 11:00pm today bftp ftp.cc.columbia.edu
```

Check the at man page for all of the options available with this command. If you want to regularly schedule batch FTP sessions, you may want to take advantage of the UNIX cron facility. The usual caveat applies—you may need to check with your system administrator to gain access to cron (and to create a crontab file).

One nice thing about computers is that they can work at hours when you don't. If you are a script-programming guru on either VMS or UNIX, you can probably come up with several enhancements to the preceding offerings. If not, however, these scripts can still be quite useful and may be able to save you some time as well as sleep.

## Gopher FTP

Once again, Gopher comes to the rescue. An anonymous FTP interface is a useful addition to the multitude of Internet services and information available through Gopher. Many popular FTP sites provide access through a Gopher interface. Because of the way Gopher operates, it has become the preferred method of access for a number of sites. An anonymous FTP session establishes a static connection to the FTP server, and as long as you are connected to that server, one connection is unavailable to another FTP user. This will matter to you if the server supports 200 connections and you are connection number 201.

Gopher works on a dynamic connection model. An Internet connection is made only while the Gopher client is reading requested information, whether that be a Gopher menu, a file, or an FTP-able software archive file. In other words, while you retrieve an FTP directory you are using a connection, but while you are perusing that directory no load is placed on the remote server. Because of the efficiency of accessing FTP archives through Gopher, it's often possible to gain access through that method when all "traditional" FTP ports are occupied.

### Some Gopherable FTP Sites

A check of the University of Minnesota Gopher (gopher.tc.umn.edu) shows a number of familiar names in the "FTP Searches/Popular FTP Sites via Gopher" menu. The following list includes such esteemed names as Boombox, SUMEX, and wuarchive. (When you can go by one name, you know you've hit it big.) More exploration or a Veronica search may also turn up some of your popular sites.

#### Popular FTP Sites via Gopher

1. Read Me First

2. Boombox Home of Gopher and POPmail/
3. Case Western Reserve University FREENET/
4. Indiana University Mac Gopher Client App (beta)/
5. Indiana Windows Archive/
6. Interest Group Lists/
7. Internet Resource Guide (tar.Z files)/
8. Latest Disinfectant (ftp.acns.nwu.edu)/
9. Lyrics/
10. NCSA Home of NCSA Telnet/
11. National Science Foundation Gopher (STIS)/
12. Newton Archives at Johns Hopkins University (bnnrcsrv.med.jhu.edu../
13. OCF Document Archives/
14. OSSIS Info Archives (slow)/
15. SUMEXAIM Archives (Includes InfoMac: a large collection of Mac../
16. Scholarly Communications Project of Virginia Tech/
17. Software Archives at MERIT (University of Michigan)/
18. Sonata NeXT software archive (sonata.cc.purdue.edu)/
19. Supreme Court Rulings (CWRU)/
20. UIUCCSO Home of the qi server (CSO phonebook software)/
21. University of Utah Mac Gopher Client App (beta)/
22. Usenet University/
23. Washington University Archive (wuarchive)/

After you find a Gopher FTP site, you will probably want to add it to your bookmark file. The following might offer you a head start in this regard. Some sample UNIX Gopher bookmarks are provided for some popular sites.

Type=1+

Name=Popular FTP Sites via Gopher

Path=1/FTP Searches/Popular FTP Sites via Gopher

Host=gopher.tc.umn.edu

Port=70

#

Type=1

Name=SUMEXAIM Archives (Mac software)

Path=

Host=SUMEXAIM.Stanford.EDU

Port=70

#

Type=1

Name=Washington University Archive (SUMEX & other mirror site)

Path=

Host=wuarchive.wustl.edu

Port=70

#

Type=1

Name=OCF (Open Computing Foundation) Document Archives

Path=

Host=ocf.berkeley.edu

Port=70

## **A Final Word on Gopher FTP**

Gopher FTP is not without its limitations. Most archived and ASCII-encoded files will transfer successfully, but executable files may not, depending on the system on which you are using Gopher. Even with a more efficient paradigm, you still might have trouble establishing a connection to some of the more popular sites (in which case, dust off that batch FTP script). On the plus side, however, there are some microcomputer versions of Gopher, like TurboGopher for the Macintosh, that will actually enable you to retrieve and then decode a file all in one operation. The other handy thing about using Gopher as an FTP access method is that you can easily display any text files (like README files, for example) directly to your terminal, eliminating the need to download and view a file as a separate operation.

## **FTP Through a WWW Browser**

Don't forget that if you can access an FTP site through Gopher, you can also use your favorite World Wide Web client to access and download files. If you know the host, a URL (Universal Resource Locator) would look similar to the following:

`gopher://any.gopher.host.com/`

This URL will connect at the root starting point, and you can then traverse the Gopher structure. Most Gopher installations use port 70 for incoming connections, but some might use an alternate number. You can specify such a different port as follows (using 9999 as an example):

```
gopher://any.gopher.host.com:9999/
```

You can also access FTP sites directly using the following URL format:

```
ftp://any.ftp.host.com/pub/any/directory
```

or

```
file://any.ftp.host.com/pub/any/directory
```

Most Web browsers enable you to view text files directly and automatically download any binary files you select.

## The File's in the Mail

There are two main facilities for retrieving files from anonymous FTP sites. BitFTP is maintained by Princeton University as a service for Bitnet (U.S.), NetNorth (Canada), and EARN (Europe) nodes that do not have TCP/IP access to the Internet. FTPMail is an unsupported facility at gatekeeper.dec.com and several other sites on the Internet. FTPMail was written by Paul Vixie. Both facilities enable you to send a mail message containing FTP commands and have any resulting output mailed to you at your return e-mail address. Both are facilities of last resort.

In concept, FTP by e-mail sounds good, but some of the charm is taken away when you realize that a 1MB file sent in pieces will generate at least ten separate incoming mail messages. (Both facilities will break large files into multiple mail messages, and both have a limit of about 100,000 characters per file part.) Those mail messages will have to be downloaded, merged back together, and then possibly uuencoded. The following notes on both facilities may make your use of them more efficient. More information on either one can be obtained by sending the command HELP to the FTP server address.

### BitFTP

BitFTP can be accessed by sending mail to the address `bitftp@pucc.princeton.edu`. It is implemented on an IBM mainframe system, so the commands are a subset of IBM's TCP/IP software. This is an advantage if you need FTP access to a specific mainframe on the Internet, but only have electronic mail as a means of communication. BitFTP supports the ACCT command used on VM/CMS systems, and you can specify EBCDIC as a file transfer mode (to transfer text files from mainframe to mainframe). In addition, BitFTP supports the FTP commands CD, DIR, ASCII, BINARY, GET, LS, MODE (block or stream I/O), PWD, QUIT, SYSTEM (to get the name of the FTP host's operating system), TYPE (a mainframe-specific command supporting various transfer modes), and USER.

Some items to remember when using BitFTP are the following:

- FTP sessions are begun with the command `FTP hostname encoding_type`. *encoding\_type* can be NETDATA (used on NJE networks only), uuencode, or xxencode. By default, NETDATA format will be used, so unless you are on a system that can decode NETDATA (an IBM mainframe and some VAX systems), you will probably want to specify uuencode. An alternate FTP port number can be specified following the host-name field.
- You can specify a username and password by using the command following the FTP command, `USER username password`. If you specify anonymous as the username, no password is necessary and BitFTP supplies your return e-mail address to the FTP server as the password.
- A maximum of five GET commands is allowed per BitFTP request.
- The maximum retrievable file size is 17MB.
- Uuencoded and xxencoded files will be split into separate mail files containing no more than 1,500 62-byte records each (about 90KB).

- BitFTP does not provide a PUT capability.

The following message shows a typical BitFTP log message.

Date: 13Jul94 22:48 EDT

From: Princeton BITNET FTP Server > INTERNET:BITFTP1@PUCC.PRINCETON.EDU

Subj: BITFTP Reply

Sender: bitftp1@pucc.princeton.edu

Received: from pucc.Princeton.EDU by dubimg1.compuserve.com

(8.6.4/5.940406sam)

id WAA26854; Wed, 13 Jul 1994 22:47:23 0400

MessageId: <199407140247.WAA26854@dubimg1.compuserve.com>

Received: from PUCC.PRINCETON.EDU by pucc.Princeton.EDU (IBM VM SMTP V2R2)

with BSMTMP id 8504; Wed, 13 Jul 94 22:47:05 EDT

Received: from PUCC.PRINCETON.EDU (NJE origin VMMAIL@PUCC) by PUCC.PRINCETON.EDU

(LMail V1.1d/1.7f) with BSMTMP id 3782; Wed, 13 Jul 1994 22:47:04 0400

Received: by PUCC (Mailer R2.10 ptf008) id 0699; Wed, 13 Jul 94 22:47:03 EDT

Date: Wed, 13 Jul 1994 22:47:02 EDT

From: Princeton BITNET FTP Server <BITFTP1@PUCC.PRINCETON.EDU>

To: XXXXX.XXX@COMPUSERVE.COM

Subject: BITFTP Reply

\*\*\*\*\*

\* WARNING: Do not download CDIT, which purports to be a utility to \*

\* convert an ordinary CDROM drive into a CDRecordable device. It is \*

\* a very destructive virus. \*

\* \*

\* Use the new BITFTP command "HOWTOFTP" to obtain a copy of RFC1635, \*

\* "How to Use Anonymous FTP". \*

\*\*\*\*\*

> ftp ftp.unt.edu uuencode

> user anonymous

>> FTP.UNT.EDU

<<< Connecting to FTP.UNT.EDU 129.120.1.1, port 21

<<< 220 mercury FTP server (SunOS 4.1) ready.

<<< USER (identify yourself to the host):

>> anonymous

>>> USER anonymous

<<< 331 Guest login ok, send ident as password.

<<< Password:

>> XXXXX.XXX%compuserve.com@pucc.Princeton.EDU

>>> PASS \*\*\*\*\*

<<< 230 Guest login ok, access restrictions apply.

> cd pub/misc

>> CD pub/misc

>>> CWD pub/misc

<<< 250 CWD command successful.

> binary

>> BINARY VARIABLE

>>> TYPE i

<<< 200 Type set to I.

> get quikkwic.exe

>> GET quikkwic.exe QUIKKWIC.EXE.D ( REPLACE

>>> PORT 128,112,129,99,140,55

<<< 200 PORT command successful.

>>> RETR quikkwic.exe

```
<<< 150 Binary data connection for quikkwic.exe (128.112.129.99,35895)
      (32032 bytes)
<<< .
<<< 226 Binary Transfer complete.
<<< 32032 bytes transferred. Transfer rate 32.75 Kbytes/sec.
>>>> "quikkwic.exe" sent uuencoded.
> quit
>> CLOSE
>>> QUIT
<<< 221 Goodbye.
```

## FTPMail

The primary FTPMail site can be accessed by sending mail to [ftpmail@gatekeeper.dec.com](mailto:ftpmail@gatekeeper.dec.com). The help file makes it very clear that FTPMail is not a supported service and that it might be unavailable from time to time. FTPMail supports the following FTP commands: CONNECT, ASCII, BINARY, CHDIR, GET, INDEX, LS, and QUIT.

Some other options to keep in mind:

- The CONNECT command by itself defaults to anonymous FTP access to the FTPMail host. The CONNECT command can also be specified with host, username, password, and account values, using this format:

```
CONNECT host username password account
```

- An alternate reply address can be specified with this command:

```
REPLY address
```

- There is a limit of ten GET commands per FTPMail request.
- Several compression and encoding choices are available using the following commands:

COMPRESS—Compresses binary files using Lempel-Ziv encoding.

COMPACT—Compresses binary files using Huffman encoding.

UUENCODE—Binary files will be uuencoded.

BTOA—Binary files will be encoded using BTOA format.

- The size of the pieces into which long files will be split can be controlled by the command CHUNKSIZE *size*.

The default size is 64,000 bytes. CompuServe will allow e-mail files of approximately two megabytes to be received from the Internet. Other services may vary in that restriction.

- There is a limit of ten GET commands per FTPMail request.
- PUT is not supported.

The following is a status message received after an FTPMail request was made.

Date: 14Jul94 07:30 CEST

From: "ftpmail service on ftpgw1.pa.dec.com" > INTERNET:nobody@pa.dec.com

Subj: your ftpmail request has been received [test]

Sender: nobody@ftpgw1.pa.dec.com

Received: from ftpgw1.pa.dec.com by dubimg2.compuserve.com

(8.6.4/5.940406sam)

id BAA12825; Thu, 14 Jul 1994 01:30:12 0400

Received: by ftpgw1.pa.dec.com; id AA13753; Wed, 13 Jul 94 22:30:10 0700

Date: Wed, 13 Jul 94 22:30:10 0700

MessageId: <9407140530.AA13753@ftpgw1.pa.dec.com>

From: "ftpmail service on ftpgw1.pa.dec.com" <nobody@pa.dec.com>

To: XXXXX.XXX@compuserve.com

Subject: your ftpmail request has been received [test]

XComplaintsTo: ftpmailadmin@ftpgw1.pa.dec.com

XServiceAddress: ftpmail@ftpgw1.pa.dec.com

XJobNumber: 774163809.13751

Precedence: bulk

ReplyTo: <nobody@ftpgw1.pa.dec.com>

We processed the following input from your mail message:

connect ftp.unt.edu

chdir pub/misc

binary

uuencode

get quikkwic.exe

quit

We have entered the following request into your job queue as job number 774163809.13751:

```
connect ftp.unt.edu anonymous ftpmail/XXXXX.XXX@compuserve.com
reply XXXXX.XXX@compuserve.com
chdir pub/misc
get quikkwic.exe binary uncompressed uuencode
```

There are 57 jobs ahead of this one in our queue.

You should expect the results to be mailed to you within a day or so.

We try to drain the request queue every 30 minutes, but sometimes it fills up with enough junk that it takes until midnight (Pacific time) to clear. Note, however, that since ftpmail sends its files out with "Precedence: bulk", they receive low priority at mail relay nodes.

Note that the "reply" or "answer" command in your mailer will not work for this message or any other mail you receive from FTPMAIL. To send requests to FTPMAIL, send an original mail message, not a reply. As shown in the header of this message, complaints should be sent to the ftpmailadmin@ftpgw1.pa.dec.com address rather than to postmaster, since our postmaster is not responsible for fixing ftpmail problems.

There is no way to delete this request, so be sure that it has failed before you resubmit it or you will receive multiple copies of anything you have requested.

There is no way to ask for only certain parts of a file to be sent. If you receive output from ftpmail that seems to be missing some parts, it is likely that some mailer between here and there has dropped them. You can try your request again, but chances are fairly good that if it is dropped once it will be dropped every time.

There is no way to specify that your request should be tried only during certain hours of the day. If you need a file from a timerestricted FTP server, you probably cannot get it via ftpmail.

Ftpmail was written by Paul Vixie while at the DEC Western Research Laboratory and at the DEC Network Systems Laboratory, from 1989-1993.

Ftpmail is not a product or service of Digital Equipment Corporation and no support or warranty is expressed or implied.

#### Ftpmail Submission Transcript

```
<<< connect ftp.unt.edu
>>> Connect to ftp.unt.edu as anonymous with ftpmail/XXXXXX.XXX@compuserve.com
<<< chdir pub/misc
>>> Will chdir to <pub/misc> before I do anything else
<<< binary
>>> Grab files in "binary" mode
<<< uuencode
>>> Mail binaries in "uuencode" format
<<< get quikkwic.exe
>>> get quikkwic.exe binary uncompressed uuencode
<<< quit
>>> Done rest of message will be ignored
>>> checking security of host `ftp.unt.edu'
>>> host `ftp.unt.edu' has IP addr [129.120.1.1]
>>> host `ftp.unt.edu' is also known as `mercury.unt.edu'
>>> host `ftp.unt.edu' is ok
```

End of Ftpmail Transcript

Full Mail Header From Request

From XXXXX.XXX@compuserve.com Wed Jul 13 22:30:06 1994

Received: by ftpgw1.pa.dec.com; id AA13745; Wed, 13 Jul 94 22:30:06 0700

Received: from pobox1.pa.dec.com by inetgw2.pa.dec.com (5.65/27May94)

id AA00119; Wed, 13 Jul 94 22:25:21 0700

Received: by pobox1.pa.dec.com; id AA20713; Wed, 13 Jul 94 22:25:19 0700

Received: from dubimg1.compuserve.com by inetgw1.pa.dec.com (5.65/27May94)

id AA12091; Wed, 13 Jul 94 22:21:54 0700

Received: from localhost by dubimg1.compuserve.com (8.6.4/5.940406sam)

id BAA06790; Thu, 14 Jul 1994 01:20:38 0400

Date: 14 Jul 94 01:17:11 EDT

From: Philip Baczewski <XXXXX.XXX@compuserve.com>

To: <ftpmail@gatekeeper.dec.com>

Subject: test

MessageId: <940714051711\_XXXXX.XXX\_HHB453@CompuServe.COM>

End of Request Mail Header

The following is the actual log of the FTP session (received in less than "a day or so").

Date: 14Jul94 07:58 CEST

From: "ftpmail service on ftpgw1.pa.dec.com" > INTERNET:nobody@pa.dec.com

Subj: results of ftpmail request 774163809.13751 [test]

Sender: nobody@ftpgw1.pa.dec.com

Received: from ftpgw1.pa.dec.com by dubimg1.compuserve.com

(8.6.4/5.940406sam)

id BAA08542; Thu, 14 Jul 1994 01:57:18 0400

Received: by ftpgw1.pa.dec.com; id AA14641; Wed, 13 Jul 94 22:57:02 0700

Date: Wed, 13 Jul 94 22:57:02 0700

MessageId: <9407140557.AA14641@ftpgw1.pa.dec.com>

From: "ftpmail service on ftpgw1.pa.dec.com" <nobody@pa.dec.com>

To: XXXXX.XXX@compuserve.com

Subject: results of ftpmail request 774163809.13751 [test]

XComplaintsTo: ftpmailadmin@ftpgw1.pa.dec.com

XServiceAddress: ftpmail@ftpgw1.pa.dec.com

XJobNumber: 774163809.13751

Precedence: bulk

ReplyTo: <nobody@ftpgw1.pa.dec.com>

Orig messageid: <940714051711\_XXXXX.XXX\_HHB453@CompuServe.COM>

Orig subject: test

Orig from: Philip Baczewski <XXXXX.XXX@compuserve.com>

Orig date: 14 Jul 94 01:17:11 EDT

connecting to ftp.unt.edu...

Connecting to ftp.unt.edu

220 mercury FTP server (SunOS 4.1) ready.

logging in as user=anonymous password=ftpmail/XXXXX.XXX@compuserve.com  
account=...

> USER anonymous

331 Guest login ok, send ident as password.

> PASS <somestring>

230 Guest login ok, access restrictions apply.

> TYPE A

200 Type set to A.

changing working directory to pub/misc...

> CWD pub/misc

250 CWD command successful.

=== getting 'quikkwic.exe'...

> TYPE I

200 Type set to I.

> PORT 16,1,0,6,16,233

200 PORT command successful.

> RETR quikkwic.exe

150 Binary data connection for quikkwic.exe (16.1.0.6,4329) (32032 bytes).

226 Binary Transfer complete.

mailing...

XXXXX.XXX@compuserve.com... Connecting to mailgate.compuserve.com (tcp)...

XXXXX.XXX@compuserve.com... Sent

quikkwic.exe (pub/misc@ftp.unt.edu) (1 part, 44160 bytes) sent to

XXXXX.XXX@compuserve.com

> (end of ftpmail session)

Ftpmail was written by Paul Vixie while at the DEC Western Research Laboratory

and at the DEC Network Systems Laboratory, from 1989-1993. Ftpmail is not a

product or service of Digital Equipment Corporation and no support or warranty is expressed or implied.

## **NcFTP: An Alternate Unix Client**

NcFTP is a variation on the standard FTP client that many Internet users have come to know and love (or loathe, as the case may be). NcFTP is produced by a company named NCEMRSOft, which apparently consists primarily of Mike Gleason (mgleason@cse.unl.edu) and Phil Dietz (pdeitz@cse.unl.edu). The program is freely distributed and accessible through anonymous FTP at cse.unl.edu in the /pub/mgleason directory, but an Archie search may also turn up a distribution site closer to you. Once retrieved, the program can be compiled on most UNIX systems. Silicon Graphics IRIX, AIX, SINIX, Ultrix, NeXT, and Pyramid OSx are listed in the README file as supported with no modifications. Additional instructions are included for SunOS/Solaris, HP-UX, SCO Unix, SCO Xenix, Bull DPX/2, and Sequent's DYNIX.

A list of features will make the frequent FTP user salivate:

- Supports a customizable ncftprc file that enables you to configure NcFTP settings, define entries for your favorite FTP sites, and associate macros with those sites.
- Maintains a "recent sites" file that enables you to recall and reconnect to recently accessed FTP servers.
- Supports an FTP "redial," enabling retries of an open command to particularly busy sites on a configurable time interval.
- Supports redirection of file transfer or directory output.
- Allows a one-line command shortcut to retrieve a file from a designated FTP server.
- Allows redirection of a file, retrieved through a one-line command, to the console or to a file browser.

- Supports four kinds of file transfer progress indicators.
- Supports a PDIR command to view a directory listing with a previously configured file browser.
- Supports a PAGE command to view a remote file with a previously configured file browser.
- Enables you to retrieve an IP number for a fully qualified domain name.
- Enables you to customize your prompt.
- Supports an FTP session log file.

After some definitions are made in your NcFTP RC file, the amount of work it takes to accomplish an FTP transfer can be greatly reduced. The following example shows one possible file of this type that sets some customization variables, defines one FTP host, and includes a short macro to be executed after a connection is made.

The following is a typical NcFTP RC file. This file can be named ncftpr, netrc, .ncftprc, or .netrc. (This last name is also compatible with the standard FTP program.)

```
#set logfile ~/.ftplog

#set pager "less"

#set progressreports 2

#unset startupmsg

machine ftp.unt.edu

    login anonymous

    password baczewski@unt.edu

    macdef init

        cd /pub/libraries

        dir
```

(A macro must end with a blank line.)

The utility of these statements is seen in the following listing. After the NcFTP command, only the string `unt` was specified and NcFTP matched the first host it found in the RC file containing that string. Notice that there was no prompting for username and password. Upon connection, a change was made to the `/pub/library` directory and a `DIR` command was performed. A `PAGE` command was used to view a file, and then the file was transferred to the local machine.

This only scratches the surface of the features of NcFTP, but it is enough to show the utility of this program. After you begin using it, you may never go back to the "old" FTP. For more information on NcFTP, consult the man pages and other documentation files distributed with the software.

The following shows a typical NcFTP session.

```
~ % ncftp unt
```

Guest login ok, send ident as password.

Guest login ok, access restrictions apply.

Logged into ftp.unt.edu.

total 351

```
drwxrwxrwx  2 0          1          512 Sep 28  1993 .cap
lrwxrwxrwx  1 0          1          13 Sep 28  1993 libraries.adr > libraries.txt
rrr  1 0          1          1858 Jan  1  1994 libraries.africa
rrr  1 0          1          183584 Jan  1  1994 libraries.americas
rrr  1 0          1          20360 Jan  1  1994 libraries.asia
lrwxrwxrwx  1 0          1          13 Sep 28  1993 libraries.con > libraries.txt
lrwxrwxrwx  1 0          1          13 Sep 28  1993 libraries.contacts >
libraries.txt
rrr  1 0          1          89109 Jan  1  1994 libraries.europe
rrr  1 0          1          39063 Aug 12  1993 libraries.instructions
rrr  1 0          1          2884 Aug 11  1993 libraries.intro
lrwxrwxrwx  1 0          1          13 Sep 28  1993 libraries.ps > libraries.txt
lrwxrwxrwx  1 0          1          13 Sep 28  1993 libraries.tx2 > libraries
.txt
rrr  1 0          1          795 Sep 27  1993 libraries.txt
```

Tip: You can use World Wide Web style paths instead of colonmode paths.

For example, if the colonmode path was 'cse.unl.edu:pub/mgleason/ncftp',  
the WWWstyle path would be 'ftp://cse.unl.edu/pub/mgleason/ncftp'.

```
ftp.unt.edu:/pub/library
```

```
ncftp>page libraries.txt
```

The document you tried to retrieve has gone under many changes. Please  
read the following:

The Gopher links are located at the following equivalent sites:

Name=Internetaccessible library catalogs

Type=1

Host=yaleinfo.yale.edu

Port=7000

Path=1/Libraries

Or:

Host=gopher.utdallas.edu

Port=70

Path=1/Libraries

Or:

Host=gopher.sunet.se

Port=70

Path=1/libraries/yaleinfo

If you do not have a gopher client, one appropriate for your workstation can be retrieved via anonymous ftp from the University of Minnesota, at boombox.micro.umn.edu:/pub/gopher.

The paper document itself, in ASCII format, broken up by continents, including this introduction and a list of instructions for different catalogs, can be retrieved by anonymous ftp from ftp.utdallas.edu:/pub/staff/billy/libguide.

ftp.unt.edu:/pub/library

ncftp>get libraries.txt

Receiving file: libraries.txt

100% 0 ||| 795 bytes.

ETA: 0:00

libraries.txt: 795 bytes received in 0.10 seconds, 7.76 K/s.

ftp.unt.edu:/pub/library

```
ncftp>quit
```

```
~ %
```

## Wuarchive FTPD

UNIX system managers may be interested in an alternate FTP server software package that is available. Washington University in St. Louis, the folks that brought you Wuarchive, have developed and released Wu-FTPD (probably as a result of bringing you Wuarchive). Wu-FTPD replaces your standard FTP server and adds the following features:

- Logging file transfers
- Logging FTP commands
- Compression and archiving on the fly
- Classifying users by type and location
- Limits based upon user class
- Upload permissions at the directory level
- Restricted guest accounts
- System-wide and directory-level messages
- Support of a directory alias
- A cdpath command to restrict available directories
- A filename filter

This software provides a greater degree of flexibility in providing FTP services. This is one way, for example, to limit the number of anonymous FTP users who can simultaneously access the server. You can also provide differing degrees of access based upon who users are or where they are located. For example, local users may be able to view the entire directory structure, whereas anonymous connections might be restricted to a particular directory. The access configuration file shown below will give you some idea of the level of customization that is possible with this package.

Wu-FTPD is available on [wuarchive.wustl.edu](http://wuarchive.wustl.edu) in the /packages/wuarchive-ftp directory. If you want to know more about Wu-FTP or discuss its various features, you can subscribe to two mailing lists. The lists wu-ftp and wu-ftp-announce are both maintained on listserv@ [wunet.wustl.edu](mailto:wunet.wustl.edu).

The following is a sample ftpaccess configuration file that is distributed with the Wu-FTPD package.

```
loginfails 2

class    local    real,guest,anonymous *.domain 0.0.0.0

class    remote  real,guest,anonymous *

limit    local    20    Any                /etc/messages/msg.toomany

limit    remote  100   SaSu|Any18000600   /etc/messages/msg.toomany
```

```

limit    remote  60  Any                /etc/msgs/msg.toomany
readme  README*   login
readme  README*   cwd=*
message /welcome.msg                login
message .message                cwd=*
compress      yes                local remote
tar           yes                local remote

# allow use of private file for SITE GROUP and SITE GPASS?
private      yes

# passwdcheck <none|trivial|rfc822> [<enforce|warn>]
passwdcheck  rfc822  warn

log commands real

log transfers anonymous,real inbound,outbound

shutdown /etc/shutmsg

# all the following default to "yes" for everybody

delete      no      guest,anonymous      # delete permission?
overwrite   no      guest,anonymous      # overwrite permission?
rename      no      guest,anonymous      # rename permission?
chmod       no      anonymous            # chmod permission?
umask       no      anonymous            # umask permission?

# specify the upload directory information

upload /var/ftp *                no
upload /var/ftp /incoming      yes      root      daemon  0600  dirs
upload /var/ftp /bin           no
upload /var/ftp /etc           no

# directory aliases... [note, the ":" is not required]
alias  inc:    /incoming

```

```
# cdpath

cdpath  /incoming

cdpath  /pub

cdpath  /

# pathfilter...

pathfilter  anonymous  /etc/pathmsg  ^[AZaz09_\.]*$  ^\.  ^

pathfilter  guest      /etc/pathmsg  ^[AZaz09_\.]*$  ^\.  ^

# specify which group of users will be treated as "guests".

guestgroup  ftponly

email  user@hostname
```

## The Future of FTP

Without doubt, the standard FTP client is going to be useful for some time to come. There is a paradigm shift occurring on the Internet, however, as to how files are accessed for FTP. Services like Gopher and WWW, which do not require a static connection, are gaining favor among system managers, who want to ease the load on their FTP servers. They are also gaining favor among Internet users, because the user interface for those programs tends to be fairly intuitive. Programs like NcFTP provide a greater degree of automation and configuration to the FTP user. The management of FTP servers has been given a greater degree of configurability and control by programs like Wu-FTP. FTP can only get easier as time goes on.





- [8](#)
  - [Telnet](#)
    - [--by Peter Scott](#)
      - [UNIX telnet](#)
        - [Using the Esc Character](#)
        - [Machines in the Same Domain](#)
        - [Using the set Command](#)
        - [Using the ! Command](#)
        - [Nonstandard Port Numbers](#)
        - [Scott Yanoff's Internet Services List](#)
        - [The Difference Between telnet and tn3270](#)
      - [NCSA Telnet](#)
        - [Editing the config.tel file](#)
        - [Starting a Single Remote Session](#)
        - [Quitting a Remote Session](#)
        - [Opening Multiple Sessions](#)
        - [What the Status Line Indicates](#)
        - [A Note on Hosts](#)
        - [Working with the Scrollback Feature](#)
        - [The config.tel File and Alt-P](#)
      - [WinQVT/Net](#)
        - [Downloading to Your PC](#)

## 8

# Telnet

## [--by Peter Scott](#)

This chapter attempts to explain some tips and tricks for the telnet command. It does not attempt to cover all telnet programs on all platforms; that would be too onerous a task. Instead, it concentrates on NCSA Telnet, which runs on MS-DOS machines with network connections; QVTNET, which runs under Windows in a networked environment; and UNIX Telnet, with which most Internet users will be familiar. In all of these versions, telnet enables a user to connect to another Internet machine, whether it be in the next room or halfway around the globe. The phrase *Internet machine* refers to a system that is running the TCP/IP protocols. These protocols are what make Internet connectivity possible. Telnet is often referred to as *remote login*, which simply means connecting one machine to another so that a user may interact with that other machine as though he or she were actually using it locally.

To explain this concept simply, take a look at a typical telnet session using the UNIX version.

## UNIX telnet

A UNIX user performs most functions from a local prompt, which may consist of the % character or a machine name (for example, herald%), or a path (for example, scottp@mary:/home/homec/scottp%). If a user merely types telnet at the prompt, the following prompt appears:

```
telnet>
```

 **Note:** By keying help or ?, the following list of commands is presented:

<i>Command:</i>	<i>Description</i>
close	closes current connection
display	displays operating parameters
mode	tries to enter line or character mode (mode ? for more)
open	connects to a site
quit	exits telnet
send	transmits special characters (send ? for more)
set	sets operating parameters (set ? for more)
unset	unsets operating parameters (unset ? for more)
status	prints status information
toggle	toggles operating parameters (toggle ? for more)
slc	changes state of special charaters (slc ? for more)
z	suspends telnet
!	invokes a subshell
environ	changes environment variables (environ ? for more)
?	prints help information

In most cases it is not necessary to use or even understand all of the commands. Let us, instead, concentrate on those that are useful and often deployed. By issuing the command:

```
telnet> open willow.usask.ca
```

and pressing Enter, the user connects to a machine named willow, which is situated at the University of Saskatchewan in Canada:

```
telnet> open willow.usask.ca
```

```
Trying 128.233.74.5...
```

```
Connected to willow.usask.ca.
```

```
Escape character is '^]'.

```

```
UNIX(r) System V Release 4.0 (willow)
```

```
login:
```

The local system tried to connect to the remote system willow.usask.ca, which also has the Internet number 128.233.74.5. In

this case, the user is being told that the keyboard characters Ctrl+] will break the connection, that willow is operating with UNIX(r) System V Release 4.0, and that it is ready to accept a user login name. If the user has an account on the remote system, then the username is entered:

```
login: scott
```

This brings up the password prompt, enabling the user to complete the login:

```
password: xxxxxxxx
```

and receive the remote prompt:

```
willow>
```

When the user has finished using the remote system, he or she may want to return to the local system. A simple logout or exit command usually suffices.

## Using the Esc Character

Sometimes, it is not obvious which command is valid for logging out of a remote machine. A user may key any number of terms to break the connection and still not succeed. The best way to break a connection is with the escape character—in this example, by holding down the Ctrl key and the right bracket at the same time. This brings up the familiar telnet> prompt, at which point the word quit or close is entered.

## Machines in the Same Domain

If a user is connecting from a machine at, say, usask.ca to another machine at the same site, it is necessary only to key the remote machine's name at the telnet prompt:

```
telnet> open machine-name
```

## Using the set Command

Often, a user may be using telnet to connect to another machine (host 1) that, in turn, connects to a further machine (host 2). If the user keys the traditional escape key to leave host 2, then the connection to host 1 will also be lost. This may be quite inconvenient. Instead, the set command can be used to change the escape key on the local machine. At the telnet> prompt, the following may be issued:

```
telnet> set escape new-character
```

It is always wise to define the new key in such a way that it will not conflict with any keystroke available on the host or remote system.

## Using the ! Command

At times it is necessary to temporarily leave a telnet session in order to perform a task on the local machine. Here is an example. Suppose a user has used telnet to connect to an online catalog:

```
herald% telnet
```

```
telnet> open sundog
```

This puts the user into the remote site. In order to leave, the escape key is entered, bringing up the telnet> prompt. The ! character is entered, and the user receives the local prompt again:

```
telnet> !
```

```
herald%
```

at which point local machine commands may be entered. When exit is keyed, the local prompt is returned. Hitting Enter again places the user back into the remote site.

## Nonstandard Port Numbers

When connecting to a remote computer with telnet, a user is automatically connected to port 23. This is the port reserved by UNIX systems for incoming telnet sessions. However, it is possible to telnet to another port to run certain applications. For example, if a user wanted to run the finger program on a remote system, the following command would be issued:

```
telnet> open herald 79
```

Once connected, the user merely hits the Return key to see who is currently logged in to that remote machine:

Login	Name	TTY	Idle	When	Office
ford	Ron J. Ford	02	37	Tue 09:37	
jjk136	Justina Jill Koshins	03		Tue 06:41	
burke	John Burke	04	2	Tue 12:18	
bergg	Gary Berg	05	46	Tue 11:47	
anderson	Cathy Anderson	06	41	Tue 10:02	
stefaniu	Debbie Stefaniuk	07	3:41	Tue 08:53	

A local user can also receive specific information that a remote user has made public through his or her .plan file. For example, John A. Magliacane, of Brookdale Community College in New Jersey, "publishes" the *SpaceNews* online newsletter. To read it, the user types the following:

```
telnet> open pilot.njin.net 79
```

and when connected, types the username magliaco to receive the following:

```
Login name: magliaco
```

```
In real life: John Magliacane
```

```
Directory: /njin/u4/magliaco          Shell: /bin/ocsh
```

```
Last login Tue Jun  7 09:33 on ttyu7 from basie.rutgers.ed
```

```
New mail received Tue Jun  7 13:12:01 1994;
```

```
unread since Tue Jun  7 10:37:11 1994
```

```
Plan:
```

```
SB NEWS @ AMSAT $SPC0606
```

```
* SpaceNews 06-Jun-94 *
```

```
=====
```

```
SpaceNews
```

```
=====
```

```
MONDAY JUNE 6, 1994
```

SpaceNews originates at KD2BD in Wall Township, New Jersey, USA. It is

published every week and is made available for unlimited distribution.

<text edited out>

Of course, regular UNIX users know that the finger program can accomplish the same task; I am merely pointing out the power that telnet has. It should be noted, however, that many non-UNIX users find that using telnet to port 79 is the only way to access such information, because remote fingering is not always possible.

Here are some other useful information sources available via telnet to port 79. Note that the telnet address is being keyed in without the user being in open command mode:

```
telnet oddjob.uchicago.edu 79
```

Enter: copi

This is an Almanac of Events maintained by Craig J Copi.

The following give current earthquake information for various parts of the United States. Enter quake after being connected:

- telnet geophys.washington.edu 79 (Washington, Oregon)
- telnet fm.gi.alaska.edu 79 (Alaska)
- telnet scec2.gps.caltech.edu 79 (Southern California)
- telnet slueas.slu.edu 79 (Southern Illinois)
- telnet seismo.unr.edu 79 (Nevada and California)
- telnet andreas.wr.usgs.gov 79 (Northern California)
- telnet eqinfo.seis.utah.edu 79 (Utah, Wyoming, Montana)

### Scott Yanoff's Internet Services List

Get a copy of the Internet Services List, compiled by Scott Yanoff. This list gives instructions for connecting to many Internet resources via telnet, especially those with nonstandard port numbers. It is available via telnet to `csd4.csd.uwm.edu 79` (enter yanoff when connected). The list includes connections to Chat Services, Online Games, Sports Information, and certain Bulletin Board Services.

### The Difference Between telnet and tn3270

Most remote computers can be accessed with telnet, but there are still many around that need to be connected via tn3270. tn3270 allows a UNIX machine to connect to an IBM-mainframe machine, using full-screen and full-duplex emulation. Suppose, for example, a user needed to connect to the Pickler Memorial Library, Northeast Missouri State University, which has the Internet address `academic.nemostate.edu`. Telnetting to that address would connect the user but would not allow a public login. The user is instead presented with

```
_8VM/ESA ONLINE_8
```

which is not particularly enlightening. With a tn3270 connection the following is shown:

```
VM/ESA ONLINE
```

```

NN      NN      MM                MM      SSSSSSSS      UU      UU
NNN     NN     MMM             MMM     SS           SS      UU      UU
NNNN    NN     MMMM          MMMM    SS                UU      UU

```

```

      NN NN NN      MM MM MM MM      SSSSSSS      UU      UU
      NN  NNNN      MM   MMM   MM      SS      UU      UU
      NN   NNN      MM     M     MM      SS      SS      UU      UU
      NN     NN      MM              MM      SSSSSSS      UUUUUU

```

VM/ESA Rel. 2.0

If using TELNET to connect to our system, TAB to the COMMAND line and type DIAL VTAM. Otherwise, type VMEXIT on the COMMAND line to return to the

A - NOTIS, B - MUSIC screen.

Fill in your USERID and PASSWORD and press ENTER

(Your password will not appear when you type it)

USERID ===>

PASSWORD ===>

COMMAND ===>

.....  
This is a typical, IBM-mainframe opening screen. The instructions should probably have used the term tn3270 instead of TELNET. By entering the phrase DIAL VTAM, the user is presented with a menu choice:

WELCOME TO COMPUTING AT N.M.S.U.

Type the letter of the system you desire

and Press ENTER

A - Enter the NOTIS Library System

B - Enter the MUSIC Academic System

.....  
Selecting A brings up the online catalog menu:

LUIS: LIBRARY USER INFORMATION SERVICE

Pickler Memorial Library, Northeast Missouri State University

To search for: Type a command and press <ENTER>

Author a=dickens charles

Title t=gone with the wind

```

Subject          s=education elementary
Children's subject      sc=family problems
Keyword          k=clancy and hunt and october
Call number      cl=bx7432

```

Type EXP OPT and press <ENTER> for a complete list of options.

REMOTE USERS: Type STOP and press <ENTER> to log off.

For more information on searching in the Catalog, press <ENTER>.

----- + Page 1 of 2 -----

Enter search command <F8> FORward page

NEWS

NEXT COMMAND:

.....

The user performs the search, enters STOP, and is presented with the menu choice again. Even though it is not stated, the command for logging off is UNDIAL. If the logoff command is not readily apparent, the user can hit Ctrl+C to receive the tn3270 prompt, at which time the word quit may be entered to break the connection. tn3270 connections are becoming less common, but there are still enough around to warrant this explanation.

## NCSA Telnet

Users who wish to make telnet connections from their IBM personal computer may do so with NCSA Telnet. This software was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Unlike the UNIX version, this client resides on a user's own personal computer, thus allowing direct access to other computers on the Internet running the TCP/IP protocols.

In order to run the software, the personal computer must have the following:

MS-DOS version 2.0 or later

An Ethernet adaptor board and cable

384KB memory (minimum)

An Internet address (usually supplied by a system administrator)

The software is available via anonymous FTP from

ftp.ncsa.uiuc.edu

After you log in to the site, change the directory to PC/Telnet/msdos. The files to receive are either tel2307b.zip or tel2307s.zip. The file tel2307b.zip is a PKZip 2.04 zipfile. It contains all the binaries and configuration files needed to run PC telnet and is 717,475 bytes in size.

The file tel2307s.zip is also a PKZip 2.04 zipfile. It contains all the source code used to compile all the binaries, and the makefiles. It is 679,778 bytes large. There are no binaries in the source distribution.

### Editing the *config.tel* file

It is essential that the config.tel file be edited, so that the machine's IP number can be listed. If editing this file is problematic, then a system administrator should be consulted. The administrator will also know if the machine has the correct Ethernet adaptor attached. Because the software resides on a personal computer, it can take advantage of the capabilities of the machine. For example, colors can be chosen, the clock set, and capture files can have names chosen.

Let's now look at how the program is started, how the terminal type is set, and how a telnet session is closed.

## Starting a Single Remote Session

The simplest way to start NCSA Telnet is from the DOS prompt:

```
c:\ telnet hostname
```

This command will, if the remote system is available, make an instantaneous connection. The user then logs in with a username and password if required.

Pressing Alt-H will bring up the help screen, overlaying the current session:

.....

Alt-A	add a session	Alt-Y	Interrupt Process
Alt-N	next session	Alt-B	Previous Session
Alt-D	Dump Screen to Capture file	Alt-O	Abort Output
Alt-Z	message screen	Alt-Q	Are you there?
Alt-E	escape to DOS shell	Alt-U	Erase line
Alt-G	graphics menu	Alt-K	Erase Kharacter
Alt-C	toggle capture on/off	Alt-V	Paste Capture to Session
Alt-R	reset VT100 screen	HOME	exit graphics mode
Alt-H	this help screen	Ctrl-HOME	clear/enter graphics mode

ScrLock pause/restart screen (DO NOT use Ctrl-NumLock)

ScrLock enter/exit scroll-back mode

Alt-F start file transfer as if typed:ftp [internet address]

Alt-I send my internet address to host as if typed

Alt-S skip scrolling, jump ahead

Alt-P change a parameter, one of: color, capture file name, backspace, session name, screen mode

Alt-X close connection

CTRL-Shift-F3 abort program completely. STRONGLY discouraged

Press ESC for information page, space bar to return to session:

We will look at some of these commands later.

## Quitting a Remote Session

With NCSA Telnet, the simplest way to quit a single session is to log out of the remote machine, and be returned to the DOS prompt. However, it could be that there is no clear way of logging out. Just like the UNIX version of telnet, this program also has an escape key: Alt-X.

## Opening Multiple Sessions

So much for a single session. One of great strengths of NCSA Telnet is the ability to have more than one session live at any given time. There are two ways of performing this. First, a user can enter multiple machine names at the DOS prompt:

```
C:\ telnet machine1 machine2 machine3
```

and be prompted for login and password information on each machine. The second way is to use Alt-A to add another session.

## What the Status Line Indicates

The program uses VT100 terminal emulation. This means that only 24 of the screen's 25 lines are used by the terminal. The final line is used by the program to indicate the status of the sessions. Each session name is indicated in reverse video, with the current session shown on the far left. The far right of the screen is reserved to show whether scrollback, capture, or FTP is enabled.

## A Note on Hosts

The program can connect only to computers that have IP addresses. A user can find legal names by checking the config.tel file. This file can be configured to use the local domain-based name server to check names and addresses of machines. Like the UNIX version of telnet, this program enables a user to enter only the specific machine name (for example, willow, herald, or duke), providing they are all in the same domain—in this case, usask.ca. If the IP numbers are known, then it is necessary only to enter the last part of the number. So, if the local machine's number is 128.233.33.3 and the user wishes to connect to 128.233.33.7, then all that needs be entered is #7. Whereas UNIX allows a space between an IP address and a port number, this program requires that a pound sign (#) be inserted (for example, 128.233.33.3#5000).



**Tip:** Watch for the asterisk to disappear to show that you are connected.

It can sometimes take a couple of minutes to connect to a remote machine. While the connection is being made, an asterisk appears in the status box. This disappears when the connection is made. However, it may take further time for a login command to appear. At this point, it may make sense to break the connection with Alt-X and begin the procedure again.

## Working with the Scrollback Feature

One of the most frustrating things about logging in to remote machines is that the information a user has received appears to drop off the top of the screen and be lost. NCSA Telnet has a very useful set of commands to enable a user to refer back to that information. This is all achieved by the Scroll Lock key. When depressed, that key enables a user to refer back to received information with Page Up, Page Down, and the up and down arrow keys. The information can not only be read but manipulated with cut-and-paste features. Here's how it works:

1. Hit the Scroll Lock key.
2. Place the cursor at the start of text to be copied.
3. Press the Spacebar.

4. Move the cursor to the end of the text.
5. Press the Spacebar.
6. Press Alt-C to place text into the buffer.
7. Hit the Scroll Lock key.
8. Switch sessions with either Alt-B or Alt-N.
9. Move cursor to pasting position and press Alt-V.

This procedure can also be accomplished with a mouse, but takes some dexterity:

1. Press the right mouse button.
2. Move to the start of the text to be copied.
3. Press the left mouse button.
4. Move to the end of the text.
5. Press left mouse button.
6. Press and hold the left mouse button.
7. Press the right mouse button.
8. Release both buttons. This copies the text into the buffer.
9. To exit Scrollback, press the right mouse button.

To paste the captured text, the user hits Alt-B or Alt-N to switch sessions. Then:

1. Press and hold the right mouse button.
2. Press left mouse button.
3. Release both buttons.

NCSA Telnet also enables a user to capture text to a file on the personal computer or to a printer. An entire session or just a single screen can be captured. The Alt-C key combination turns capture on and off and appends the text to the file designated in a user's parameters. To see the current name of the capture file and to change it if necessary, Alt-P is depressed. This change can be made permanent by editing the config.tel file, which comes with the software.

Sometimes, a user may wish to capture only one screenful of information, particularly if that information is garbled by an Alt-C capture. To do this, use Alt-D. In any event, only one capture can occur at any given time. A user cannot capture text with Alt-C in one session and Alt-D in another. If the captured text is to be sent to a printer instead of a file, the user should enter PRN as the capture file name.

In order to edit the captured file—and indeed, to perform other DOS commands—NCSA Telnet includes a DOS Shell

feature. This is controlled by Alt-E. Typing exit returns the user to the program. While in the DOS shell, a user may run other programs. However, any other program that wants to use the Ethernet connection will cause NCSA Telnet to crash. Other programs also may cause crashes. Learning which programs to avoid using is a matter of trial and error. Also, trying to run programs that consume lots of random-access memory may cause problems.

## The *config.tel* File and Alt-P

The *config.tel* file determines how NCSA Telnet will run. It is a text file that can be easily edited with a DOS text editor. Some instructions may not be clear to novices, however, in which case a systems person should be consulted for advice. Some parameters—for example, screen colors, capture-file name, terminal type, and so on—can be changed while in the program. Depressing Alt-P brings up the following:

```
<      Select parameters, F1 to accept, F10 to leave unchanged      >

----- Color setup and session parameters -----

Text:      normal      reverse      underline

Normal Foreground (nfgcolor)      -      green
Normal Background (nbcolor)      -      black
Reverse Foreground (rfcolor)      -      black
Reverse Background (rbcolor)      -      white
Underline Foreground (ufcolor)      -      blue
Underline Background (ubcolor)      -      black
Use remote echo or local echo      -      Remote echo
Backspace key sends      -      Delete

Session name      *>      ahostname
Terminal type      -      VT102 and Tek4014
Line Wrapping      -      Wrapping On
Output Mapping      -      Mapping Off

----- Parameters which apply to all sessions -----

Capture file name      *>      capfile

Screen mode (for BIOS compatibility)      Direct to screen

File transfer is      -      Enabled
Remote Copying is      -      Enabled

Clock is      Enabled

Use arrow keys to select, Enter clears changeable field (*>)
```

To move through the list, the Up, Down, Home, and End keys are used. The left and right arrow keys allow changes to be made to fields by means of toggles. The fields marked \*> can have new values added by hand.

NCSA Telnet comes with some other useful utilities that mirror their UNIX counterparts:

- *Finger*: Enables a local user to gain information about users on remote machines.
- *FTP*: Enables a local user to upload and download files from the local to the remote machine.
- *Lpq*: Displays printer queue jobs.
- *Lpr*: Sends a job to the printer.
- *Lprm*: Removes a job from the printer queue.
- *Rsh*: Similar to telnet but enables a user to add a personal name to a remote login.

Finally, here is a summary of frequently-used program commands:

Alt-A	Adds a new session.
Alt-B	Goes back to a another session.
Alt-C	Toggles capture file on and off.
Alt-D	Dumps the current screen into the capture file.
Alt-E	Spawns a DOS shell.
Alt-F	Starts file transfer as if typed: <i>FTP Internet address</i> .
Alt-G	Displays the Graphics menu.
Alt-H	Displays the Help screen.
Alt-I	Sends the internet address to the host.
Alt-K	Erases a character.
Alt-N	Goes to the next session.
Alt-O	Aborts output.
Alt-P	Changes a parameter (for example, color, capture file name, backspace).
Alt-Q	Sends the query, Are you there?
Alt-R	Resets the VT100 screen.
Alt-S	Skips scroll and jumps ahead.
Alt-U	Erases the line.
Alt-V	Copies text from clipboard onto the current session.
Alt-W	Sends your password when you're FTPing back to your PC.
Alt-X	Closes the connection.
Alt-Y	Interrupts the process.
Alt-Z	Displays the Message screen.
Ctrl-Home	Clears/enters graphics mode.
Home	Exits graphics mode.
Scr Lock	Enters/exits scrollback mode.

For more information on The National Center for Supercomputing Application, a user can connect to the NCSA Home Page (shown in Figure 8.1) on the World Wide Web:

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

### **Figure 8.1. The NCSA home page.**

A version that runs under Windows 3.1 is called wintelb3.zip, available via anonymous FTP from ftp.ncsa.uiuc.edu in the /PC/Telnet/windows directory. It is currently in unsupported beta mode and will not be discussed here. Rather, I will look at another Windows telnet program, WinQVT/Net.

## **WinQVT/Net**

Currently, the best Windows-based telnet program is WinQVT/Net, produced by QPC Software, P.O.Box 226, Penfield, NY, 14526 (e-mail djpk@troi.cc.rochester.edu for further details). The registered version costs \$40, and the shareware version is available via anonymous FTP from biochemistry.cwru.edu in the directory /gopher/pub/qvtnet as:

```
qvtws397.zip      Windows Sockets version
qvtne397.zip      packet-driver version
qvtnt397.zip      32-bit version for Windows NT
```

I presume that the user knows how to obtain the software and create an icon for it in Windows. To start the program, the user clicks the icon and is presented with a screen similar to Figure 8.2.

### **Figure 8.2. The WinQVT/Net startup screen.**

Notice here that only the Terminal and FTP options have been configured.

To start a terminal session, which can be either a telnet or rlogin connection, the user merely clicks the Terminal button to receive the Start Terminal Session window, as shown in Figure 8.3.

### **Figure 8.3. The Start Terminal Session window.**

A description of the elements of this screen will be useful. The Host Name or IP Address: box can be used to enter the site to which a user may wish to connect. The telnet or rlogin button is selected, and the OK button is clicked.

---

 **Note:** A username *must* be entered if making an rlogin connection.

---

Both telnet and rlogin connections function similarly to their UNIX counterparts; that is, telnet connects to port 23 of the remote machine, while rlogin connects to port 513. But while UNIX rlogin requires a user to key `-L username`, WinQVT/Net accepts just the user's name. Also, if the connection requires another remote port number, then it can be added in the IP Port box.

Notice the large box on the right of Figure 8.3. This is the hosts list, which was created in the qvtnet.rc file. In this example, all machine names are local, meaning they belong to the usask.ca domain. However, any valid Internet address can be added to the file. By selecting the entry for the remote machine herald and clicking OK, a connection is made, bringing up the login: prompt, as shown in Figure 8.4.

### **Figure 8.4. A connection is made.**

Once the user is logged in, this screen can be turned into an icon in traditional Windows fashion, and another session can be started. In fact, up to fifteen concurrent sessions can be running at any given time.

---



**Note:** Use Alt-C to bring up the Console window. This brings up the Console window, from which other sessions can be started. It is, of course possible, to also start FTP, mail, news, and printer sessions from the Console window, and have them all running simultaneously.

---



**Note:** Once a number of sessions are running, a user can key the Alt-N combination to switch between sessions. This can be faster than doing the same thing with a mouse. Other useful key combinations are Alt-Tab and Alt-Esc, which place a window in the background.

---

WinQVT/Net's terminal sessions allow screen capture, logging a session to a file, the execution of scripts, cut and paste from one session to another, color and font selection, and the ability to redefine keys. Being a Windows-compatible program, WinQVT/Net can also be used by such Worldwide Web browsers as Cello and Mosaic when spawning telnet sessions. It is a good idea to have it included in the Windows StartUp window, so that it loads automatically at login time.

There are other Windows-based telnet utilities worth looking at. These, like WinQVT/Net, are constantly being revised, so some of the filenames may change. They are all available via anonymous FTP from [winftp.cica.indiana.edu/pub/pc/win3/winsock](ftp://winftp.cica.indiana.edu/pub/pc/win3/winsock):

<i>Program Name</i>	<i>Date</i>	<i>Description</i>
ewan1c.zip	940503	EWAN a free Winsock 1.1 telnet
qws3270.zip	940325	qws3270 Winsock tn3270 emulator (3/25/94 release)
trmptel.zip	940311	Trumpet TELNET (VT100) Terminal for Winsock
vt220.exe	940519	Dart Communications VT220 for workgroups Winsock
wintelb3.zip	931202	NCSA Telnet for MS Windows (unsupported beta 3)

---



**Tip:** Get a copy of Hytelnet for the most complete index of telnet-accessible resources.

---

The very best source of telnet-accessible sites can be found in the Hytelnet utility. Here is the announcement for Version 6.7, which was released May 16, 1994:

Hytelnet version 6.7, the utility that gives an IBM-PC user instant-access to all Internet-accessible library catalogs, FREE-NETS, CWISs, BBSs, Gophers, WAIS, etc. is now available. You can get it via anonymous FTP from: [ftp.usask.ca](ftp://ftp.usask.ca/pub/hytelnet/pc/latest) in the [pub/hytelnet/pc/latest](ftp://ftp.usask.ca/pub/hytelnet/pc/latest) subdirectory. It is listed as [hyteln67.zip](ftp://ftp.usask.ca/pub/hytelnet/pc/latest/hyteln67.zip):

URL= <ftp://ftp.usask.ca/pub/hytelnet/pc/latest/hyteln67.zip>

Version 6.7 is a major upgrade. Much redundant information has been deleted, and errors have been corrected. All the new/updated files created since Version 6.6 have also been incorporated. The unzipped files total over 1.2 MB. . .but remember, you can always edit out any information you do not need, in order to save space.

Information from Roy Tennant follows, slightly edited, describing how to obtain Hytelnet 6.7 from the FTP site:

1. To retrieve Hytelnet, at your system prompt enter `ftp ftp.usask.ca` or `ftp 128.233.3.11`.
2. When you receive the Name: prompt, enter anonymous.
3. When you receive the Password: prompt, enter your Internet address.
4. When you are at the `ftp>` prompt, enter binary.

5. At the next ftp> prompt, enter: cd pub/hytelnet/pc/latest.
6. Enter get hyteln67.zip.
7. After the transfer has occurred, enter quit.

Proceed with the following instructions to retrieve the UNZIP utilities (which you need unless you already have them).

The Hytelnet program is archived using a PKZIP.EXE. To unarchive it, you must be able to unzip the file. If you have the file PKUNZIP.EXE, it will unarchive the hyteln67.zip file. If you do not have it, you may retrieve it by following these instructions:

1. Use the previous instructions for connecting to oak.oakland.edu.
2. At the ftp> prompt, enter binary.
3. Enter cd pub/msdos/zip.
4. Enter get pkz204g.exe.
5. After the transfer has occurred, enter quit.

You can also unarchive with UNZIP.EXE. To retrieve it, follow these instructions:

1. At your system prompt, enter ftp oak.oakland.edu.
2. When you receive the Name: prompt, enter anonymous.
3. When you receive the password prompt, enter your Internet address.
4. When you are at the ftp> prompt, enter binary.
5. At the next ftp> prompt, enter cd pub/msdos/zip.
6. Then enter get unz50p1.exe.

## Downloading to Your PC

Because of the plethora of PC communications programs, I will not attempt to give step-by-step instructions on how to download to your PC. You should check the instructions for your software for downloading a binary file from your Internet account to your PC.

To Unarchive hyteln67.zip, make a new directory on your hard disk (for example, mkdir hytelnet) Copy PKUNZIP.EXE or UNZIP.EXE and hyteln67.zip into the new directory. Make sure you are in that directory, then enter pkunzip hyteln67.zip. This will unarchive hyteln67.zip, which contains the following files:

HYTELNET.ZIP

READNOW

The file READNOW gives full instructions for unarchiving HYTELNET.ZIP. Simply put, you *MUST* unZIP the file with the -d parameter (if you are using PKUNZIP.EXE) so that all the subdirectories will be recursed. If you are using UNZIP.EXE,

no parameters are required.

To use Hytelnet, you should refer to the instructions in the release announcement or to the README file included with the package.

Please note that I offer these instructions as a service for those who are unfamiliar with the steps required to download and use files from network sources. I cannot be responsible for any local variations in these procedures that may exist. Please contact your local computer support staff if you have difficulty performing these tasks.

The UNIX/VMS version, created by Earl Fogel, is available for browsing by telnet to access.usask.ca; login with hytelnet (all lowercase). For more information on this version, contact Earl at fogel@herald.usask.ca.





- [9](#)
  - [Usenet](#)
    - [--by James "Kibo" Parry](#)
      - [Volume, Volume, Volume](#)
        - [Crossposting and Redirecting](#)
      - [The Art of the Killfile](#)
      - [Where the Heck Is That Group?](#)
      - [Creating a New Group](#)
      - [Kibo](#)
        - [--by Tod Foley](#)
      - [Searching For Things](#)
        - [Searching Subject: and From: Lines](#)
        - [rn Text Searches](#)
        - [grep: Searching at the Lowest Level](#)
        - [Larry Wall's clip](#)
        - [The Shape of Agents to Come](#)

## 9

# Usenet

[--by James "Kibo" Parry](#)

## Volume, Volume, Volume

If you've read the Foreword, you've heard me ranting about how much there is to read on Usenet, the Internet's news/discussion service. It is text in bulk, much of it is just vacuous chatter or repetitious arguments, and worst of all, it's not exactly well-indexed.

When I say "not exactly well-indexed," I mean that all of the information is sorted by topic (into *newsgroups*). Discussion of, for instance, Macintosh hardware goes in comp.sys.mac.hardware, and discussion of field hockey is in rec.sport.hockey.field. The problem is that it's often hard to find which newsgroup, out of over 9000, the topic you want to read is in; and even with this many groups there are topics where there's no clear association with any one group. There are also groups where the topic is broad enough that 200 new articles appear in that group every day. The key to mastering Usenet lies in the ability to find only the interesting or useful stuff, without having to read the rest.

This chapter will go into a bunch of detail about that later, but first I'll touch on some simpler topics that people frequently ask about. You may already know the first couple of these—but then again, you may not.

## Crossposting and Redirecting

As I'm sure you know, when you post a new article to Usenet, you get to choose which newsgroup to put it in. Let's say I've just thought of a question I need to ask about my neon tetras. I would, naturally, put it in rec.aquaria. The post would start

with the following header:

```
Newsgroups: rec.aquaria
```

But suppose that I've just thought of a wonderfully ribald pun about neon tetras. The people in rec.aquaria would probably enjoy it, but people interested in jokes in general might also enjoy it. Therefore, I'd want it to go to both rec.aquaria and rec.humor. But if I posted the article once to each group, the people who read both groups will see both articles, and they'll get mad at me for wasting 10 valuable seconds of their time by repeating myself. Fortunately, there's a way to post one copy of an article to two (or more) groups, by *crossposting*:

```
Newsgroups: rec.aquaria,rec.humor
```

The only thing to remember is that there are no spaces around the comma, but there always has to be one after the colon in any Usenet header.

Of course, so far, this has been a purely hypothetical discussion. If I really were posting a pun about neon tetras, I'd want to crosspost it to a third group (alt.religion.kibology) because the people in said group want to see the stuff I write (hence its name):

```
Newsgroups: rec.aquaria,rec.humor,alt.religion.kibology
```

The article shows up in all three groups, but only one copy of it is normally stored on each computer—cross-posts don't take up extra disk space or "bandwidth" on machines where the machine can save the file in the rec.aquaria directory and make symbolic links to it in the other two. (More later on how machines store Usenet articles.)

A question arises now: If someone makes a followup posting (that is, a public response) to my article, it will by default go to all three newsgroups (unless the responder edits that Newsgroups header line). Because I'm making a silly joke, the silly people in rec.humor and alt.religion.kibology will likely respond with something silly, which may or may not be about tropical fish. If I want to set it up so that any discussion the article generates will go into rec.humor and not disturb the aquarium fans, I just have to type an extra header:

```
Newsgroups: rec.aquaria,rec.humor,alt.religion.kibology
```

```
Followup-To: rec.humor
```

Again, the space after the colon is necessary, and you need to capitalize that *T*. Now, by default all followups will go to rec.humor, except if the responder edits the Newsgroups list manually.

Some people, when they see a string of articles (a *thread*) that they would like to see moved out of a newsgroup (say they're tired of a "Macs vs. PCs" argument in soc.penpals), will try to move the thread by posting an article like this one:

```
Newsgroups: soc.penpals
```

```
Followup-To: alt.flame
```

```
From: spot@doghouse.yard.net (Spot, a dog)
```

```
Date: Mon Aug 15 17:05:34 EDT 1994
```

```
I'm tired to this thread, followups to alt.flame. —Spot
```

...but this won't work. It will only redirect the followups to the new article, not to any of the dozens of older articles.

The only other catch to the Followup-To: header is that when you post a followup, you should read your own Newsgroups line before you post—quite often a thread will be redirected without warning. This is occasionally done as a practical joke of sorts:

```
Newsgroups: rec.arts.startrek.fandom
```

```
Followup-To: alt.test,misc.test,alt.sex,alt.alien.visitors,alt.magic
```

From: lucy@babaloo.net (Lucy Ricardo)

Date: Mon Aug 15 13:41:02 EDT 1994

"Star Trek" sucks! "Space: 1999" was better because Martin Landau could beat up William Shatner any day! Majel Barrett was only on Star Trek because she was married to William Shatner!!!

When careless Trekkies post followups, frothing at the mouth over the insult to the show, and patiently explaining that Majel was married to Gene Roddenberry and not William Shatner, they will be surprised to find out (later) that their post has gone to alt.magic (they'll get mail saying "Why are you posting this drivel to alt.magic?") and to misc.test (they'll get automatic replies from a dozen sites saying "Your article was received here in Norway"). This is just one of the many ways Usenet is used to get "newbies" to show themselves. (I'd tell you the others, but I don't want to ruin the market for elitist pranks; what would Usenet be without an elaborate social stratification?)

Note that all this discussion about how to edit Newsgroups: and Followup-To: lines is moot if you're using one of the few commercial systems that allows access to Usenet but doesn't let you edit any headers (I've heard Portal is one such system). In normal environments, you can move the cursor around and edit most of the headers to your liking, in case you want to change your Organization or Summary line. (Some of the headers are automatically set by the news software, so changing Date or From won't do too much good.)

## The Art of the Killfile

Let's go back to the hypothetical question of "What do you do about the Macs vs. PCs flamewar in soc.penpals?" Some people would simply post to soc.penpals asking the participants in the flamewar to shut up. Bad idea—that's just one more article everyone has to read, and it probably won't do much good. If only there were a key you could press to simply make that flamewar vanish forever!

Welcome to the world of the *killfile*. A killfile is simply a list of things you don't want to see. If I have a killfile for soc.penpals that removes all articles with the words "Macintosh" or "PC" in their Subject headers, I can ignore that flamewar. (Note that it will not stop *anyone else* from seeing the flamewar! A few people misunderstand the concept of killfiles—probably because of the word "kill" in the name—and will rant about censorship if they find out that someone has killfiled them. Killfiles will not blow someone off the face of the earth—although sometimes you'll wish they could.)

The exact operation of a killfile depends on what sort of system you're using and (more importantly) what program you're using to read Usenet. Most of them (such as the rn family) keep a directory with a separate file for each group you want to operate on. Others do not literally keep *killfiles* but instead keep a list of killable things somewhere in their database.

rn and its derivatives (rrn, xrn, xrrn, trn, trrn, and so on) all have the same basic killfile functionality, and they're quite popular (as you can see by the plethora of variants), so I'm going to talk about how killfiles work in them. If you use nn, some of this will also apply to you (but not all); if you use gnus, none of this will do you any good; gnus is unlike anything else. (You can do anything you want in gnus, but it requires writing bits of LISP code.) And for those of you running software without killfiles (for example, some shareware newsreader on your 386), you should consider running something else, as killfiles are quite useful.

In rn (or trn, and so on), if you're looking at an article, and you never want to see another one with the same subject, just press K (that's Shift-k.) You will see something along these lines:

```
Marking subject "I hate your computer\.\.\." as read.
```

```
Depositing command in /home/foyer/kibo/News/soc/penpals/KILL...done
```

```
Searching...
```

What this means is that rn has recorded, in a file for soc.penpals, that the subject I hate your computer... is verboten.



**Note:** Why the backslashes in front of the periods? Because punctuation marks have various special meanings. The period, for instance, stands for any single character. If you really want all the gory details, read the documentation for a program called `grep`.

The file that `rn` created for `soc.penpals` now looks like so:

```
THRU 40697
```

```
/: *I hate your computer\.\.\.:j
```

That first line is just to remind `rn` what the last article it checked was, so that it won't have to scan them all again tomorrow. The next line does the actual killing. When you press `K`, it automatically adds another one of those lines with the current Subject header inserted in the middle. You can do far more than just kill a particular subject, though. If you press `Ctrl-K` instead, `rn` will let you edit the contents of the killfile, and you can type in new killfile lines.

Each line has three parts: a pattern to search for, an optional modifier that tells it what to search, and then an `rn` command.

```
/pattern/modifier:command
```

`J` is the key you press to "junk" an article (to make it go away and mark it as having been read so that it won't show up next time). Therefore, to junk all articles whose Subject lines contain—*anywhere*—the word `marshmallow`, you can type this:

```
/marshmallow/:j
```

This would kill a subject such as "What's the best flavor of marshmallow?" as well as "I like MaRsHmaLLowS" (it'll work on upper- and lowercase letters).

The optional modifier before the colon is what gives you the real power. Normally `rn` searches the subject line for your pattern. What if you want to search *all* the headers?

```
/marshmallow/h:j
```

The `h` makes it junk any articles about marshmallows, or from someone with the word "marshmallow" in their name (it could happen, I suppose), and even articles that pass through a computer named "marshmallow." `h` searches all the headers. Want more? Here's how to kill any article that mentions marshmallows anywhere, in the headers, the text, or the signature:

```
/marshmallow/a:j
```

With a little ingenuity, you can use those modifiers to do various things. To kill any postings made by John Smith (`jsmith@slackvax.subg.com`) you can type:

```
/^From:*jsmith@*subg.com/:j
```

The `^` (caret) stands for the beginning of a line; the `*` (asterisk) stands for any text. We didn't include the word `slackvax` because someday `jsmith` might post from another workstation at the same site. If you wanted to kill all articles from anyone at that darn `subg.com` site, it's as simple as using an asterisk: `eliminate *@*subg.com`.

Here's another idea: you're reading `rec.arts.prose` to see some original fiction, but you don't want to see all the followups that discuss it. Because followups normally have `Re:` added somewhere in the subject line, you can kill the subject `Re:`.

To eliminate crossposted articles from newsgroups you dislike, you can do something along the lines of:

```
/^Newsgroups:*alt\.sex\.fetish\.feet/:j
```

There's also a *global* killfile that is processed whenever you read any group. You can edit this by pressing `Ctrl-K` when you're not reading articles in any group. Of course, it can sometimes slow down your session greatly if you have an elaborate global killfile, but if there's something you're just plain sick and tired of seeing *everywhere*, this is the optimal answer.

If you're using `nn`, the `K` key will perform similar functions—and some others—that involve `nn` asking you some questions about what sort of articles you wish to automatically kill or select (the `K` key does both). It's nice that `nn` lets you create

killfiles with this interactive method, useful if you hate typing strange characters like colons. Read the nn manual (pages 33 to 37) for details.

If you're using trn (or trrn, the networked version) you can do everything you can in rn—plus, you have the f modifier that searches only the From: line.

If you have your newsreader set up to automatically eliminate *all* articles you don't specifically ask to see (nn works this way by default, and trn can be made to do it), you can use the killfiles to *select* the good articles instead of deleting the bad ones. If you find yourself wanting to read only a few articles in most of the newsgroups you visit, here's a handy little trick that makes trn display menus of articles and then show you only the ones you select:

```
trn -a -x -X1X>
```

Because that > is actually part of the command and not the usual > that tells the shell to write the output to a file, you may need to escape it with a \ immediately in front. In csh, you can add this to your .cshrc:

```
alias trn "trn -a -x -X1X\> !*"
```

The -a and -x enable some useful features, and -X1X> defines the default command (that is, the one the Spacebar does) to page you through the menus of articles and then delete all the unmarked ones. You can mark an article by selecting its letter from the menu or by using a killfile; + is the command to mark an article.

Here's a trn killfile entry that, if used with the preceding options, will show you only the articles from guys named Bernie:

```
/Bernie/f:+
```

For more details about any of this stuff, read the manual for the appropriate program, and also see the rn KILL file FAQ that is posted periodically to news.newusers.questions and news.answers.

## Where the Heck Is That Group?

One of the more frustrating situations you'll face reading Usenet is that you may suddenly decide that you want to read about football, but you don't know if it would be filed under rec.football, rec.sport.football, rec.sport.pro-football, talk.football, or football.general. There are a couple of ways of going about locating a group. (Actually, there are more, if you count telephoning someone and asking them where it is, or if you ask news.newusers.questions. These are, of course, last resorts, as it's more educational to find out for yourself.)

On any system that has nn available, there will be a companion program called nngrep. (You don't have to use nn to enjoy nngrep—it's a utility for everyone.) Typing nngrep -a pattern lists all available newsgroups whose name contains that pattern:

```
kibo@world> nngrep -a football
```

```
alt.sports.football.arena
```

```
alt.sports.football.mn-vikings
```

```
alt.sports.football.pro.atl-falcons
```

```
alt.sports.football.pro.buffalo-bills
```

```
alt.sports.football.pro.car-panthers
```

```
[...the list goes on, and on...]
```

```
clari.sports.football
```

```
clari.sports.football.college
```

```
clari.sports.football.games  
fido7.sport.football  
fj.rec.sports.american.football  
fj.rec.sports.football  
rec.sport.football.australian  
rec.sport.football.canadian  
rec.sport.football.college  
rec.sport.football.fantasy  
rec.sport.football.misc  
rec.sport.football.pro  
uiuc.sport.football
```

The reason you have to type the -a in the middle of the command is that otherwise, it'll only search the newsgroups you're already subscribed to—and if you're already reading them, you probably don't need to search for them.

---



**Tip:** If you decide you want to read rec.sport.football.misc but you're not subscribed to it, in the rn family you can type grec.sport.football.misc; in nn you type G, rec.sport.football.misc, and then j.

---

A shortcut in rn for finding all groups whose names contain a pattern, and then subscribing to them, is to type a followed by the pattern; for example, afootball will add the fifty groups just listed.

There's also a program named newsgroups on some sites that works like nngrep.

```
kibo@world> newsgroups football
```

Completely unsubscribed newsgroups:

```
clari.sports.football  
clari.sports.football.college  
fj.rec.sports.football  
rec.sport.football.australian  
rec.sport.football.college  
rec.sport.football.misc  
rec.sport.football.pro
```

[Type return to continue]

You'll notice the list is much shorter than the one we got before. This is because newsgroups will often leave out some or all

of the newsgroups on your system. nngrep is preferable for this reason.

One last thing to try before asking for help is to look in news.lists or news.answers for the various lists people post of newsgroups. These long documents give the names and one-line descriptions of most all the groups in the world. Here's a short excerpt from David Lawrence's list:

comp.admin.policy	Discussions of site administration policies.
comp.ai	Artificial intelligence discussions.
comp.ai.alife	Research about artificial life.
comp.ai.fuzzy	Fuzzy set theory, aka fuzzy logic.
comp.ai.genetic	Genetic algorithms in computing.
comp.ai.jair.announce (Moderated)	Announcements & abstracts of the Journal of AI Research.
comp.ai.jair.papers	Papers published by the Journal of AI Research. (Moderated)

However, when you get down to the alt groups, you'll notice that there is no "official" description for some of them—or rather, that some of them will have "joke" descriptions. Here's an excerpt from one of the various unofficial listings:

alt.artcom	Artistic Community, arts & communication.
alt.arts.ballet	All aspects of ballet & modern dance as performing art.
alt.ascii-art	Pictures composed of ASCII characters.
alt.ascii-art.animation	Movies composed of ASCII characters.
alt.asian-movies	Movies from Hong Kong, Taiwan and the Chinese mainland.
alt.astrology	Twinkle, twinkle, little planet.
alt.atari-jaguar.discussion	As opposed to simply alt.atari.jaguar.
alt.atari.2600	The Atari 2600 game system, not 2600 Magazine.
alt.atheism	Godless heathens.
alt.atheism.moderated	Focused Godless heathens. (Moderated)
alt.atheism.satire	Atheism-related humour and satire.
alt.authorware	About Authorware, produced by Authorware. So subtle.
alt.autos.antique	Discussion of all facets of older automobiles.
alt.autos.camaro.firebird	A couple of American sports cars.
alt.autos.rod-n-custom	Vehicles with modified engines and/or appearance.
alt.backrubs	Lower...to the right...aaaah!

alt.banjo

Someone's in the kitchen with Dinah.

No newsgroup is ever "on topic" 100 percent of the time, so don't be surprised to find that the people in an automobile group are all discussing Flintstones reruns. There is no rule that says people have to discuss what anyone wants them to, except for peer pressure—at least in the unmoderated groups. (The moderated groups carry fewer articles because the chatter generally isn't permitted there.) Also, some groups on the list might be ones that nobody's posted anything to in months.

Even with 9,251 groups (at my site, at this moment—your mileage may vary), you may still want to discuss something for which there is no group. If, after slaving away with the preceding methods, you can't find a place for the topic, and nobody else can, and you think there's a need for a new group. . . I feel a section heading coming on. . .

## Creating a New Group

Ever wonder where the dozen or so new groups that appear every day are coming from? (About a week ago, three *thousand* arrived at once here, due to circumstances beyond anyone's control!) There's a special kind of "control" article called a newgroup. When a Usenet article is posted with the proper headers, it spreads around the world alerting all the computer operators that someone wants a new group to be added, and then they get to decide whether to make it available at their site or not.



**Note:** This is different from the problem of adding an existing group to your site. This section discusses how to create a *new* group. If you want to read an existing group but can't get to it at your site—if you try to read rec.sport.football.pro but it's not there—what you should do is to ask your site's administrator or operator to add it, usually with the addgroup command. Now, back to our story.

If you were to just type in a newgroup control message (let's say you're creating rec.sport.mud-wrestling) and send it out, most likely nothing would happen because there are rules you're supposed to follow—or at least guidelines—and if you don't follow them, the administrators of thousands of sites will just ignore you. (Even if a few of them add the group at their sites, articles in the group won't be able to travel from those sites to others; you need to get the group on a large percentage of sites before articles can flow around the world with ease.)

The aforementioned *rules* govern the "main" ("Big Seven") Usenet hierarchies, which are comp.\*, news.\*, soc.\*, sci.\*, rec.\*, talk.\*, and misc.\*. The *guidelines* govern the "alternative" hierarchy, alt.\*, where anything goes, sort of. For other hierarchies—the special-purpose ones like bionet.\*, clari.\*, and so forth, and the regional ones like de.\* (Germany), fr.\* (France), capdist.\* (Schenectady/Albany/Troy), different situations apply in each case.

Getting a new group created in the main hierarchies—the "Big Seven"—is quite a challenge. First the idea should be discussed in groups related to your topic. Next a Request For Discussion (RFD, sometimes CFD for "Call"), that outlines the proposed group, should be drawn up and posted to news.announce.newgroups (which is moderated) as well as news.groups and the places where the idea was discussed. Now more discussion—in news.groups. At this stage there will likely be a flamewar with people who feel we don't need a new group arguing with people who think the proposed group has a bad name. (Take a peek into news.groups sometime if you want to experience the thrill of being a flamewar spectator; it's like alt.flame but slightly more intelligent.) About a month after the RFD is posted, you can hold a vote (for another month) where the object is to get 100 more "yes" votes than "no" votes, *and* the "yes" votes must outnumber the "no" votes by a factor of two. After the voting, the results will be posted (a list of who voted for what) and another flamewar will ensue in which random people claim there was vote fraud. If the group's vote was dominated by the "yes" people, the moderator of news.announce.newgroups (currently David Lawrence, mentioned earlier—the busiest person on Usenet) will issue the newgroup message.

Whew. And that's just the tip of the iceberg! For a full list of rules, see "How to Create a New Usenet Newsgroup," and for some more tips, "Newsgroup Creation Companion"; these articles are posted periodically to news.answers and other prominent places. (All the important FAQs and periodic postings are repeated in news.answers; news.newusers.questions also gets some of the most important ones.)

What this all boils down to is that whenever you suggest a new group for the Big Seven, there will be a lot of people who have an opinion as to whether it should or should not be created—and they will argue vociferously (and some of them are good at it because the same people do it for almost every proposed group). It's quite a pain.

For alt.\* groups, there are no hard-and-fast rules, but the days where anyone can issue a newgroup message and have it work are long gone—due to there being over 1600 alt.\* groups at the moment, operators are getting rather reluctant to add every alt.i-hate-barney-the-dinosaur.kill.kill.kill.die.die.die that someone sends down the pipe. alt.\* is often where groups that don't make it into the Big Seven are banished. Some of the most popular groups, like the alt.sex.\* hierarchy, are in alt.\* (as is my favorite, alt.religion.kibology). Some are moderated, but most are unmoderated, as elsewhere. The procedure for creating a new alt.\* group is similar to that for Big Seven, except there's no voting and discussion should take place in alt.config, not news.announce.newgroups or news.groups.

(Why is it alt.config and not alt.groups? I don't know either.) Basically, the group should be suggested and discussed in alt.config, and if people there generally seem in favor of it (allow a week or two for discussion), it would be a good time to send the newgroup message. The point of the discussion is to make the people who care about such things aware that there's support for your idea, so that when the newgroup arrives at various sites, the operators will recognize it as a good idea and add the group.

See "So You Want to Create an Alt Newsgroup" (posted periodically to news.answers, as well as alt.answers and alt.config) for more suggestions and guidelines.

Note that not only can anyone post a newgroup message to request the addition of a group, but anyone can post the evil twin of it—the rmgroup message. An rmgroup sends a note to the operator of each site telling them that someone would like a group removed. Like newgroup, it's usually ignored unless the person posting it has some authority and seems to have a valid reason.

I'm not going to discuss the actual mechanics of issuing a newgroup message here, as it's usually best if you just ask some noted authority to post it for you anyway, as their word will carry more weight. (I try not to get involved in such things, for political reasons, so please don't ask me.)

## Kibo

**--by Tod Foley**

Although it will take HappyNet months, maybe years, to improve all areas of daily existence in all possible ways, it will be obvious to the most casual reader that HappyNet is better than Usenet. Those who aren't casual readers—well, they will come to agree. In time, they will even love me. In fact, soon they will beg to love me! But I, Leader Kibo, want only the best for everyone. After all, I am one of the readers of Usenet, so I can make the readers of Usenet happy by making me happy FIRST. DEATH TO USENET! LONG LIVE HAPPYNET! TO THE MOON!

*--From Kibo's HappyNet Proclamation and Manifesto*

"Kibo," the Net persona of graphic artist and ontological guerilla James Parry, is a virtual god to many Usenet users, who read of his exploits (or just hang out with him) in any of the several newsgroups devoted to discussion of the man, the myth, and whatever other weird thoughts pop into their peculiar little heads. The g-word is not used lightly here, either; alt.religion.kibology is the third most popular religious newsgroup on the Net (tied with soc.religion.islam). To all appearances, this mysterious and quasi-mythical Kibo character possesses powers that are nothing short of awe-inspiring (alt.exploding.kibo and alt.imploding.kibo are both popular groups)—he even ran for president of the United States in 1992 on the electronic platform of alt.politics.kibo.

Late one evening this summer, Kibo and I played tag with each other through the ever-netsplitting IRC terrain; due to his immense popularity, we were forced to communicate through personal /msg commands. In his sometimes-serious, sometimes-surreal style, Kibo chatted with me about his unusual Net religion, his interests and philosophies, and his ideas on Usenet posts as communicative medium and performance artform.

**TF:** Your reputation precedes you to such a degree that at times (many times!) it must be an awful impediment. How do you feel about this?

**Kibo:** Well, the burden of pseudo-fame is slowly increasing. Recently I had to take three weeks leave of my e-mail, and now there are 715 messages in my mailbox. Mail is actually harder to keep up with than Usenet, as it requires more replies, and the tools for dealing with it aren't as good at weeding out the stuff you don't want to see.

**TF:** Are you working on any AI ki-bots to shoulder some of the infoload?

**Kibo:** I already have a few. I tried out my first one about six years ago. It was written in, believe it or not, BASIC. There's a crude piece of ELISP code that generates Kibo-style nonsensical slogans about Kibology, and a C translation of the same—which actually can put together a grammatical sentence on occasion.

**TF:** How long do you figure before most online persons have some sort of intelligent filtering agent?

**Kibo:** Depends on how you define intelligence. A newsreader killfile is a very, very primitive sort of agent; you can extend them to do a lot in the way of auto-selecting and managing Usenet. Eventually someone will write a good content-sensitive one, which will figure out what you want to read by searching the text of everything, and classifying it by what keywords it contains. Of course, the agent wouldn't *understand* what it's selecting for the user, but if it sees something containing cichlids and it knows I want to read about tropical fish and it can cross-reference the two in a dictionary, that's good enough.

**TF:** What grepwords are prominent in your filter?

**Kibo:** Depends on the groups. Often I include or exclude articles based on their authors. Certain words in subject lines are searched for, too. As far as searching the text of Usenet articles, about all I look for are variants of my name (good for finding followups to articles I've posted, .signatures that quote me, and so on). I don't filter my mail (but I may have to *soon*.)

**TF:** How did all this kibomania begin?

**Kibo:** Oh, it was when I was in college (the first of three colleges). I had already adopted the nickname "Kibo" (for reasons that are not clear any more) and when I was having dinner with two friends, in a Chinese restaurant in Troy, NY, one of them said, "There should be Kibology!" So we made up a stupid ol' doctrine and began proselytizing on the local computer conferencing system. One thing led to another. . . a few years after that, alt.religion.kibology was created on Usenet, and I started running that program to search for my name (people seem to find that last part the most interesting).

**TF:** Was the original idea of kibology computer-related *per se*, or was it just "James' type of weirdness?"

**Kibo:** I don't think it really involved computers. Hard to say, given that the doctrine was completely nonsensical and content-free. But computers were involved from the beginning. Nowadays alt.religion.kibology really has little to do at all with any sort of mock religion; it's more of a forum for me and people who seem not to mind being around me.

**TF:** Kibo and the wacky disciples of noncontent. What do y'all do there?

**Kibo:** I keep getting asked this about alt.religion.kibology and I really can't say. We talk about stuff that's not worth talking about, is the best I can describe it. The newgroup is an intellectual vacuum except that it's full of keystrokes. And, of course, there are a few Kibo groupies in there, and some genuine wackos. Occasionally, I'll do something like running for President of the World or other pointless activity. Keeps the peons amused, you know. They get sick of attending the gladiatorial games and Giant H Fights unless I throw 'em a bone. :-) Some of those people have *no* lives, let me tell you. A lot of them want to date *me*, and I have no social skills, so they must be really pathetic if they think I'm a step up. Lest I insult my followers too many times, I should point out to your readers that my followers know that I'm *always* sarcastic (except when writing chapters in books on how to use the network).

**TF:** Heh. What is it that people seem to want from being around you, or what do they want from their association with kibology?

**Kibo:** Entertainment? Someone to laugh at and/or with. I try to be both to everyone.

**TF:** A new form of public service... perhaps one day people will be professional kibos.

**Kibo:** I don't try to be *funny*, of course, because many people can do that better. I try to fail to be funny. That way people get to laugh at me. Someday I want to make a film worse than *Plan 9 From Outer Space*. I'm a little like Max Headroom in

a way; I pop in in random places on Usenet and make life more surreal for people there. Of course, I don't wear as much makeup as he does. And I don't do as many product placements as he once did.

*"A reporter from Australia recently asked me the meaning of life. I said it was like this: Suppose you like candy so you devote your life to getting all the candy in the world, and you put it all in one big pile, and there's so much candy in the pile that it becomes sentient and chases you around with big teeth, so you have to use the atomic bomb on it, and then there's candy everywhere. In retrospect, I think that may not have been nonsensical enough."*

—Kibo

**TF:** what are some of the basic tenets of kibology today?

**Kibo:** Um, um, um... um... can you ask me an easier question, like "How can we prove that quantum mechanics is a better model for the observed universe than general relativity?" I think it would go something like this: 1. Kibo is God and/or a bozo. 2. Spot is a stupid little puppy. 3. You're allowed. 4. Spot isn't—he's JUST a DOG! [damn line breaks.] 5. Damn the line breaks! 6. Kibo can make up any doctrine he wants in any interview because nobody cares anyway. 7. Kibo should be interviewed more often. And if I were a tree, what kind of tree would I be, you ask? Why, a binary tree, of course.

**TF:** Okay. What does kibo like?

**Kibo:** On Usenet, my favroite groups to read (besides alt.religion.kibology / alt.exploding.kibo / alt.politics.kibo / etc.) include alt.usenet.kooks, sci.physics, alt.folklore.urban, alt.conspiracy, and a slew of other ones where weird stuff happens. There are many where I just skim the list of subject lines and read only the 0.5 percent of articles that look deranged enough to be funny. For example, in alt.sex.fetish.feet I look for the posts by the guy who wants women to videotape their feet flooring their car's accelerator. (I am not making this up.) Other Usenet groups I read include alt.fan.cecil-adams, alt.fan.mike-jittlov, and alt.tv-babylon-5, but I only select those posts by the actual "celebs" the groups are devoted to (Ed Zotti, Mike Jittlov, and J.M. Straczynski); it's interesting to just read the posts by the minor celebrities who are on the Net. (*Major* celebrities, for example, actors and well-known musicians, generally don't participate in Usenet directly, if at all; they'd be swamped.) There's a twilight of people who are only well-known to a small enough degree that they can personally participate in the Net without being swamped. "J. Michael Straczynski" isn't a household name, but as creator of the show *Babylon 5* there are thousands of people on the Net who know who he is, and they have a group to go to, and in places like that you get this sort of action you'd get at a rock concert, where they've all come to see one person perform. (Yeah, I know, the thousands of fans also talk amongst themselves, but what I like to see is the "celeb" dealing directly with them.)

**TF:** Is that what they want you to do? This is performance—is it art?

**Kibo:** When I used to do standup comedy (as an amateur) I preferred to call it "performance art" because usually I tried to be bizarre and to confuse the audience, rather than simply trying to entertain them. My presence on Usenet is pretty similar; if I just tried to tell jokes or be "funny," that would probably come out pretty lame. I like to keep 'em guessing. Right now I'm trying to start a panic that there's a computer virus named READ.ME that attaches itself to any program you download, and I'm trying to teach a certain poster in sci.physics the word *phlezofigle*.

**TF:** OoOoOOOOOoOoOoOoOo...

**Kibo:** Geez, that sounds dumb, doesn't it?

**TF:** Actually, the one about the READ.ME virus is catchy. . . .

**Kibo:** One of the things that passes for interactive entertainment on Usenet is "trolling for newbies." Trolling is the art of saying something completely outrageous (for example, "William Shatner was only on Star Trek because he was married to the producer, Gene Roddenberry!") and having people react to it; most people will realize that that must have been said in jest, but a few, who lack the "bozo detection circuit" in their brains, will say completely serious things like NO HE WASNT !!!!!1 ARE U STUPPID OR SOME THING ??????// Again, this sounds really pathetic when described instead of actually performed. Hmm, maybe this should tell us something. Usenet, the interactive entertainment form where patheticness is entertaining.

**TF:** So we're talking guerilla surrealist performance art here.

**Kibo:** I guess that would be a good way to put it. Sort of like those evil mimes who follow people around and make fun of them, only combined with the fine performance art talents of Yoko Ono or Chris Burden. (If any of the readers out there know who Chris Burden is or what he did with a VW Beetle, they win a SPECIAL PRIZE!!!) Now I'm talking like Zippy the Pinhead. Ever notice that CARTOON CHARACTERS say lots of WORDS in BOLDFACE in each SENTENCE? YOW! My socks are shaped like ETHERNET CONNECTORS! Have I got a FIBER-OPTIC NERVE yet? Also, I have the *most* over-inflated ego in the *whole world!* I'm also the most humble person on earth. Like the way Barney is so wuvvable that you just want to hate him, I'm so obnoxious that you just want to wuv me.

**TF:** I know who Burden is—one of his coolest works was one where he sat the audience members on high ladders in a deep room, and then started flooding the room with water, and then passed a high-voltage electrical current through the water. . .

**Kibo:** Yeah, that's Burden. He made his chest explode once with a welding arc, and once he nailed himself to a VW's hood and someone drove it into a wall. I hope the NEA gave him enough money to cover the hospital bills. YOU KNOW I THINK IT'S SICK THAT THE NATIONAL EDUCATION ASSOCIATION NOT ONLY FUNDS SESAME STREET BUT ALSO THESE SICKO ARTISTS !!!!!!!!!1111!!!!!!@!!!!11

I meant to make all those typos.

**Kibo:** Have you read my .sig and the HappyNet manifesto? Those are probably the two pieces of Net activity for which I'm best known. I used to get two complaints about my .sig for every compliment, and now the ratio has more than reversed. It's eerie, especially since all I ever do is make the .sig more obnoxious. Apparently I passed the point of maximum obnoxiousness and the sign bit flipped around (two's complement) and now it's maximally nonobnoxious. Eek, I made a binary joke. Shoot me. STOP ME BEFORE I TURN INTO A NERRRRD!

"While watching today's Star Trek: The Next Generation episode for the fifth time, I noticed that the controls on the Transporter were slightly different than--"\*\*\*BANG\*\*\*"--OW!" Thud.

**TF:** Your mail didn't include a .sig. send me one. While we're on the subject, send me some bio stuff on yourself as well. I've read the HappyNet manifesto, but where is it located for our readers to FTP?

**Kibo:** ftp.std.com, login as anonymous, password *your address*, cd to pub/alt.religion.kibology. There's a whole mess (and I mean *mess*) of stuff there. The HappyNet directory there has the manifesto. I'll mail the .sig.

I should point out that the HappyNet manifesto was intended as a parody of what people perceive to be Usenet's weaker points, and it's slowly becoming less fictional. Someone's going to have to seize control of alt.\* someday and clean out a few of those thousand groups that carry no traffic.

**TF:** Why is Usenet your favorite service/environment?

**Kibo:** Usenet has everything. It has discussion on every *possible* topic (all the way down to "I'm turned on by women who floor gas pedals really hard") and every intelligence level (serious discussions of high-energy particle physics to the latest Baywatch episode).

**TF:** Do you think that any other form of Net service could sustain kibology as well as Usenet does?

**Kibo:** I'd love to have the Usenet equivalent of a cable public-access show, where I'd just post little QuickTime movies of me that you could play on your screen. I'm going to start posting *audio* clips soon since people have been requesting it.

Video would be better, of course, because I could then put in product-placement shots. "Look! Kibo's wearing Doc Martens! I guess those shoes really do make you more handsome!" I was recently approached by an ad agency that represented a store that sells Doc Martens and assorted other grunge-type clothing...

**TF:** Unreal!

\*\*\* **Kibo:** No such nick/channel

**TF:** /ping kibo

\*\*\* **Kibo:** No such nick/channel

**TF:** /quit

After more than two hours of IRC-hopping, Kibo had disappeared from the Internet Relays. I bounced up a few times, hoping to find him there, but my /pings yielded nothing. Upon checking my e-mail, however, I found two messages of note: The first was Kibo's .sig file (yes, it's just as obnoxious as promised), and the second was a short note from the deity himself:

Date: Wed, 10 Aug 1994 02:44:09 -0400

From: kibo@world.std.com (James "Kibo" Parry)

To: asif@well.sf.ca.us

Subject: what I was saying

Well, looks like the Split will be permanent. This is the last thing

I said in response to the Doc Martens discussion:

Nothing ever came to pass of it, but the idea of an agency rep talking a chain-store owner into using Kibo to advertise Doc Martens on MTV felt like it was a sign of the Apocalypse.

And then the Net broke when I said "Apocalypse."

-- K.

(If the Net doesn't heal in about ten minutes, I'm going to bed.)

& r

To: asif@well.sf.ca.us kibo@world.std.com

Subject: Re: what I was saying

> And then the Net broke when I said "Apocalypse."

There's a bang of an ending if ever I've seen one...!

thanx

LVX

TF

## Searching For Things

And now, we come to the final section of this chapter, the one everyone's been waiting for. Several years ago, after I became slightly well-known on the Net, I noticed that people were quoting me in their signatures. I started searching all of Usenet (that is, the text of all articles) for my name. That had the added advantage that I could find all the followup articles people made to my posts, a real timesaver. But its greatest effect, wholly unexpected, was that after I caught people discussing me in random newsgroups and jumped into the conversation, a legend sprung up that I would respond to any mention of my name on Usenet. It ain't so, Joe. I am narcissistic enough to want to read what people are saying about me,

but responding to all the articles that mention me anywhere (up to two hundred a day) would be no fun at all. (I tend to perform the full search once a week and print out a long list of all the results to read at my leisure.)

This is a technique that is, in my opinion, *the* most powerful way to read Usenet. Remember that searching for information of any sort is as easy as searching for your name. It's not the most straightforward way, but if you know you want to read all articles about, say, marshmallows, you can search for the word marshmallow across all groups and get just the few articles that actually mention the squishy little things. I'm going to give you various approaches to do just that; not all of them will work on your site, but some of them should do something for you. (They all work for me, but then my account on world.std.com is on a machine with a lot of resources available.)

## Searching *Subject:* and *From:* Lines

My single favorite feature in nn (and one you can use even if you normally use another newsreader) is its ability to merge all existing articles into one enormous *pseudogroup* (with about a million articles!) and then rapidly find the ones with particular header lines. You can't search the *text* of the articles, but this method is very fast (relatively speaking, anyway) and simple to do. Here are three command lines that search *From:* headers:

```
nn -mxX -nPerot all
```

```
nn -mxX "-nRoss Perot" all
```

```
nn -mxX -nSquiggy rec.sport.
```

The first command searches for all articles from anyone whose "real name" includes Perot. (The "real name" is what usually shows up after the e-mail address. nn searches only the address in articles where no real name is given, so try the real name first, and try the address if all else fails.) The second version is a more fussy one that will only find people named "Ross Perot" (but not "Ross Q. Perot"). I prefer just to specify the last name or part of it for this reason. Another reason is that nn shortens the names it displays—Charles Nelson Reilly may show up as Charles N Reilly or C N Reilly—and searches are performed on the *short* version.

The third will search all articles from Squiggy in the groups that start with rec.sport. The documentation says you can say rec.sport.\*, but leaving the asterisk off actually works better for me.

Similar searches can be made on the *Subject* header lines:

```
nn -mxX -spizza all
```

```
nn -mxX "-spepperoni pizza" all
```

```
nn -mxX -splutonium sci.
```

On the machine I use, these searches generally take 5 to 15 minutes, depending on how busy the machine is. It may be slower or faster for you, depending on your machine's speed and the number of articles that exist at your site (on some machines they "expire" and evaporate in just a few days, and on others they stick around for weeks). Here's a sample screen of results from the "plutonium" search:

```
MERGED NEWS GROUPS:  66 ARTICLES
```

```
a nelson@usuhsb      26  >>chlorine vs. plutonium
```

```
b Jay Mann          16  >>>>>
```

```
c Jay Mann          16  >>>>>
```

```
d Jay Mann          16  >>>>>
```

```
e robert wood       38  >>>>>>
```

```

f robert wood      38  >>>>>
g robert wood      38  >>>>>
h Bob Savas        10  >>>>>
i Bob Savas        10  >>>>>
j Bob Savas        10  >>>>>
k.James Lynch      4   Abian Vs. Plutonium
l.Benjamin Carter  8   >
m B. Alan Guthrie  61  >>>Purifying Plutonium
n B. Alan Guthrie  61  >>>
o B. Alan Guthrie  61  >>>
p Ludwig Plutoniu  369  PLUTONIUM ATOM TOTALITY THEORY FAQ (1 OF 2)
q Ludwig Plutoniu  369  -
r.Ludwig Plutoniu  369  -
s.Ludwig Plutoniu  369  -
t David Erwin      13  >
u David Erwin      13  >

```

```
-- 06:08 -- SELECT -----63%-----
```

Read 20542 articles in 22 seconds (62 kbyte/s)

All you need to know about nn at this point is that you should press the letters corresponding to the articles you want to see, and then press the Spacebar. You can select the whole screenful of articles with @; and if you want to save all the selected articles, press S (Shift-s), then type the name of a file to save to, press y and +.

## **rn Text Searches**

Remember when I was talking about killfiles, and I said that /marshmallow/a:j would kill all articles mentioning marshmallows anywhere? Well, those commands don't have to be in an rn (or trn, and so on) killfile, but they can be typed in while reading articles. (Then the command applies to the current articles only, because the command isn't saved to the killfile.) In rn, leaving off the :j will simply find the next matching article.

```

/marshmallow/      find the next unread article with that subject
/marshmallow/a     find the next one with "marshmallow" anywhere
/marshmallow/ar    add "r" to search unread and old articles too
?marshmallow?ar    find, going backwards

```

Searching backwards with `?` is most useful if you type `$` first, which will move you to the newest article in the group; then, `?` will find you the most recent matching article.

Because `trn`, unlike `rn`, has those nice menus of articles' subjects, you can do a command like `/marshmallow/a` and then press `Shift-X` to junk all the articles it didn't find. If you've got `trn` configured with the options I gave several pages ago to make it never show you any articles you didn't select, you can make a killfile that says `/marshmallow/a:+` (the plus sign selects articles) and, coupled with the `-X1X>` option, all you'll ever need to do is press the Spacebar. You'll see all the articles mentioning marshmallows, and nothing else.

Automatically selecting based on Subject lines or From lines is quite quick in `trn` if you use this method—but the `a` modifier will slow things down considerably because `trn` will have to read every article all the way through. I have a global killfile that auto-selects anything containing a few variants of my name, anything written by certain people I like, and anything with a subject line containing certain keywords. This, in combination with a few extra killfiles for specific groups, saves me a lot of time when I use `trn`.

## **grep: Searching at the Lowest Level**

`grep` is one of the oldest and most basic UNIX programs. If your machine isn't running UNIX, stop, do not pass go, please proceed to the next session. This magical `grep` will search files for strings. (Why the funny name? It's short for `g/re/p`, which is a `sed` command to Globally find a Regular Expression and Print it.) There are several variants of `grep`; your machine probably has `egrep`, which is faster (and has extra options), and might have `agrep` (which is the fastest and has the most options). You can substitute any of them for the method detailed here.

This method is going to get tricky fast, so please bear with me. Because `grep` searches files, and because each Usenet article is stored in a file somewhere, the first thing you need to do is to find where the files are. On some sites the files are actually stored on a neighboring machine, and the newsreader programs retrieve them with a protocol called NNTP. If your machine uses NNTP, this method won't work. (If your newsreader is named `rrn` or `xrrn` or `trrn`, the extra `r` for *remote* means you're using NNTP. Often you'll use NNTP without being able to tell, though.)

Look through various directories until you find where the Usenet files are stored. `/usr/spool/news`, `/usenet/spool`, `/var/news/spool`, and so on, are places to try (it's different on every system). Sometimes the `whereis` `spool` command tells you some likely candidates. You'll know you've found the right directory if it contains subdirectories named `alt`, `comp`, `talk`, and so on. If you can't find such directories anywhere, you're probably on a site that uses NNTP, and you'll have to skip to the next section.

On my site, the articles are in subdirectories in `/usr/spool/news`. This means that the articles for `sci.physics` are in `/usr/spool/news/sci/physics`; each filename is a number indicating in what order the articles arrived. To search all articles in `sci.physics` for my name and display all the results at once, I type:

```
grep -i "kibo" /usr/spool/news/sci/physics/*
```

The `-i` means it will match upper- and lowercase letters. The results will be a list of lines from files containing `kibo`, with the name of the file before each. You may also see a bunch of error messages scrolling past; don't worry about them.

The biggest limitation of `grep` is that it can search only the files in one directory. Because each newsgroup is stored in a different directory, we have to introduce a few other programs into the mix.

```
find /usr/spool/news/* -type f -print | xargs grep -i "kibo" >output
```

Here's where it gets complicated. The `find` program, with the preceding arguments, will find all files (`-type f`) in the Usenet directories, meaning all files in all groups. It will print out this horribly long list and pass it to program `xargs`, which will run `grep` on each of those files. The `>output` part saves all the results (many pages) in file output.

Major drawback: The preceding command line will take at least a few hours to run (maybe all day), and it may spit out thousands of spurious error messages you'll want to ignore. Here's how to work around that; create a file named `myscript` containing these three lines:

```
#!/bin/sh
```

```
rm -f output
```

```
find /usr/spool/news/* -type f -print | xargs grep -i "kibo" >output
```

Now, do `chmod 700 myscript` to make `myscript` into an executable script. To start this script running so that it will chug away overnight—even if you're not logged in!—and to make it not display any error, run this command:

```
nohup nice -10 myscript >&/dev/null
```

`nohup` is a program that runs other programs in such a way that they won't stop when you logout and go to bed. (On some machines, the administrators may terminate all your processes when you leave. If that's so, you'll have to stay logged in while it runs.) `nice` is a program that runs other programs at a low priority, to avoid annoying the other users of the machine. (If you have a workstation all to yourself, you can leave out `nice -10`. Because I run the program overnight while I'm not logged in, I don't care if it runs slowly.) Finally, `>& /dev/null` sends all the spurious errors to Richard Simmons. No, just kidding—it sends them into a very deep hole where they'll never come out.

So, what we've done is to tie a handful of little programs—UNIX commands—together to do something reasonably powerful with just a few lines of typing. This is why I like UNIX. In just thirty seconds, you can write a script that will take all day to run.

### Larry Wall's *clip*

Larry Wall, author of several important programs (including the original `rn` and the programming language Perl) is also known on Usenet for the ability to find mentions of his name anywhere. He uses a program named `clip`, and not surprisingly, it's written in Perl, and not surprisingly, Perl is optimized for doing things like searching files for text. (It's useful for millions of other things, too.)

Because `clip` is six pages long, I won't show the code here. You can get `clip` via anonymous FTP from `ftp.netlabs.com`, and the filename is `/pub/outgoing/clip/clip`. (Larry's address is `lwall@scalpel.netlabs.com` in case you need help finding it, but please don't send him too much adoring fan mail—he's busy enough from getting even more mail than I do.)

The patterns `clip` searches for, along with other parameters, are stored in a file named `.cliprc` in your home directory. Because perl is different from `grep`, and so on, the patterns are specified in a different manner than the others we've been looking at. Here is a simple `.cliprc`:

```
# How much do we slow down the system?

$MAXLOAD = 10;

# Newsgroups to be skipped.

&NGSKIP( <<'END' );

    &skip if /alt\.religion\.kibology/;

END

# Patterns I'm interested in scanning for.

&SCANNER( <<'END' );

    &hit if /\bKibo\b/i;

    &hit if /(James|Jim)( W.|) Parry/i;

END
```

The `\b` represents a word boundary, so that it will find `Kibo` and not `Skibo` or `Kibozo`. The complicated line will match

James W. Parry, Jim Parry, James Parry, and Jim W. Parry. (As if anyone's going to call me that last one. Ugh.) This is just the tip of the iceberg; with Perl, as with grep, if you know what you're doing it can get quite byzantine. Here's an example from Tom Christiansen's .cliprc file:

```
if (/\\btchrist\\b/i || /tom[^\0]christ(ia|e)ns[eo]n/i) {{
    next if /\n\s*Tom Christiansen\s+tchrist\@convex.com/;
    &hit;
}}
```

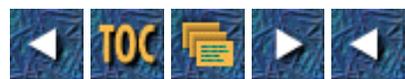
A similar program to clip is newsclip, by Brad Templeton (brad@clarinet.com). You may remember him as the moderator of rec.humor.funny a few years ago. Currently he runs the ClariNet electronic newspaper service (that is, AP and Reuters news that is broadcast to sites paying for the clari.\* groups); you don't have to be a ClariNet user to use newsclip, of course. newsclip, being even fancier than clip (and harder to install), can't really be described here; suffice it to say that you can FTP it from ftp.clarinet.com, in sources/nc.tar.Z, and that it's got many excellent features for selecting or killing articles—you can even patch rn or rrn to let you filter articles with newsclip while reading them.

## The Shape of Agents to Come

Well, so far, I've talked about some methods for searching text for strings. That's fine if you're looking for a *word*, but what if you're looking for a *concept*? How do you find "all articles relating to philosophy applied to dialectical materialism?" And how do you even deal with finding all possible misspellings of "marshmallow?" What do you do if, even when you narrow your search as much as possible, it still turns up too much to read—how can you prioritize the articles in order of importance?

Such questions will be addressed in the future by the development of *intelligent agents*—not just for Usenet, but for other hypertext and electronic news services. These tools for managing information will need to be able to imitate our own preferences for what to read and figure out what we'd like to see. The amount of programming that will be involved in these compares to the little scripts given here the way a zeppelin does to an amoeba.

Someday, while your grandchildren are scanning the world hypertext network for information on their favorite musicians, you'll be able to tell them you remember the good ol' days, when there were only nine thousand newsgroups, and you had to search them by hand, while walking ten miles to school in the snow. . .





- [10](#)
  - [World Wide Web](#)
    - [by Billy Barron](#)
    - [Locating Items in WWW](#)
      - [Understanding URLs](#)
      - [Hotlists and Personal HTML Documents](#)
      - [Subject Hierarchies](#)
      - [Organizational Hierarchies](#)
      - [Geographical Hierarchies](#)
      - [New Web Sites](#)
      - [Indexes](#)
    - [Your Own Home Page](#)
    - [Server Software](#)
    - [Page Design Issues](#)
      - [Lists versus Full-Text](#)
      - [Number of Links](#)
      - [Links Back](#)
      - [Inline Graphics](#)
      - [Client Independence](#)
      - [CGI Scripts](#)
      - [Forms](#)
      - [Image Maps](#)
    - [Client Configuration](#)
      - [Performance](#)
      - [Mirroring Pages](#)
      - [Related Viewers](#)
      - [Downloading Images/Icons](#)
    - [Spiders](#)
    - [Keeping Current with Future Development](#)

# 10

## World Wide Web

by **Billy Barron**

The World Wide Web (WWW) is growing at an explosive rate, and even we Internet gurus are having trouble keeping up with all the new developments. New WWW functions and interesting WWW pages are appearing on a daily basis. In May 1994, the traffic of the Web was in the top 10 protocols and had even surpassed Gopher. However, we gurus have developed some tricks in dealing with the problems of the explosive growth of the Web.

In this chapter, we will look at these tricks as well as other tricks. Other tricks involve faster ways to navigate the Web, designing Web pages, and finding out what resources are available.

This chapter also assumes that you, the reader, have acquired and installed a WWW client before reading. All known clients are documented in a file <http://info.cern.ch/hypertext/WWW/Clients.html> (this is a URL specification, which is discussed later in this chapter). However, you will be unlikely to look at this document without a client. Therefore, you would best start by asking a friend or associate to use their client and work up from there.

### Locating Items in WWW

One of the largest problems with the WWW system as it stands today is that it is difficult to find resources, even for the WWW guru. While Veronica does a nice job of indexing Gopherspace, no direct equivalent exists for the Web that has the completeness of Veronica. However, this is because it is easier to index Gopherspace. Veronica can just get all of the Gopher menu and know what is there.

With the Web, however, an indexer has to download the text of every document. Then it has to see if the document is HTML. If so, the document must be parsed and interpreted to get titles and to see if it contains links to other documents. It is a very resource-intensive process.

The guru does have some tricks that can be used to find resources in the Web. None of them will guarantee to find a particular resource, but they do help greatly in the search. Before getting to the tricks, though, you need to have a firm grasp on some background information.

### Understanding URLs

To be a guru of the Web, a solid understanding of Uniform Resource Locators (URLs) is an absolute must as a first step. Every item on the Internet has at least one URL that defines its location. By having the URL of an item, you should be able to find and access it, provided the resource is up and running and there are no security barriers in your way.

The official document on URLs is at

`file://info.cern.ch/pub/www/doc/url-spec.txt`

This is itself a URL, but it will be explained in a moment. It contains a very detailed specification of them. We will not discuss them in such depth, but we will cover the most common URLs that will cover almost any case you will run across while using the Internet. Also, a document called a "Beginner's Guide to URLs" is available at

<http://www.ncsa.uiuc.edu/demoweb/url-primer.html>

URLs are usually made up of a protocol type, an address, an optional port number, and a path. The protocol types are listed in Table WW.1. The address is the address the server runs on and can be specified as either a hostname or a numeric IP address. The port number is the number of IP port that the server is running on. If the server is running on the standard port for that protocol, the port number is unnecessary. If the server is running on a non-standard port, the port must be specified. For example, a HTTP (HyperText Transfer Protocol) server has a standard port number of 80, so the port does not need to be specified in the URL if the HTTP server is running at port 80. By the way, HTTP is the protocol that is used by Web clients and servers, so an HTTP server is a WWW server. The path is the path to the particular item the URL is referring to.

**Table 18.1. Protocol types.**

<i>Protocol Type</i>	<i>Description</i>
http	WWW server
gopher	Gopher server
ftp or file	FTP server
mailto	E-mail
telnet	Remote login
wais	Wide Area Information Server

The basic formats of a URL are

*protocol://hostname:port/path*

or

*protocol://hostname/path*

To best understand URLs, let's run through a few examples. <http://www.utdallas.edu/> refers to the main WWW menu for the University of Texas at Dallas (UTD). First, we see the protocol is http, so we know it is a WWW server. Second, we see the node name is www.utdallas.edu. Since no port was specified, this resource uses the default HTTP port of 80. The path is just a simple /. However, sometimes you may see it referenced as <http://www.utdallas.edu> instead. The ending / is assumed if it is not there, so the two forms are equivalent.

My personal Web page can be found at <http://www.utdallas.edu:80/acc/billy.html>. The protocol and hostname remain the same as our previous example. The port number is specified as 80 here, though it is unnecessary, and the path is /acc/billy.html.

<gopher://yaleinfo.yale.edu:7000/11/Libraries> is a URL pointing at the Internet Library List at Yale University. The protocol is Gopher and the hostname is yaleinfo.yale.edu. The port is 7000 instead of the standard Gopher port of 70. The path is /11/Libraries. However, to really understand the paths for Gopher, you need to understand the Gopher protocol. The short of it is that the 11 is a type specifier and /Libraries is the Gopher path. The type specification 11 means that it is a Gopher menu.

<ftp://ftp.utdallas.edu/pub/staff/billy/libguide/libraries.asia> is a URL to a document covering the Internet Libraries in Asia. The protocol is FTP. The hostname is ftp.utdallas.edu and the path to the document is /pub/staff/billy/libguide/libraries.asia. In this case, it is anonymous FTP. Ways to specify userids and passwords exist, but they are rarely used.

<telnet://dra.com> is a URL to a telnet session to the host dra.com. There are ways to specify users and passwords with this URL too, but the use of this feature is rarely seen, so look in the URL documentation if you need to do this.

Another URL you may run across is mailto. For example, it might be used like <mailto:billy@utdallas.edu>. If it is

selected by the user, an e-mail message will be sent to [billy@utdallas.edu](mailto:billy@utdallas.edu) after you type it in. However, not all clients support this feature, so it should be used with care. It is documented in the page <http://siva.cshl.org/email/index.html>.

It is important to understand URLs because the Web is totally based on them. Without a firm understanding of them, becoming a guru of the Web is impossible—so take time to learn them if you don't understand them fully at this time. You will be seeing quite a few URLs.

In the future, there will also be Uniform Resource Names (URNs). A Uniform Resource Name will be like an ISBN number on a book. Each item (defined as having the same URN) can exist in several locations on the network. Each of these locations is defined by a URL, which is similar to a call number in a library.

## **Hotlists and Personal HTML Documents**

Any Web explorer has run into the problem where a very interesting resource is found once, and then you can never find it again. It can be a most frustrating experience. The reason this happens more in the Web than Gopher or other systems is that hypertext, especially poorly written hypertext, tends to allow you to drift off on unrelated tangents. Eventually, you will find some places interesting, but you have no idea how you arrived there.

Currently, there are two solutions to the problem. The first is the use of the Hotlist or Bookmark feature of your Web client. When you find a resource that is very useful, you just save it to your hotlist. Then whenever you want to go back to this location, you can just pop up your hotlist and select the item.

The second method is to create your own Web document in HTML (HyperText Markup Language), which contains useful links. Obviously, this is quite a bit more work. However, the big advantage of it is that the Hotlist is just a sequential list of items. In your HTML document, you can have headers, notes, comments, and even pictures. You can also take it one step further and make a set of pages like this all linked together.

The one problem these solutions do not solve is the movement of items around the Web. Also, sometimes items totally vanish. Unfortunately, at this stage, even the guru doesn't have a good solution to the problem. However, being a guru, if you move a page of your own, you should leave a page at the old URL for a period of time, informing the users of the move.

## **Subject Hierarchies**

Several different individuals or groups have pages that attempt to break access to WWW pages down by subject classifications. A Web guru needs to know about these to be able to find information fast. However, all of these classification systems are very incomplete. This is unlikely to change because nobody can keep up with the growth or even the sheer size of the Web.

These classification schemes can generally be divided up into two basic categories. The first is controlled by an individual or small group of people. These people look at pages around the Web and find useful pages and include them under a subject classification. These effects will always be incomplete because the authors can't possibly deal with the hundreds, if not thousands, of new Web pages that are being created every day.

The other category is a self-registration mode. Basically, the author provides the users of a page with a form. This form enables the user to enter a new page into the scheme and requires the user to enter information about the new page. Some even enable the user to create new subject headings. These systems tend to contain more pages written by WWW gurus and less by novices. None of the systems I have seen to date have a good security/verification system. There is the potential for abuse, where someone creates links to pages in the wrong category. Also, many of the pages that are included in such systems have little or no value to most people.

In any case, both types of systems are useful and should be looked at when trying to find information in the Web. I will provide a list in Table WW.2 of the currently available subject classification schemes. Over time, more will

appear, so keep in mind that this list is probably not comprehensive by the time you read it. In addition, some of these schemes will eventually die off. However, for the guru, this is the nature of the game called Internet.

**Table 18.2. Subject classification schemes.**

<i>Name</i>	<i>URL</i>
CERN - WWW by Subject	<a href="http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html">http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html</a>
EINet Galaxy	<a href="http://galaxy.einet.net/galaxy.html">http://galaxy.einet.net/galaxy.html</a>
Joel's Hierarchical	<a href="http://www.cen.uiuc.edu/~jj9544/index.html">http://www.cen.uiuc.edu/~jj9544/index.html</a>
Mother-of-all-BBES	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/summary.html</a>
NCSA Meta-Index	<a href="http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/MetaIndex.html">http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/MetaIndex.html</a>
Netlink	<a href="http://honor.uc.wlu.edu:1020">http://honor.uc.wlu.edu:1020</a>
Nova-Links	<a href="http://alpha.acast.nova.edu/start.html">http://alpha.acast.nova.edu/start.html</a>
Project DA-CLOD	<a href="http://schiller.wustl.edu/DACL0D/dacloD">http://schiller.wustl.edu/DACL0D/dacloD</a>
Yahoo	<a href="http://akebono.stanford.edu/~jerry/bin/yahoo">http://akebono.stanford.edu/~jerry/bin/yahoo</a>

## Organizational Hierarchies

In addition to the subject hierarchies, some sites provide a set of pages that break down the WWW by the types of organizations that have pages. For example, it might list Universities, Corporations, and Non-Profit Organizations. Then Corporations are often subdivided into subhierarchies like Accounting, Aerospace, and Chemicals.

The organizational hierarchies are useful when you are looking for a particular organization to find out about their services and/or products. Also, it is helpful when you are shopping for a particular type of service or product to be able to locate information about it on the Web.

Just like the subject classification schemes, the organizational hierarchies fall into the same two basic categories. A list of known organizational hierarchies follows (Table 18.3).

**Table 18.3. Organizational Classification Schemes.**

<i>Name</i>	<i>URL</i>
American Universities	<a href="http://www.clas.ufl.edu/CLAS/american-universities.html">http://www.clas.ufl.edu/CLAS/american-universities.html</a>
Best Commercial Sites	<a href="http://wings.buffalo.edu/contest/awards/money.html">http://wings.buffalo.edu/contest/awards/money.html</a>
Commercial Services	<a href="http://tns-www.lcs.mit.edu/commerce.html">http://tns-www.lcs.mit.edu/commerce.html</a>
Community Colleges	<a href="http://gopher1.faytech.cc.nc.us/">gopher://gopher1.faytech.cc.nc.us/</a>
Companies	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/09/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/09/summary.html</a>
Computer Science Depts	<a href="http://www.cs.cmu.edu:8001/Web/People/anwar/cs-departments.html">http://www.cs.cmu.edu:8001/Web/People/anwar/cs-departments.html</a>
Corporations	<a href="http://www-external.hal.com/pages/hops.html#CORP">http://www-external.hal.com/pages/hops.html#CORP</a>
Freenets	<a href="http://herald.usask.ca/~scottf/free.html">http://herald.usask.ca/~scottf/free.html</a>
Government Agencies	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/11/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/11/summary.html</a>

Museums	<a href="http://www-external.hal.com/pages/hops.html#MUSE">http://www-external.hal.com/pages/hops.html#MUSE</a>
Organizations	<a href="http://www-external.hal.com/pages/hops.html#ORGS">http://www-external.hal.com/pages/hops.html#ORGS</a>
Other Colleges	<a href="http://www.dartmouth.edu/Pages/otherSchools.html">http://www.dartmouth.edu/Pages/otherSchools.html</a>
Research Centers	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/10/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/10/summary.html</a>
Universities and Schools	<a href="http://www-external.hal.com/pages/hops.html#UNIV">http://www-external.hal.com/pages/hops.html#UNIV</a>

## Geographical Hierarchies

Another scheme people are using to break down the Web is to list all the Web sites in a particular geographic location or region. This can be useful for several reasons. First, you might only need information about businesses or educational institutions in a specific geographical region. Or you might be planning a trip and need information about a city. You may also just need information from your own city such as building codes, events, or even a restaurant. The possibilities are endless.

You will find text based and graphical interfaces to geographical hierarchies (see Table WW.4). Of course, if you have a text based client, you cannot use the graphical interfaces that many Web pages have. Each has its merits and uses. Additionally, you will find that some geographical regions have good pages of this nature, and other regions are voids—you will be unable to locate any geographical scheme that documents the region in any depth.

**Table 18.4. Geographical Classification Schemes.**

<i>Name</i>	<i>URL</i>
Alberta	<a href="http://www.sal.ists.ca/services/w3_can/alta.html">http://www.sal.ists.ca/services/w3_can/alta.html</a>
British Columbia	<a href="http://www.sal.ists.ca/services/w3_can/sbc.html">http://www.sal.ists.ca/services/w3_can/sbc.html</a>
Canada	<a href="http://www.sal.ists.ca/services/maps.html">http://www.sal.ists.ca/services/maps.html</a>
Connecticut	<a href="http://www.cs.yale.edu/HTML/YALE/MAPS/connecticut.html">http://www.cs.yale.edu/HTML/YALE/MAPS/connecticut.html</a>
Delaware	<a href="http://www.udel.edu/delaware/map.html">http://www.udel.edu/delaware/map.html</a>
Europe	<a href="http://www.eeb.ele.tue.nl/maps.html">http://www.eeb.ele.tue.nl/maps.html</a>
Florida	<a href="http://www.cis.ufl.edu/home-page/fl-servers/">http://www.cis.ufl.edu/home-page/fl-servers/</a>
Indiana	<a href="http://www.music.indiana.edu/tourist/">http://www.music.indiana.edu/tourist/</a>
Iowa	<a href="http://caesar.cs.uiowa.edu/iowa/">http://caesar.cs.uiowa.edu/iowa/</a>
Kentucky	<a href="http://www.uky.edu/kentucky-network-services.html">http://www.uky.edu/kentucky-network-services.html</a>
Manitoba	<a href="http://www.sal.ists.ca/services/w3_can/man.html">http://www.sal.ists.ca/services/w3_can/man.html</a>
Massachusetts	<a href="http://donald.phast.umass.edu/misc/mass.html">http://donald.phast.umass.edu/misc/mass.html</a>
Mexico	<a href="http://info.pue.udlap.mx/www-mex-eng.html">http://info.pue.udlap.mx/www-mex-eng.html</a>
Netlink	<a href="http://honor.uc.wlu.edu:1020/-ge">http://honor.uc.wlu.edu:1020/-ge</a>
New Jersey	<a href="http://www.stevens-tech.edu/nj.html">http://www.stevens-tech.edu/nj.html</a>
New York	<a href="http://wings.buffalo.edu/world/nywww.html">http://wings.buffalo.edu/world/nywww.html</a>
North Carolina	<a href="http://sunsite.unc.edu/nc/nchome.html">http://sunsite.unc.edu/nc/nchome.html</a>
Oregon	<a href="http://www.willamette.edu/~tjones/Oregonmap.html">http://www.willamette.edu/~tjones/Oregonmap.html</a>

Quebec	<a href="http://www.sal.ists.ca/services/w3_can/qc.html">http://www.sal.ists.ca/services/w3_can/qc.html</a>
Saskatchewan	<a href="http://www.sal.ists.ca/services/w3_can/sask.html">http://www.sal.ists.ca/services/w3_can/sask.html</a>
Southern Ontario	<a href="http://www.sal.ists.ca/services/w3_can/sont.html">http://www.sal.ists.ca/services/w3_can/sont.html</a>
Texas	<a href="http://www.quadralay.com/www/TexasInfoServers.html">http://www.quadralay.com/www/TexasInfoServers.html</a>
Utah	<a href="http://wings.buffalo.edu/world/utahwww.html">http://wings.buffalo.edu/world/utahwww.html</a>
World (from CERN)	<a href="http://info.cern.ch/hypertext/DataSources/WWW/Servers.html">http://info.cern.ch/hypertext/DataSources/WWW/Servers.html</a>
World (from Colorado)	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/14/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/14/summary.html</a>

## New Web Sites

The WWW guru keeps abreast of the new pages that are popping up around the Web. The National Center for Supercomputer Applications (NCSA) provides a page that contains announcements of new pages. The page is called "What's New," and its URL is

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/whats-new.html>

Announcements of new sites can be sent to [whats-new@ncsa.uiuc.edu](mailto:whats-new@ncsa.uiuc.edu).

However, many interesting new pages never make it into the NCSA list. Therefore, the guru who needs to know about more new sites and has time on his hands can always do additional exploring on his own.

Many sites keep their own "What's New" page, also. This kind of page, however, usually covers changes on only that one site instead of the whole Web. It is still useful, though, if you use that particular site frequently.

## Indexes

All of the schemes we have been talking about are known as *browsal schemes*. You look at a page of choices and select some. However at times, it is even more useful to be able to search on a word and find pages that have that word in their title or in the page.

These indexes are generated in several different ways. Some use a person's global history file. Others are built from the Hotlists of a large number of people. With some, the users enter their own pages into the index. Some others use a Spider or Worm, which will be discussed later in this chapter, to capture the information. A handful of the available indexes can be found in Table 18.5.

**Table WW.5. Indexes.**

<i>Name</i>	<i>URL</i>
ALIWEB	<a href="http://web.nexor.co.uk/aliwebsimple">http://web.nexor.co.uk/aliwebsimple</a>
COMMA Hotlist DB	<a href="http://www.cm.cf.ac.uk/htbin/AndrewW/Hotlist/hot_list_search.csh">http://www.cm.cf.ac.uk/htbin/AndrewW/Hotlist/hot_list_search.csh</a>
EINet Galaxy Search	<a href="http://galaxy.einet.net/www/www.html">http://galaxy.einet.net/www/www.html</a>
Infobot Hotlist	<a href="ftp://ftp.netcom.com/pub/ksedgwic/hotlist/hotlist.html">ftp://ftp.netcom.com/pub/ksedgwic/hotlist/hotlist.html</a>
Joe's Global History	<a href="http://www.galcit.caltech.edu/~joe/cgi-bin/hstsrch-joe">http://www.galcit.caltech.edu/~joe/cgi-bin/hstsrch-joe</a>
Jumpstation	<a href="http://www.stir.ac.uk/jsbin/js">http://www.stir.ac.uk/jsbin/js</a>
Nomad	<a href="http://www.rns.com/www_index/intro.html">http://www.rns.com/www_index/intro.html</a>

NorthStar	<a href="http://comics.scs.unr.edu:7000/top.html">http://comics.scs.unr.edu:7000/top.html</a>
RBSE's URL Database	<a href="http://rbse.jsc.nasa.gov/eichmann/urlsearch.html">http://rbse.jsc.nasa.gov/eichmann/urlsearch.html</a>
SIMON	<a href="http://web.elec.qmw.ac.uk:12121/server/search.html">http://web.elec.qmw.ac.uk:12121/server/search.html</a>
SURANetGuide-All	wais://nic.sura.net:210/SURANetGuide-All
W3 Search Engines	<a href="http://cui_www.unige.ch/meta-index.html">http://cui_www.unige.ch/meta-index.html</a>
WebCrawler	<a href="http://www.biotech.washington.edu/WebCrawler/WebQuery.html">http://www.biotech.washington.edu/WebCrawler/WebQuery.html</a>

## Your Own Home Page

Most WWW gurus have their own Home Page that describes themselves and has links off to items they find interesting in the Web. Gurus, therefore, must either learn HTML or find a program that can convert from their favorite format to HTML. A list of available HTML converter programs and editors can be found at the URL

<http://info.cern.ch/hypertext/WWW/Tools/Filters.html>



**Note:** BBEDIT (shareware) for the Macintosh also supports HTML extensions and is easy to use to create HTML documents. The shareware version is available on world.std.com in /ftp/pub/bbedit as well as on sumex-aim.stanford.edu in /info-mac/TextProcessing.

Even if a converter can be found, a guru may wish to learn HTML. An easy to learn introduction to HTML can be found in a document called "Beginner's Guide to HTML," which is available as

<http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>

The official HTML specification can be found at

<http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html>

Finally, there is a HTML Developer's Page that can be accessed at

<http://oneworld.wa.com/htmldev/devpage/dev-page.html>

After learning to use one of the HTML converters or learning HTML, creating your own home page should be fairly easy to do. Before designing your own page, however, you might want to look at the home pages of some other people on the Internet to get an idea of what some look like (see Table 18.6).

**Table 18.6. Example home pages.**

<b>Name</b>	<b>URL of Home Pages</b>
Aurelius Prochazka	<a href="http://www.galcit.caltech.edu/~joe/coolpage.html">http://www.galcit.caltech.edu/~joe/coolpage.html</a>
Billy Barron	<a href="http://www.utdallas.edu/acc/billy.html">http://www.utdallas.edu/acc/billy.html</a>
Brandon Plewe	<a href="http://wings.buffalo.edu/staff/plewe/home.html">http://wings.buffalo.edu/staff/plewe/home.html</a>
CMU CS students	<a href="http://www.cs.cmu.edu:8001/web/personal.homes.html">http://www.cs.cmu.edu:8001/web/personal.homes.html</a>
Dave Brennan	<a href="http://www-external.hal.com/~brennan/index.html">http://www-external.hal.com/~brennan/index.html</a>
Eriq Neale	<a href="http://lipsmac.acs.unt.edu/">http://lipsmac.acs.unt.edu/</a>
Kevin Hughes	<a href="http://www.eit.com/people/kev.html">http://www.eit.com/people/kev.html</a>

Meng Weng Wong	<a href="http://www.seas.upenn.edu/~mengwong/meng.html">http://www.seas.upenn.edu/~mengwong/meng.html</a>
Rob Hartill	<a href="http://www.cm.cf.ac.uk/People/Robert_Hartill.html">http://www.cm.cf.ac.uk/People/Robert_Hartill.html</a>
Tim Berners-Lee	<a href="http://info.cern.ch/hypertext/WWW/People/Berners-Lee-Bio.html">http://info.cern.ch/hypertext/WWW/People/Berners-Lee-Bio.html</a>
Various	<a href="http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/58/summary.html">http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/58/summary.html</a>

After looking at some of these home pages, you will hopefully notice the different design philosophies that people use. Some people include their pictures while others do not (likely because they do not want people to know what they look like or they do not have access to a scanner). Some people are very serious and include resumes, current projects, and other such things while others have made their page totally for fun. Others still, such as myself, combine the serious and the fun together.

## Server Software

Many different Web server packages exist as listed in Table 18.7. The guru needs to look at the features of each package and then decide which is the best for him or her.

In fact, the guru may end up deciding that a server package is unnecessary because other alternatives exist. It is possible to serve up HTML documents from other types of Internet servers, such as anonymous FTP and Gopher. However, by doing this you lose some of the more advanced features that the server provides, such as image maps and CGI scripts, both of which will be discussed later in this chapter. Also, using FTP is less efficient, slower, and limits your security options in some cases.

If a WWW server package is needed, you will need to first pick a platform. It will generally be in your best interest to choose UNIX if possible, unless you do not have the ability to use it or there is an overriding reason why another platform suits you better. The UNIX servers are the most popular, almost always the most current, and tend to have good performance.

**Table 18.7. Web Server Software.**

<i>Software</i>	<i>Platform</i>	<i>URL</i>
CERN HTTPD	UNIX	<a href="ftp://info.cern.ch/">ftp://info.cern.ch/</a>
CERN HTTPD	VMS	<a href="ftp://info.cern.ch/pub/www/src/www2161betavms.tar_Z">ftp://info.cern.ch/pub/www/src/www2161betavms.tar_Z</a>
GN	UNIX	<a href="ftp://hopf.math.nwu.edu:70/9/gn2.11.tar.gz">ftp://hopf.math.nwu.edu:70/9/gn2.11.tar.gz</a>
HTTPS	Windows NT	<a href="ftp://emwac.ed.ac.uk/pub/https">ftp://emwac.ed.ac.uk/pub/https</a>
MacHTTP	Mac	<a href="ftp://oac.hsc.uth.tmc.edu/public/mac/MacHTTP/machttp_beta.sit.hqx">ftp://oac.hsc.uth.tmc.edu/public/mac/MacHTTP/machttp_beta.sit.hqx</a>
NCSA HTTPD	UNIX, VMS	<a href="ftp://ftp.ncsa.uiuc.edu/pub/web">ftp://ftp.ncsa.uiuc.edu/pub/web</a>
NCSA HTTPD	Windows	<a href="ftp://ftp.ncsa.uiuc.edu/Web/ncsa_httpd/contrib/winhttpd/whtp13p1.zip">ftp://ftp.ncsa.uiuc.edu/Web/ncsa_httpd/contrib/winhttpd/whtp13p1.zip</a>
Plexus	UNIX	<a href="ftp://austin.bsdi.com/plexus/2.2.1/dist/Plexus2.2.1.tar.Z">ftp://austin.bsdi.com/plexus/2.2.1/dist/Plexus2.2.1.tar.Z</a>
Region 6	VMS	<a href="http://kcg11.eng.ohio-state.edu/www/http_server_13.tar">http://kcg11.eng.ohio-state.edu/www/http_server_13.tar</a>
SerWeb	Windows	<a href="ftp://winftp.cica.indiana.edu/pub/pc/win3/winsock/serweb03.zip">ftp://winftp.cica.indiana.edu/pub/pc/win3/winsock/serweb03.zip</a>
SerWeb	Windows NT	<a href="ftp://emwac.ed.ac.uk/pub/serweb/serweb_i.zip">ftp://emwac.ed.ac.uk/pub/serweb/serweb_i.zip</a>
WEB4HAM	Windows	<a href="ftp://ftp.informatik.uni-hamburg.de/pub/net/winsock/web4ham.zip">ftp://ftp.informatik.uni-hamburg.de/pub/net/winsock/web4ham.zip</a>

Also, many gateways available for the various Web servers enable you to tie into other software or databases. These

gateways are documented on the page

<http://info.cern.ch/hypertext/WWW/Daemon/Gateways.html>

In addition, you can write your own gateways using CGI (Common Gateway Interface), which we will discuss in a little bit.

There is always the choice of writing your own server. This should only be done when you have a particular reason for doing so. Information on writing servers can be found on the page

<http://info.cern.ch/hypertext/WWW/Daemon/Overview.html>

After your server is installed and it is getting usage, you may want to see how much and what kind of usage it is getting. First, you must turn on the logging function of your server if it has one. After that, you can either write your own programs for generating statistics or acquire a Web statistics package. Several packages are available and listed in Table WW.8.

**Table 18.8. WWW server statistics packages.**

<i>Software</i>	<i>URL</i>
getstats	<a href="ftp://ftp.eit.com/pub/web.software/getstats/">ftp://ftp.eit.com/pub/web.software/getstats/</a>
wusage	<a href="http://siva.cshl.org/wusage.html">http://siva.cshl.org/wusage.html</a>
wwwstat	<a href="ftp://liege.ics.uci.edu/pub/arcadia/wwwstat/">ftp://liege.ics.uci.edu/pub/arcadia/wwwstat/</a>

## Page Design Issues

A well-known problem with hypertext systems, such as WWW, is that most people do not know how to write good hypertext documents. In general, the worse hypertext documents are those where the author uses graphics and different types of links just to show off his/her ability in using the technology.

While I will be the first to admit that I am not an expert at writing good hypertext documents, I have learned over time how to avoid some bad techniques. In addition, I have picked up some good techniques to use. I will show you them here because they should be known to gurus of WWW. Other suggestions can be found in the pages

<http://info.cern.ch/hypertext/WWW/Provider/Style/Overview.html>

and

<http://www.willamette.edu/html-composition/strict-html.html>

## Lists versus Full-Text

Hopefully, when you were learning about HTML by reading "The Beginner's Guide to HTML," you noticed that HTML enables you to make lists of links as well as place links in the body of a paragraph. Both techniques are useful, but both can also be misused. Care must be taken to use them appropriately.

While WWW is much more flexible and powerful than a paper book, many of the good book writing techniques apply equally as well to hypertext as they do to books. First of all, books typically have a table of contents. A good collection of related HTML documents in the Web should also have the equivalent to a table of contents. Otherwise, it is difficult for the reader to grasp the structure of the document collection effectively. A table of contents can be easily developed using a list of links.

However, once you get past the overview level(s), such as the table of contents, and on to the actual guts of the

material, adjustments need to be made. Lists of links should generally be abandoned to define structure. Instead, paragraphs with links at appropriate places is much more effective in most cases.

## Number of Links

A common mistake by new HTML authors is to use too many links or not enough. In theory, it is possible to define every part of a document as being a link to other documents. However, the user of this document would find it useless and frustrating.

In a good hypertext document, the links only take the user to important associated topics and not to irrelevant ones. To put this another way, by specifying a link, the author is telling the reader what the important related topics are. By not specifying a link, the author is saying that the topic is not that important or that no additional information exists on that particular topic.

On the other end of the spectrum, if the document has too few links, the author limits the ability of the reader to find related information. In fact, if all documents are like this, there is no point in using a hypertext system such as the Web at all, because its major strength is being ignored. In this case, Gopher is probably a superior system to use because it is easier to implement.

There are no hard and fast rules as far as the number of links per paragraph. It is dependent on the content of the individual document. The best way to learn what is an appropriate number is to browse around the Web and find some good and bad examples to learn from. You will know when you are looking at a document with too few, too many, or just the right number of links.

## Links Back

Most good hypertext pages have a link back to their "official" parent page at the end of the page. The reason is that readers may get to the page from any other page in the Web or from an index. Once there however, the reader may be interested in other documents in the same collection. If no link is provided, it is difficult to see the rest of the collection.

One example is that a link takes you to the page for the printer products by company XYZ. However, then you want to see other information about XYZ. It is very easy to do if a link taking you back to the parent page or XYZ's home page is available. Without the link, it requires some searching on the guru's part to find the parent page. For the novice, it is nearly impossible.

## Inline Graphics

In a perfect world, readers would not take a noticeable performance hit when viewing HTML documents with inline graphics. However, at the current time, almost all readers notice a significant slowdown due to online graphics. Even for those with high speed connections (T1 or higher) to the Internet and fast workstations, the performance problems show up from time to time.

There are two cases where the performance problems are very bad. The first is a user who is using a SLIP/PPP connection to the Internet. The second is when a slow link exists between the user and the WWW server. One frequent case of this is when the reader and the server are on opposite sides of the Atlantic Ocean. In these cases, hopefully, the user has set up his client to not automatically download images. In addition, some client programs such as Lynx do not display the images because they are geared for ASCII terminals and, therefore, will be faster.

The author of an HTML document needs to take these problems into account. First, the author needs to decide whether his/her audience has fast network links and a reasonably fast computer. If they do, using inline graphics will make the page prettier and, if used appropriately, are easier to understand. On the other hand, if the users of a page have slow links, then inline graphics should be kept to a bare minimum. Otherwise, the response time will make the

pages unusable. A frequently used and good technique is to shrink the images and include them as small inline images. Then allow the user to click on the inline images to get the full image.

For most of us though, we will have some users in both categories. At the present time, in most cases, it is better to err on the side of too few graphics than too many. Over the course of the next few years, this balance will change as computers and networks get faster so that it will be better on average to include more graphics.

A related note is that some people are using ASCII based clients or have the inline images turned off in their client. Therefore, overdependence on unnecessary graphics will disenfranchise these users. However, at these times though, graphics are critical to the presentation of certain material. When this occurs, by all means, use graphics.

## **Client Independence**

The Web and HTML in particular are built around the idea of client independence and being format-free. If you look at HTML, you will find no markers to specify exact font sizes. Also, you will find no way to hardcode locations on the screen in the language. This was intentional in the design of the WWW system.

Users of the system will be on various machines. Some will be able to handle advanced graphics, while others will not be able to handle graphics at all. Some will have very large windows for their WWW clients while others will be limited to 24 rows and 80 columns. Therefore, any attempt on the part of the server or the author to specify more than the most general formatting options, such as bolding and a relative heading size, will run into difficulties.

It is a common mistake on the part of the novice HTML author to try to make his page look nice on a particular client package, usually Mosaic. However, this time is basically wasted because any of the work that goes into this will be useless on any other client. In fact, it is often the case that the more effort spent in customizing for the screen for one client the worse it will look on other clients.

Also, HTML authors need to be careful about checking to see what advanced features his audience's client package supports. For example, while forms are a very useful feature of the Web, many clients have yet to support it. Also, the image map functionality, which will be explained later in this chapter, only works on graphical clients and by definition cannot work on ASCII based client packages.

Finally, it is very important that the author consider what file formats his readers can process when selecting the file format of graphics, sound, and video. Almost all WWW client packages will call external programs to process this kind of data. However, it is always an unknown whether or not the reader has installed these external programs or not. For inline graphics, the client packages can only handle very few types of graphics files, such as GIF.

## **CGI Scripts**

Many WWW servers support the ability of the server to call scripts. These scripts allow the Web guru to design custom Web functionality. For those familiar with Gopher and the go4gw system, CGI (Common Gateway Interface) serves similar purposes. Sometimes you may see references to another script system known as HTBIN. HTBIN is an older system than has been superseded by CGI. You should write any new scripts for the CGI interface specification and not HTBIN. Also, any HTBIN script should be upgraded to use CGI.

The method of implementing the CGI functionality is somewhat dependent on your server package. However, the programming interface that a CGI script uses remains the same with any server package that supports CGI. Therefore, once you have written a CGI script, it will work with any server that supports CGI—but you may have to spend some time figuring out how to best tie it into the server.

Writing a CGI script requires understanding something about the HTTP protocol. Therefore, you should read up on that first if you haven't already. After that, take a look at the page <http://hoohoo.ncsa.uiuc.edu/cgi/>, which describes CGI in detail.

## Forms

Many of the newer Web clients support forms. Forms are a wonderful addition to the Web because they allow the user to give information back to the server besides just simple mouse clicks. Forms support text fields, password fields (the text does not show up on the screen), checkboxes, radio buttons, menus, reset buttons, and submit buttons. They are quite powerful and can replace almost any paper form. Unfortunately, at the present time not all Web clients can support forms, so care must be taken when using forms for a particular application.

For help on using forms, a good page exists at

<http://nearnnet.gnn.com/forms/help/form-help.html>

A more formal description of HTML forms can be found in

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html>

## Image Maps

One of the more advanced design features that can be used is known as *image maps*. The more commonly used term, especially among novices, is a *clickable* map. It enables you to design Web screens so that users can use your Web page as a GUI interface. Instead of a link being a text phrase or a whole image, it can be a part of a graphic image. Therefore, it is possible to have maps with different countries, click on a country, and then get a page on that country. However, it is important to remember that this is advanced functionality that is not available in all client packages and, therefore, some people will be unable to use it.

A tutorial on using image maps can be found at

<http://wintermute.ncsa.uiuc.edu:8080/map-tutorial/image-maps.html>. This document only covers how to configure the maps for users of the NCSA HTTPD server. If you are using another type of server, you will have to look at the documentation of the server for additional instructions. Also, some examples are helpful in designing maps (see Table 18.9). In addition, many of the geographical organizational schemes use image maps and are good examples.

**Table 18.9. Examples of maps.**

<b>Name</b>	<b>URL</b>
Europe	<a href="http://s700.uminho.pt/europa.html">http://s700.uminho.pt/europa.html</a>
Honolulu Community College	<a href="http://www.hcc.hawaii.edu/hccmap/hccmap.html">http://www.hcc.hawaii.edu/hccmap/hccmap.html</a>
Icon Browser	<a href="http://www.di.unipi.it/iconbrowsers/icons.html">http://www.di.unipi.it/iconbrowsers/icons.html</a>
London Underground	<a href="http://web.cs.city.ac.uk/london/travel/underground/map?central">http://web.cs.city.ac.uk/london/travel/underground/map?central</a>
Museum of Paleontology	<a href="http://ucmp1.berkeley.edu/">http://ucmp1.berkeley.edu/</a>
Planet Earth Home Page	<a href="http://white.nosc.mil/pehp.html">http://white.nosc.mil/pehp.html</a>
SWISS2DPAGE Map Selection	<a href="http://expasy.hcuge.ch/htbin/map2">http://expasy.hcuge.ch/htbin/map2</a>
Syracuse University	<a href="http://www.syr.edu/">http://www.syr.edu/</a>
Univ Cal Berkeley	<a href="http://server.berkeley.edu/htbin/imagemap/campus">http://server.berkeley.edu/htbin/imagemap/campus</a>
World Map	<a href="http://wings.buffalo.edu/world">http://wings.buffalo.edu/world</a>

Also, a couple of different map editing programs exist that can help speed the development of image maps along. mapedit is an X based editor from image maps that can be found at

<http://sunsite.unc.edu/boutell/mapedit/mapedit.html>

For the Macintosh, a product known as HyperMap is available. It can be found at

<http://libertynet.upenn.edu/~gasser/projects/hypermap.html>

## Client Configuration

The more powerful WWW clients, such as Mosaic, contain many client configuration options. A guru will know the options available with his/her package. If the package does not support the particular option needed, the guru can locate a separate utility that will accomplish the function if necessary.

## Performance

The Web can be slow at times. There are several ways that the guru can maximize his/her performance by using a few tricks.

First of all, if he/she is coming into the Internet via a slow speed modem connection, it is best if automatic inline imaging download is disabled (in Mosaic, this is the turn off image download option) except when absolutely necessary. The guru also disables this option when downloading a page from a server that is known to be slow or one that is far away (usually defined as across an ocean). The guru may sometimes disable this option when downloading a page with a large amount of inline graphics. Finally, the guru may just use this option anytime the graphics are unneeded and speed is of the essence.

Most Web servers are typically least busy between midnight and 6 a.m. using the server's local time. Also, the busy transatlantic links tend to be least used between about 7 p.m. and 4 a.m. Eastern Standard Time (EST). Additionally, any time on a weekend, servers and links are less used. A WWW guru may use these times to access slow servers or servers across slow links so that the server responds faster, and his/her usage does not slow down other Internet users as much.

## Mirroring Pages

Another trick that WWW gurus often use to improve reliability and performance is to mirror HTML pages locally. In other words, the guru uses a particular Web page or set of pages frequently. Instead of repeatedly connecting across the Internet to access the page, the guru will copy (mirror) the page(s) to his/her local Web server. Then the guru can use this copy of the page(s) and receive faster response. Also, this can help lessen the load on a busy remote Web server (like CERN or NCSA) and overloaded network links.

An important part of the mirror process is figuring out how often the mirrored page should be refreshed. For very volatile pages, once a day may be good, though the mirror itself should occur at night if possible. For pages that are infrequently updated, getting a new copy once a week is acceptable. It just depends on the situation. In general, though, I would say that the refresh time should be between one day and one month.

Several different packages and methods for mirroring are available. There are basically three different methods of mirroring. The most straight forward is mirroring a single page exactly like it is. Next, a page can be mirrored, but all of the relative URLs in it are converted to be absolute URLs. The benefit of this method is that the links on the page will continue to work in all cases, whereas sometimes with the straight forward copy, they do. Finally, a page can be mirrored recursively. In this method, when a page is mirrored, it is parsed to see what pages it references in the same directory structure. These pages are then also downloaded and parsed. This continues until no more pages in the directory structure can be found.

Three popular mirroring tools are w3get, htget, and the Web client Lynx. Lynx only supports straight forward mirroring. This function can be accessed by typing `lynx -source URL > output-file`. w3get performs only recursive mirroring. htget is a later version of a script originally known as hget. htget is the most powerful of the three tools mentioned here and supports all three types of mirroring. It can be found at

[http://cui\\_www.unige.ch/ftp/PUBLIC/oscar/scripts/README.html](http://cui_www.unige.ch/ftp/PUBLIC/oscar/scripts/README.html)

The htget syntax is very simple. For straightforward mirroring, just type htget URL. For converting to absolute format, it is htbin -abs URL. Finally, for recursive copying, use the command htbin -r URL.

Another alternative to mirroring is a caching Web server. In the cache system, the client first checks to see if the cache server has the document. If so, the cache server gives the document to the client at faster than normal speed because the cache server should be local. If not, the cache server will download the document to keep a copy for caching purposes.

An experimental caching server has been developed. At the time of this writing, it had not been publicly released. Hopefully, it will be in the near future. To acquire more information, look at the page

<http://www.hensa.ac.uk/hensa.unix.html>

In addition to mirroring, it is often a good idea to occasionally check to see if the links in your pages are still good. A package called "checkweb" is available for doing this. Unfortunately, it does not have a URL on the network from which you can download it. Instead, ask around or try to find a copy of the February 17, 1994 posting of it on the newsgroup comp.infosystems.www.

However, you should probably test suspected dead links more than once. It is often the case on the Web that a link will be dead for minutes or hours and then reappear. This is often due to network connectivity problems or the WWW server crashing.

## Related Viewers

Most Web clients by default can render a small number of file types by themselves. For example, they all know how to render HTML. Most can render Gopher menus and FTP directories. Many can view small inline graphics. However, it is commonly known wisdom in the computer programming arena that no one package can do everything well. As it is, many people complain that Mosaic and some other Web clients do too much already, and you can see warts if you look closely for them. Therefore, all Web clients, including Mosaic, draw a line on what is directly included and what must be supported externally.

To deal with other types, most Web clients have the ability to call other utilities, often called *viewers*, to render types that the client itself cannot deal with. Therefore, with most clients, it is important for the guru to install a good set of viewers (see Table 18.10). For starters, viewers for graphic files, sound, and video are needed in addition to the client.

**Table 18.10. Associated Viewers.**

<i>Viewers</i>	<i>URL</i>
Quicktime for Unix	<a href="ftp://public.x.org/R5contrib/xanim264.tar.Z">ftp://public.x.org/R5contrib/xanim264.tar.Z</a>
Various Unix	<a href="ftp://ftp.ncsa.uiuc.edu/Web/Mosaic-viewers">ftp://ftp.ncsa.uiuc.edu/Web/Mosaic-viewers</a>
Various Windows	<a href="ftp://ftp.ncsa.uiuc.edu/PC/Mosaic/viewers">ftp://ftp.ncsa.uiuc.edu/PC/Mosaic/viewers</a>

## Downloading Images/Icons

On occasion, the guru may see a picture, sound clip, movie, or icon that is particularly interesting. He/she might decide to make a local copy of this file. There are several ways to accomplish this, but first the guru must tackle the ethical issues. Often, these files are of copyrighted material. Remember that in many countries, including the U.S., a work is copyrighted until the author declares it is public domain. Even if the work is copyrighted, the author will

often allow use of the work without charge if nicely asked.

If the guru is making a copy for his/her own personal use, it is usually safe in the U.S. at least to make a copy of the work under the Fair Use Doctrine. If the guru wants to use the file on his/her Web page or in another work, the guru needs to contact the author for permission. Unfortunately, most of these image files in the Web have not been accredited to an author. Therefore, it is often difficult to gain proper permission.

However, fortunately a few Icon libraries are available on the Web (see Table 18.11). The images available in these libraries should all be public domain and can probably be used in the design of your Web pages.

**Table 18.11. Available icon libraries.**

<i>Name</i>	<i>URL</i>
Anthony's Icon Library	<a href="http://www.cit.gu.edu.au/~anthony/icons/">http://www.cit.gu.edu.au/~anthony/icons/</a>
General Icons	<a href="http://www.ncsa.uiuc.edu/General/Icons/">http://www.ncsa.uiuc.edu/General/Icons/</a>
Gopher Icons	<a href="http://eitech.com/web/gopher.icons/gopher.html">http://eitech.com/web/gopher.icons/gopher.html</a>
Icon Browser	<a href="http://www.di.unipi.it/iconbrowsers/icons.html">http://www.di.unipi.it/iconbrowsers/icons.html</a>
Icon Leiste	<a href="http://www.cosy.sbg.ac.at/www-doku/iconbar.html">http://www.cosy.sbg.ac.at/www-doku/iconbar.html</a>
WWW Icons	<a href="http://bsdi.com/icons/">http://bsdi.com/icons/</a>

Downloading images and icons can be tricky. For images that are not inline, usually the easiest way is to tell your external viewer to save the image when it is displaying the image. However, not all external viewers are capable of doing this. When this fails, you can try to find an option in your client called something like "load to local disk." Enable it and then try to view the image. A dialog box asking you for a file name to save it under should appear. If your client does not have this option, look at using the software discussed in the mirroring section of this chapter to save the images.

Inline images are a little more difficult to deal with. The first way is to use the "load to local disk" option and then reload the page you are on. Alternatively, you can use the "view source" option in your client to find the URL of the inline image, and then use an external viewer as discussed above. If your external viewer cannot save, still use the "view source" option and then use mirroring software to pull down the inline image.

## Spiders

*Spiders*, sometimes also known as *robots*, *knowbots*, or *worms*, are programs that transverse the Web in search of something. Some build indices of the Web. Others measure the growth of the Web, measure throughput and latency in the net, and do textual analysis. Spiders hold the possibility of doing all kinds of other useful analysis of the Web. However, they can also be a large resource drain on the network if not used properly.

Programming a Spider is a task only for a guru of the Web. Programming one requires knowledge of HTML and the HTTP protocol. Without accurate knowledge of both and careful debugging, many Spiders will put a strain on the Web, due to an infinite loop or downloading too many documents too quickly. Fortunately, Martijn Koster took on the task of defining what a good Spider should and should not do. This definition can be found at <http://www.nexor.co.uk/mak/doc/robots/robots.html>. Any new Spider needs to follow these rules to be Web-friendly.

If you as a guru are thinking about writing a Spider, please look at the other Spiders first to make sure nobody else is doing the same thing. The Web is huge, and the fewer unnecessary Spiders running the better.

If you are a server administrator and do not want Spiders transversing your Web server, a method for telling a Spider to leave you alone exists and is documented on

<http://www.nexor.co.uk/mak/doc/robots/norobots.html>

It is up to the individual robot to support this function. Some do and some do not. If it does not, try to find the Spider author and ask him to stop running his Spider against your server. Finally, if absolutely necessary, you can tell your HTTP daemon to not allow connections from the Spider site.

## Keeping Current with Future Development

It is critical that the guru remember that the Web is in its infancy, and that to stay a guru requires a significant investment in time to keep up with all the new developments. Much discussion is underway about important topics like a new version of HTML called HTML+.

HTML+ is documented in

<ftp://ds.internic.net/internet-drafts/draft-raggett-www-html00.txt>

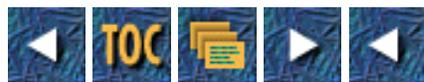
HTML+ will be a superset of HTML so that all current HTML documents will continue to be valid. HTML+ will probably have better table support and add in a few formatting features, such as right-justification. These discussions will lead to major changes in the WWW system over the next few years.

Many exciting rumors about the future of Web clients and servers abound. As I was writing, an enhanced commercial version of Mosaic was released. From various commercial client vendors, there is talk of adding security features such as authentication and encryption. Others are talking about more business oriented versions of the Web.

The Web is discussed on the USENET newsgroups comp.infosystems.www.misc, comp.infosystems.www.providers, and comp.infosystems.www.users. In addition, monitoring the group comp.infosystems.announce is useful because announcements of new software and sites are posted regularly. These sites include Web sites as well as sites using other protocols such as Gopher and WAIS.

The Web also contains many resources for WWW developers that were not covered in this chapter. A good starting point for finding out about these resources is a system known as CyberWeb. The CyberWeb can be accessed at

<http://nyx10.cs.du.edu:8001/~arichmon/Web.html>





- [Part III](#)
  - [A Guru's Diversions](#)

## Part III

### A Guru's Diversions





- [11](#)

- [Online Entertainment for the Internet Guru](#)

- [How to Program Muds](#)
- [Chapter Overview](#)
- [Notation](#)
- [Overview of Common Mud Commands](#)
  - [Connecting to the Mud](#)
- [You can connect to the mud using telnet:](#)
  - [Connecting to a Character](#)
  - [Obtaining Help](#)
- [This may lead to more specific help topics. You can type](#)
  - [Talking](#)
  - [Acting out Actions](#)
  - [Looking at Objects](#)
  - [Moving Around](#)
  - [Whispering to Other Players](#)
  - [Paging other Players](#)
  - [Picking up Objects](#)
  - [Dropping Objects](#)
  - [Listing What You Are Carrying](#)
  - [Finding out Who Else Is Playing the Mud](#)
  - [Returning to Your Home](#)
  - [Quitting the Mud](#)
- [How to Program MUCKs, MUSHes, and MUSEs: A Tutorial](#)
  - [Objects](#)
  - [DBREF](#)
  - [Money](#)
  - [Wizards and Officials](#)
  - [At-Commands](#)
  - [Flags](#)
  - [Messages](#)
  - [Creating and Naming Objects](#)
  - [Attributes](#)
  - [Action Lists](#)
  - [Examining](#)
  - [More Movement Commands](#)
  - [Killing](#)
  - [Giving and Receiving Objects](#)

- [Finding and Searching for Objects](#)
- [Rooms and Exits](#)
- [Rooms as Homes](#)
- [Examining Exits](#)
- [Exit Attributes](#)
- [Special Exits](#)
- [Other Flags on Rooms and Exits](#)
- [Unlinking Exits](#)
- [Containers](#)
- [Enter and Leave Attributes](#)
- [Pronoun Substitutions](#)
- [Try It Out!](#)
- [Programming a MUCK](#)
  - [Popular MUCKs](#)
- [Things Specific to MUCKs](#)
  - [MUF Programming](#)
  - [Using the MUF Editor and Compiler](#)
  - [MUF Basics](#)
  - [Basic MUF Functions](#)
  - [Variables](#)
  - [Messages](#)
  - [Conditional Statements](#)
  - [An Example MUF Program](#)
- [Programming a MUSH](#)
  - [Popular MUSHes](#)
- [Things Specific to MUSHes](#)
  - [Possessive Pose](#)
  - [Puppets](#)
  - [Listening and Responding to Messages](#)
  - [Percent Substitutions](#)
  - [Registers](#)
  - [Triggers](#)
  - [Numbered Variables](#)
  - [Braces](#)
  - [Using an Object](#)
  - [Functions](#)
  - [The v\(\) Function](#)
  - [The s\(\) Function](#)
  - [The get\(\) Function](#)
  - [The rand\(\) Function](#)

- [The Switch Command](#)
- [User-Defined Commands](#)
- [User-Defined Attributes](#)
- [The Queue](#)
- [The Wait Command](#)
- [The Halt Command](#)
- [Semaphores](#)
- [Putting It All Together: Exploding Jawbreaker Example](#)
- [Programming a MUSE](#)
  - [Popular MUSEs](#)
  - [MuseNet Worlds](#)
- [Things Specific to MUSEs](#)
  - [Commands](#)
  - [Classes](#)
  - [Zones](#)
  - [Interactive Narration: "Monster Island" Example](#)
- [After a few seconds, we see...](#)
- [MUCK, MUSH and MUSE Reference](#)
  - [Concept Reference](#)
  - [Control](#)
  - [Costs](#)
  - [Credits](#)
  - [Drop-To](#)
  - [Enactor](#)
  - [\\$ and ! Events](#)
  - [Exits](#)
  - [Failure](#)
  - [Home](#)
  - [Linking](#)
  - [me](#)
  - [Names](#)
  - [NULL Exits: Bogus Commands](#)
  - [Numbers](#)
  - [Objects](#)
  - [Pronouns](#)
  - [Stack](#)
  - [Substitutions](#)
  - [Success](#)
  - [V-Registers](#)
  - [Verb Commands](#)

- [Tips For Programmers](#)
- [At-Commands Reference](#)
  - [At-Commands](#)
- [Sets the number of columns that WHO prints. The default is 80.](#)
  - [Flag Reference](#)
  - [Flags](#)
- [Attribute Options Reference](#)
  - [Attribute Options](#)
  - [MUF Function Reference](#)
  - [Notation](#)
  - [Parameter Types](#)
- [MUF Functions](#)
  - [Arithmetic Functions](#)
  - [Comparison Functions](#)
- [MUSH and MUSE Function Reference](#)
  - [Functions](#)
- [Returns the logarithm base 10 of the number.](#)
  - [Lock Reference](#)
  - [Locks](#)
  - [Object Design Worksheet](#)
- [How To Program a MOO](#)
  - [Popular MOOs](#)
  - [Basic Concepts](#)
    - [Basic Values](#)
    - [Object Numbers](#)
    - [Inheritance](#)
    - [Verbs](#)
    - [Properties](#)
    - [Messages](#)
    - [Inspecting Objects](#)
  - [Basic Commands](#)
  - [At-Command Equivalencies](#)
  - [MOO Programming](#)
    - [Expressions](#)
    - [Statements](#)
    - [Entering Verbs](#)
- [MOO Reference Manual](#)
  - [At-Command Reference](#)
  - [Message Reference](#)
    - [Messages](#)

- [Error Reference](#)
- [Built-In Function Reference](#)
- [Summary](#)

# 11

## Online Entertainment for the Internet Guru

--by Joseph R. Poirier and [Tod Foley](#)

### How to Program Muds

Several years ago I joined the Information Superhighway and took the on ramp marked MUD, or Multi-User Domain.... What makes these games so enticing is that they are a step up from chat and e-mail. Not only is it possible to assume whatever personality you want, but you can act any way you want. You can kill without ever seeing blood, roar threats or cross swords without ever breaking into a sweat.... With MUDs and other on-line multiplayer games, you can live vicariously in a new universe and even attain godhood, thereby achieving omnipotence.

—Kristina Harris, "Godhood: Not All Fun and Games," "My Turn," *Newsweek*, July 18, 1994, p. 12.

One of the advantages of access to the Internet is the increased ability for people from all over the world to meet, and one of the major meeting places is *interactive multi-user virtual realities*. Also known as *multi-user dungeons*, or *MUDs*, these environments have expanded out of the game-playing niche into educational uses, corporate uses, and research. Some people call them multi-user domains in an attempt to de-emphasize the Dungeons and Dragons™ role-playing influences and highlight the multi-user side. Nowadays, the acronym MUD has been generalized to lowercase *mud*.

Muds are a cross between a text adventure game and a chat line. Muds contain a number of rooms, each having a description and possibly containing objects that players can use. Players of the mud—known as *mudders*—move from room to room around the mud. They can pick up objects and use them, meet and talk to other players, solve puzzles, play games, read mud newspapers, combat monsters, and explore virtual reality.

Muds are becoming more and more mainstream. Once the bastion of male computer science students who also liked a good role-playing game from time to time, they now attract all sorts of people from many walks of life. On a mud, you can bump into someone, male or female, who is a film director in real life, or perhaps an English professor, a high school teacher, a sociologist or an engineer—all in the same room! *Newsweek* magazine, for instance, has mentioned muds several times in its new Cyberscope Department, a page-long collection of short articles related to Cyberspace.

There are some graphical muds being developed, but the vast majority are text-based. Some muds emphasize social aspects; others emphasize combat. On social muds, players tend to use the mud as a communication medium. Players talk to other players, create and play games, and join group discussions. On combat muds, players buy weapons and armor, join guilds, learn skills, and fight monsters to gain experience. With enough experience, a player can become a wizard on the mud. Wizards have the ability to create new objects, rooms, and puzzles on combat muds.

There are several major types of muds, such as MUCKs, MUSHes, MOOs, MUSEs, LP-MUDs, Diku-MUDs, and others. Each type of mud differs in the underlying programming environment. Sometimes you will see the word *mud* expressed as MU\* in newsgroups and research papers. This is due to the various mud types that have those two letters in the beginning of their names.

Which type of mud you end up programming for (also known as *building*), often depends as much on happenstance as on deliberate decision. After all, one tends to hang out wherever one's friends hang out. But there are distinct differences from one variety of mud to the next, in both the structure of their programming languages and in the kind of players and builders they tend to attract. For what it's worth, just about any kind of virtual environment *could* be created in any of the mud languages,

and yet certain languages have developed genre- or function-based followings, and have evolved further than other, lesser-used types of muds.

## Chapter Overview

This chapter describes how to use and program several different types of muds: MUCKs, MUSHes, MUSEs and MOOs. LP-MUDs and Diku-MUDs, which are also very popular, are not discussed here. Many small examples will be presented throughout the chapter.

If you are totally new to muds, you might want to begin by reading through "Interactive Multi-User Realities: MUDs, MOOs, MUCKs and MUSHes," which can be found in *The Internet Unleashed*, an earlier SAMS book. That chapter describes mud basics in more detail. It describes several resources you can use on the Internet, such as newsgroups and FTP directories, to obtain more information on muds. It also details mud history and reflects on various social issues involving muds.

Here is a quick outline of mud expertise. It gives a few broad categories of mud programming knowledge so that you can see where your overall skills currently lie.

- Novice

- Connects to a mud for the first time.

- Uses a "guest" character.

- Beginner

- Knows how to connect to a mud.

- Has his or her own character.

- Knows common mud commands such as talking, acting, looking, and moving.

- Uses some type of mud client program.

- Intermediate

- Knows about at-commands.

- Knows about the various flags available on objects in a mud.

- Has tried building a few rooms on a mud.

- Has programmed a few objects to have basic descriptions and attributes.

- Has used functions and simple switch commands in object programs.

- Advanced

- Has programmed an object with user-defined commands and user-defined attributes.

- Understands the queue, semaphores, and time synchronization.

- Has programmed a reasonably complex object.

- Expert

Has built whole sections of a mud that contain interlocking puzzles and themes.

Has constructed very complex objects and commands, such as a mud e-mail system or a multi-player poker game.

Administers a mud.

This chapter deals with beginner, intermediate, and some advanced levels. First, a quick summary of basic mud commands is presented. Following this is a tutorial that talks about intermediate and advanced mud programming issues in a step-by-step manner. Many examples, throughout the tutorial, will help you understand the concepts presented. After the tutorial, there is a detailed reference of mud commands, attributes, functions, and flags.

## Notation

Throughout this chapter, commands and actions that you actually type will appear in monospaced type, and *italicized* type for placeholder terms for which you supply the appropriate item.

In addition, examples will use the > character as your prompt on the mud, to denote the commands you type as opposed to the messages the mud sends back to you. In those examples, you don't actually type the > character.

A | (vertical line) indicates two or more options exist. Choose one. Don't type the |.

Terms in [] (brackets) denote optional arguments. Don't type the brackets.

Finally, throughout this chapter, you will be using a character named Speedy.

## Overview of Common Mud Commands

Each type of mud has its own set of commands, and these commands differ from mud to mud. However, there are several common commands that you can expect no matter which type of mud you use. To refresh your memory, here is a quick summary of some basic commands you should know.

### Connecting to the Mud

#### You can connect to the mud using telnet:

```
telnet machine-name port-number
```

However, most mudders use some kind of mud client program, which interprets and displays mud messages to them in a nicely formatted fashion.

### Connecting to a Character

To connect to your character after you are on the mud, you type:

```
> connect character password
```

Some muds take care of password entry—or even the entire login sequence—for you. Just follow the prompts.

On most muds, you have to send e-mail to the mud administrator in order for him or her to create a character for you to play. Most muds provide for a "guest" character also, so that you can explore the mud a bit before deciding to establish a real presence there.

### Obtaining Help

To access the help facilities on your mud, use:

```
> help
```

## This may lead to more specific help topics. You can type

```
> help topic
```

to obtain help on just about everything in the mud, such as commands, attributes, functions, and various general topics.

### Talking

This command allows you to talk to other players, and other players to talk to you. The text you speak is transmitted, publicly, to everyone else in the room with you.

On most muds, the command to speak out loud to other players is the double-quote ("):

```
> "message
```

You do not need to put a double-quote at the end of your message. The mud puts that in for you.

```
> "Hey, what's for dinner?"
```

You say, "Hey, what's for dinner?"

### Acting out Actions

In addition to speaking, you can act out actions on the mud. These actions create a kind of narrative element to the mud environment, and allow you to express yourself in a different manner than merely speaking.

On many muds, the act command is the colon (:):

```
> :message
```

This command will append your message to your name and then display that to everyone in the room.

```
> :ran a marathon yesterday.
```

Speedy ran a marathon yesterday.

### Looking at Objects

The command to look at an object is:

```
> look object
```

When you perform this command, the object's description is shown to you. If you leave off the object portion, the mud assumes you want to see the description of the current room. The look command can be abbreviated to l.

```
> l book
```

You see a musty, leather-bound tome.

```
> l
```

Library

You are in the library of the house. You can see several bookshelves filled with dusty books lining the walls of the room.

### Moving Around

To move around from room to room on a mud, you use:

```
> go direction
```

To move to a specific place, use:

```
> go place
```

You can also move in a particular direction by typing that direction, such as:

```
> north
```

There may be several possible directions or places to move from each room. The best muds are very good at clearly denoting the exits from rooms in the mud.

## Whispering to Other Players

To send a message secretly to another player in the same room, use the whisper command. This command differs from mud to mud:

```
> whisper player = message (MUSH, MUCK, MUSE)
```

```
> whisper "message" to player (MOO)
```

This command has an abbreviation: w.

## Paging other Players

To send a secret message to another player not in the same room as you, use the page command. This command differs from mud to mud.

```
> page player = message (MUSH, MUCK, MUSE)
```

```
> page player "message" (MOO)
```

This command has an abbreviation: p.

On some muds, the page command costs virtual money, whereas the whisper command is free.

## Picking up Objects

The command to pick up an object is:

```
> get object
```

If the object can be taken, it is placed into your inventory.

On some muds, the command is also known as take.

## Dropping Objects

The command to drop an object is:

```
> drop object
```

The object will be removed from your inventory and placed in your current room.

On some muds, the command is also known as put.

## Listing What You Are Carrying

To see a list of the objects you are carrying, use:

```
> inventory
```

This can be abbreviated to i.

```
> i
```

You are carrying:

```
    a rose
```

```
    an envelope containing:
```

a letter

some chocolates

## Finding out Who Else Is Playing the Mud

The command to see who else is on the mud varies from mud to mud:

> WHO (MUSHes, MUCKs, MUSEs)

> @who (MOOs)

## Returning to Your Home

The command to return to your place of origin in the mud is:

> home

When you first begin playing a mud, your place of origin is usually a standard home room. Later, you can create your own room and reset your home room to it.

## Quitting the Mud

The command to quit playing varies from mud to mud:

> QUIT (MUSHes, MUCKs, MUSEs)

> @quit (MOOs)

## How to Program MUCKs, MUSHes, and MUSEs: A Tutorial

This section discusses most of the key topics you should be familiar with as a MUCK, MUSH or MUSE programmer. Most of the concepts apply to all of the MU\* environments; however, there are a few topics that only apply to certain types of muds. These topics will be labeled as such in the text.

MUCK is the most basic of the three types of muds that are discussed in this section. MUSH is more advanced than MUCK, and MUSE can be thought of as MUSH with a few extensions.

Thanks go to Lydia "Amberyl" Leong for the mud help document upon which this section is based.

## Objects

MUCKs, MUSHes and MUSEs have four basic types of things: *rooms*, *players*, *exits*, and *objects*. You can think of rooms, players, and exits as special objects that have added capabilities. Rooms are the building blocks of the mud. You move through the rooms in the mud in order to explore it and meet other people. Players are the people that you meet. You, of course, are a player, and there may be many more players, besides yourself, playing the mud at the same time. Exits connect rooms to each other. Finally, objects are everything else in the mud. You can create objects yourself and program them to behave in many ways.

## DBREF

Every object in the mud has a database reference number, known as its *DBREF*. This number can be used to access properties and attributes on the object.

Most commands will accept either names or DBREF as object specifiers, but some will insist on one or the other. Try it both ways if you get stuck.

## Money

Most muds have their own monetary system. As you wander around the mud, from room to room, you may find gold coins or

other monetary types. You can then use this money to create new objects, build rooms, and other such activities—since many of these activities require a small amount of money to be used.

Different muds have different types of money, such as gold coins, drachmas, cookies, pennies, dollars, or a fictional type like "blabooies." This chapter uses credits as the monetary unit in its examples.

## Wizards and Officials

Throughout this chapter, you may see the term *wizard* or *official*. Wizards are players in the mud who have special abilities. Frequently, the wizard is the person who is actually running the mud on their machine. However, there may be other wizards who help the mud administrator, with such things as character creation, policing the mud, and keeping the environment consistent. On MUSEs, wizards are called officials.

Some commands in the mud are only available to wizards. These commands will be identified as such where appropriate.

## At-Commands

Some commands and attributes in the mud start with an "at" symbol (@), such as the command to teleport from one room to another, @teleport. These commands and attributes are collectively known as *at-commands*.

## Flags

There are several *flags* that you can set in a MUSH. The syntax for setting a flag is:

```
> @set object = flag           (To turn the flag ON)
```

```
> @set object = ! flag        (To turn the flag OFF)
```

Throughout this section, different flags will be mentioned as appropriate.

## Messages

In addition to speaking and acting, you can write messages directly to other players using the emit command: @emit *message*. The message you type will be sent to everyone in the room, including you. As an abbreviation, you can type: \\ *message*. There are a few variants of this command. @pemit *player* = *message* will display the message only to the player specified. @oemit *player* = *message* displays the message to everyone in the room *except* the player specified.

An emitted message does not print who emitted it. If you want to always see who caused an emitted message, you can turn on your NOSPOOF flag. Upon doing so, you will be informed of the player or object that causes an emitted message.

Another flag is the HAVEN flag. When this is set on a player, the player will not receive pages or @pemit messages. If you try to page a player with HAVEN set, you will be told that that player is not accepting pages. This flag does not black out @emit messages, however.

To find out where a player is located, use the @whereis *player* command. This command will display the room where the player is. After the player is located, he or she is given a message that you are looking for him or her. If you do not want to be located, set yourself to DARK.

These commands are not available in MUCK.

## Creating and Naming Objects

To create a new object in the mud, type:

```
> @create object-name = cost
```

If you leave off the cost, the object will be created with the default cost, usually 10 credits. After checking to make sure you have enough money to create the object, the mud will respond with the message "*Object* created as object #*DBREF*" and the object will be placed into your inventory. You can now refer to this object by using either its object name or its DBREF. The MUSH performs pattern-matching, so you only need to type enough of the object name to differentiate it from other objects you may be holding.

If you change your mind and you want to rename an object, use:

```
> @name old-object-name = new-object-name
```

To change your own name, you can use

```
> @name me = new-name password
```

## Attributes

Now that you have created an object, you can set its description. The description of an object is one of the *attributes* that an object can have. An attribute is simply a property on an object.

There are several types of descriptions. `@desc object = description` sets the description that is displayed to the player that is looking at the object. `@odesc object = description` is displayed to everyone except the person looking at the object. The player's name and a space are prepended to the message specified. The command `@adesc object = actions` sets a list of actions that are executed by the mud when the object is looked at.

At this point, it is useful to take note of the basic forms that attributes take throughout MUSH. Assume there is an attribute with the name `@xxx`. It is applied to the player that is dealing with the object. In addition to the basic attribute, there are usually additional attributes that correspond to the basic attribute. `@oxxx` is applied to everyone *except* the player dealing with the object, and `@axxx` runs a list of actions when the object is used in some fashion. Familiarizing yourself with the `@xxx / @oxxx / @axxx` combination of attribute names will help you understand and remember other attribute names, since they are similar.

MUCKs, in general, don't have the `@axxx` attributes, and usually do not have as many `@xxx` and `@oxxx` attributes as MUSHes and MUSEs do.

When an object is picked up, its `@succ` attribute is triggered—"succ" is short for "success". To set the success attribute on an object, use `@succ object = message`. When the object is picked up, the message will be displayed to the player who took it. The `@osucc` attribute sets the message displayed to players other than the player who took the object, and the `@asucc` attribute can be used to set actions that you want performed when the object is taken.

Conversely, the `@fail` attribute is triggered when a player fails to pick up an object, and the `@ofail` and `@afail` attributes correspond appropriately. Usually, a player fails to pick up an object because an object is locked.

It is frequently a good idea to lock an object since unlocked objects can be stolen from you. To lock an object to yourself, use:

```
> @lock object = me
```

This will allow only you to pick up the object. In a more general form, the lock command is:

```
> @lock object = key
```

The key can be an object name, or a DBREF, or the special words "me" or "here." Boolean expressions are allowed. If you want the key to be a player, you must prefix the player name to an asterisk (\*), as in `@lock spinach = *Popeye`.

Locks are related to the `@succ` (success) and `@fail` (failure) attributes in that successfully passing a lock will trigger `@succ` and failing the lock test will trigger `@fail`.

To unlock an object, use `@unlock object`.

The `@drop` attributes set the messages that are displayed when an object is dropped. Using `@drop object = message` sets the message displayed to the player who drops the object. `@odrop` is the message displayed to others, and `@adrop` are the actions performed when the object is dropped.

The `@move` attributes are triggered when the player or object on which they are set moves from one room to another. The `@move`, `@omove` and `@amove` behave as you would expect.

To destroy an object that you own, use `@destroy object-name`. You will be paid whatever it cost you to create the object. Destroying objects when you don't need them anymore helps free up space in the mud database. If you want to give anyone the ability to destroy an object that you have created, turn on its DESTROY\_OK flag.

If you want other players to be able to see how you have programmed an object, set the object VISUAL.

```
> @create root beer float
```

```
root beer float created as object #1765.
```

```
> inventory
```

```
You are carrying:
```

```
root beer float (#1765).
```

```
You have 921 credits.
```

```
> @desc root = You see a frosted mug of root beer with a dollop of vanilla ice cream floating in it.
```

```
Set.
```

```
> @succ root = You pick up the frosted mug.
```

```
Set.
```

```
> @osucc root = picks up the mug of root beer.
```

```
Set.
```

```
> @drop root = You put down the frosted mug.
```

```
Set.
```

```
> @odrop root = puts down the mug of root beer.
```

```
Set.
```

```
> drop root beer float
```

```
You put down the frosted mug.
```

```
> look root beer float
```

```
root beer float (#1765)
```

```
You see a frosted mug of root beer with a dollop of vanilla ice cream floating in it.
```

```
> get root beer float
```

```
You pick up the frosted mug.
```

```
> @destroy root
```

```
You get back your 10 credit deposit for root beer float (#1765).
```

```
Destroyed.
```

On most muds, you can type enough of the object name to make it distinguishable from any other object in the room. Thus, in

the root beer float example, referring to it as root was sufficient.

## Action Lists

The @axxx attributes (attributes that start with the letter *a*) are used to execute action lists. An action list is simply a list of one or more actions, separated by semicolons. For example, you could create a fragile vase. When it is dropped, it does two things: it prints a message describing how it shatters, and then it destroys itself.

```
> @adrop vase = @emit The fragile vase shatters.;@destroy me
```

Set.

```
> drop vase
```

The fragile vase shatters.

fragile vase has left.

You get back your 10 credit deposit for fragile vase (#1238).

Actions lists are not available in MUCK.

## Examining

In addition to the basic look command, you can also examine items in the mud. Examining an object provides you with more detailed information about the object. The command to examine an object (or player) is:

```
> examine object
```

If you do *not* control the object, the examine command will tell you the owner of the object and show any public attributes on the object, such as its description, its success and fail attributes, and so forth.

If you *do* own the object, or if the object has its VISUAL flag set, you will be able to see all the attributes and how they are programmed.

You will also see any objects that are inside the object you are looking at—provided that the containing object is not set OPAQUE.

If you set an object DARK, it will not show up in a room's contents list. If a player is set DARK, no one will be able to see what that player is carrying.

## More Movement Commands

You can teleport from one room to another by using the teleport command. You can only teleport into rooms that are set JUMP\_OK.

```
> @tel room-number
```

You can also teleport objects that you own into rooms that are JUMP\_OK:

```
> @tel object = room-number
```

So, if you know that a meeting place that you want to go to is room #222, you could get there by typing:

```
> @tel #222
```

If an undesirable player or object is in a room that you own, you can teleport it out of your room to any other room that is JUMP\_OK. You could also teleport it to any other room that is yours. So, if someone barges into your room, where you are having a private conversation with someone, and they won't leave, you can get rid of them!

You can also get rid of them by more violent method of killing them.

## Killing

Killing, in general, is frowned upon. In fact, some muds don't allow player killing. However, one must remember that muds are only games. So, if someone is being obnoxious, and won't stop, sometimes the only way to drive your displeasure home is to kill them off. The command to do this is:

```
> kill thing = money
```

The default cost of killing something is 10 credits. This gives you a 10% chance of success. For each additional credit you spend, your chance of success goes up 1%. Spending 100 credits gives you a 100% chance of success.

Killing something sends it home. The object killed is also paid half the amount of money used to kill it, as a sort of insurance policy.

When an object is killed, the @kill, @okill, and @akill attributes are triggered. They work as you would expect.

Certain things, such as wizards or objects set IMMORTAL, cannot be killed. Only wizards can set things IMMORTAL.

## Giving and Receiving Objects

Occasionally, you may want to give another player an object that you have. Or perhaps you want to give them some money. The commands to do these things are:

```
> give player = object
```

```
> give player = money
```

The player who is receiving the object must have their ENTER\_OK flag set.

Conversely, you can take an object from another player by using the possessive form of the get command:

```
> get player's object
```

The player who is holding the object must be set ENTER\_OK, and the object that you are trying to take cannot be locked against you. This is another reason to use locks on objects. If you don't lock an object, someone else can steal it from you!

Sometimes, instead of just giving a player an object, you may want to allow some other player to own an object that you have created. In order to do this, you must first allow your object to have its ownership changed by setting it CHOWN\_OK. You can then give the object to someone else. The player who receives the object may then change its owner by using the change ownership command, @chown *object* = me.

## Finding and Searching for Objects

In a mud with lots of rooms and objects, you might end up losing one of your objects and forgetting where it is. The @find *string* command will scan through the database and pick out all objects and rooms that have the given string in their names. It prints out the object name and DBREF. @find without any arguments displays all the objects and rooms in the mud that you own. However, it can cost a lot because it takes a lot of computations. Here are some examples

```
@find dog
```

```
@find Speedy
```

Related to the find command is the search command, @search. It scans through the mud and prints everything about an object. There are several arguments you can give to this command: @search *player* will display all the objects owned by that player.

@search *type* = *object-type* will display objects that have that type. For *object-type*, you can use the types: rooms, exits, objects, or players. @search *string* searches for that string. And @search *flags* = *flags-desired* will search for those flags. Here are some examples:

```
@search Speedy
```

```
@search type = objects
```

```
@search dog
```

```
@search flags = D
```

If you want to make sure you don't lose an object, you can link it to yourself. First, set its home to yourself by using the command `@link object = me`. Then, turn on the object's STICKY flag. STICKY objects are sent home when they are dropped. Thus, when the object is dropped, it will return to you.

```
> @link homing pigeon = me
```

```
> @set homing pigeon = STICKY
```

If someone then takes the pigeon, and later lets it go by dropping it, it will return to you.

## Rooms and Exits

In addition to objects and players, you can also create your own rooms, complete with their own descriptions.

As an example, consider building a tree with a treehouse in it. The tree will be in the middle of a field. The treehouse will have a balcony outside.

The field room already exists. It has been set LINK\_OK. The example will describe how to create the tree, the treehouse and the balcony. It will then describe how to link the three rooms to the existing field.

It helps to form a mental picture of the scene you wish to create. This is expressed in Figure 11.1. From the mental image, you can generalize the scene into objects, rooms, and exits, as seen in Figure 11.2.

**Figure 11.1. This is the conceptual idea of the scene.**

**Figure 11.2. Here are the generalized rooms and exits of the scene.**

Now that you have an idea of what you are trying to accomplish, you can begin to create it. To create a room, use the dig command:

```
> @dig room-name
```

This creates a room with that name. The mud responds with a message informing you of the DBREF of the room.

Following the example, create the Tree:

```
> @dig Tree
```

```
Tree created with room number #1442.
```

```
> @tel #11442
```

```
Tree (#1442R)
```

```
> @desc #1442=You see a large tree. There seems to be a treehouse up in the branches.
```

```
Set.
```

The descriptions used here are simplistic because the focus here is room construction. In a real mud, though, a more eloquent description would better convey the sense of the scenery.

This room will initially be disconnected from everything else. You can think of it as floating in hyperspace. You can then teleport into that room using the `@tel` command and set its description and other attributes. However, if your room is not linked to other rooms, you may get a message from time to time indicating that you have an unconnected room. To turn off the message, set the room's FLOATING flag.

Rooms are connected to each other using exits. Exits are the links between rooms. You can create exits by using qualifiers with the `@dig` command, or by creating them directly with the `@open` command.

The format to create exits with the @dig command at the same time you create rooms is:

```
> @dig room-name = in1;in2;in3;... , out1;out2;out3;...
```

In this command, the *in1;in2;in3;...* names are names of entrances into the new room. Each name in the in-list is an alias for the entrance. The *out1;out2;out3;...* names are the names of exits from the new room back into your current room. Each name in the out-list is an alias for the exit. The first name in the list of entrances, *in1*, will be used as the name in the Obvious exits of the current room. The first name in the list of exits, *out1*, will be used in the Obvious exits of the new room.

You can specify as many of these entrance and exit names as you wish; just remember to separate them with semicolons. Note that the list of entrance names is separated from the list of exit names by a comma.

Two links are created in the @dig command above. One link goes from the current room to the new room, and uses the in-list names. That's the entrance to the new room. The other link goes from the new room back into the current room, and uses the out-list names. That's the exit from the new room.

You can use the @dig command with exit names to create the treehouse and the deck in the example. Assume you are in the Tree room:

```
> l
```

Tree (#1442R)

You see a large tree. There seems to be a treehouse up in the branches.

```
> @dig Treehouse=treehouse;house;up;u,tree;down;d
```

Treehouse created with room number 1443.

Opened.

Linked.

Opened.

Linked.

```
> l
```

Tree (#1442R)

You see a large tree. There seems to be a treehouse up in the branches.

Obvious exits:

treehouse

Note how the treehouse exit name now appears in the Obvious exits list. That's because it was the first name in the list of entrances into the new room, *treehouse;house;up;u*.

You can then move into the treehouse by typing one of the entrance names, such as *treehouse*, or *house*, or *up*, or *u*. This moves you through the exit, into the new room that you have just created.

```
> treehouse
```

Treehouse (#1443R)

Obvious exits:

tree

The Obvious exits in the treehouse specifies tree because tree was the first name in the list of exits, tree;down;d, in the @dig command earlier.

Next, we can describe the treehouse and then create the deck:

```
> @desc here=You are in a small treehouse near the top of a large tree. A deck can be seen outside.
```

Set.

```
> @dig Deck=deck;out,treehouse;house;in
```

Deck created with room number 1446.

Opened.

Linked.

Opened.

Linked.

```
> deck
```

Deck (#1446R)

Obvious exits:

treehouse

```
> @desc here = You are standing on the deck attached to the treehouse. A field can be seen below.
```

Set.

Now the tree, the treehouse, and the deck have been created. To link the field to the tree and the deck, you can use the @open command to create specific exits

```
> @open in1;in2;in3;... = #DBREF , out1;out2;out3;...
```

which will open an exit from the current room to the room specified by the DBREF. The exit will have the entrance names specified in the in-list. Another exit will go from the DBREF room to the current room, and will have the names specified in the out-list.

You can leave off the out-list. In that case, the exit from the DBREF room to the current room is not created.

Using the @open command, we can create the exits from the field to the tree:

```
> @open tree;up;u=#1442,field,down;d
```

Opened.

Linked.

Opened.

Linked.

The messages for opening and linking are indicating to you that both the incoming and outgoing links from the tree to room #1442 have been created.

Next, we can create the exit from the deck to the field by teleporting to the deck and then using the @open command without an out-list, thereby creating a one-way link from the deck to the field, but not the other way. Assume the field room has a DBREF of 1440.

```
> @tel #1446
```

Deck (#1446R)

You are standing on the deck attached to the treehouse. A field can be seen below.

Obvious exits:

treehouse

```
> @open field;jump;down;d=#1440
```

Opened.

Linked.

```
> l
```

Deck (#1446R)

You are standing on the deck attached to the treehouse. A field can be seen below.

Obvious exits:

treehouse field

Now the example is complete!

## Rooms as Homes

To set your home to a particular room, use the @link command:

```
> @link object = #room-number
```

If you are in the room you want to change to your home, you can use the shorthand keyword here in place of the *room-number*:

```
> @link me = here
```

You can also allow other players to set their homes to one of your rooms by setting that room's ABODE flag. Rooms that are set ABODE may become other players homes.

In addition, rooms may be linked to other rooms using the same @link command

```
> @link room = #other-room
```

which will make *room* a so-called "drop-to" room. Objects that are dropped in *room* are sent to *other-room*. For instance, in the tree example, you can set the tree's @link to be the field. Then, any object dropped from the tree will fall down to the field:

```
> @tel #1442
```

Tree (#1442R)

You see a large tree. There seems to be a treehouse up in the branches.

Obvious exits:

treehouse field

> @link here = #1440

Dropto set.

> drop brick

Dropped.

> l

Tree (#1442R)

You see a large tree. There seems to be a treehouse up in the branches.

Obvious exits:

treehouse field

> down

Field (#1440R)

You are standing in the middle of a field. There is a large tree here.

Contents:

brick

Obvious exits:

tree north south east west

If a drop-to room is also set to be **STICKY**, then any objects dropped in it will fall to the next room only when all the players have left the drop-to room. Think of the **STICKY** flag on a room as a delayed drop-to action.

## Examining Exits

Exits are really one-way streets from one room to another. To create a path going from one room to another both ways, you actually have to create two exits—one exit going one direction and the other exit going the opposite direction.

Once an exit is created, it shows up if you examine the room from which it leaves. For instance, if we examined the Deck of the example:

> examine here

Deck (#1446R)

Type: ROOM

You are standing on the deck attached to the treehouse. A field can be seen below.

Owner: Speedy Key: \*UNLOCKED\*

Contents:

Speedy (#5Pc)

Field;jump;down;d (#1451E)

Treehouse;house;in (#1448E)

Thus, with the examine command, you can see all the exits that are going *out of* a room.

You can also get a list of all the links going *into* a room by typing:

```
> @entrances #room-number
```

This will also show all the objects whose homes are set to that room.

```
> @entrances here
```

```
Treehouse (#1443R) (Deck;out)
```

```
1 entrance found.
```

## Exit Attributes

Exits can have the same types of attributes that other objects have, such as @succ or @fail messages. Here is a summary of attributes as they apply to exits:

@succ

Message displayed when a player goes through the exit successfully. It is displayed to the player going through the link.

@osucc

Message displayed to the other people in the room from which the player is leaving successfully.

@asucc

Actions to be executed when a player leaves a room successfully. Not present in MUCK.

@fail

Message displayed when a player fails to go through an exit. It is displayed to the player who fails to pass through the link.

@ofail

Message displayed when a player fails to go through an exit. It is displayed to the people in the room from which the player is trying to leave.

@afail

Actions to be executed when a player fails to exit a link successfully. Not present in MUCK.

@drop

This attribute does not apply to exits. It is included here for completeness with the other drop attributes.

@odrop

Message displayed to the other people in a room that a player is entering successfully. Not present in MUCK.

@adrop

Actions executed when a player enters a room successfully. Not present in MUCK.

As a rule, you should define the @osucc, @ofail, and @odrop attributes on any exit you create. In this way, a message will be displayed to the people in the room that the player is leaving, and another message will be displayed to the people in the room that the player is entering. You can also define the @desc attribute on a exit, so that something is displayed to a player who

tries to look at it.

## Special Exits

Most exits from one room lead into another room. However, you can create special NULL exits that display messages to players rather than move them to different rooms. To do this, create an exit on a room, set its lock to #0, and then set its @fail message to the message you want to display when the player types that exit name. You should also set it DARK.

For example, perhaps you want to display the message You wave to the crowd when the player types the command wave. To accomplish this:

```
> @open wave = here
```

Opened.

Linked.

```
> @lock wave = #0
```

Locked.

```
> @fail wave = You wave to the crowd.
```

Set.

```
> @set wave = DARK
```

Set.

```
> wave
```

You wave to the crowd.

NULL exits can also be used to display a message to the user when he or she tries to move in a direction that does not exist. Normally, when a player tries to move in a non-existent direction, a default Huh? (Type 'help' for help) message is displayed to that player. The Huh? message is displayed for any unrecognized command that the player types. A better, more informative message would be something like You cannot go in that direction. To create this, create a NULL exit as mentioned above. Then, for the exit's name list, specify the directions for which you want the message to appear.

For instance, suppose a room only has a north exit. You want to display the You cannot go in that direction message when the player tries to go in a direction other than north. So, create an exit whose name list has all the non-north directions and abbreviations, and then set up its @lock and @fail:

```
> @open east;e;west;w;south;s = here
```

Opened.

Linked.

```
> @lock east = #0
```

Locked.

```
> @fail east = You cannot go in that direction.
```

Set.

```
> @set east = DARK
```

Set .

> east

You cannot go in that direction.

> west

You cannot go in that direction.

> south

You cannot go in that direction.

## Other Flags on Rooms and Exits

To allow people to be able to teleport into your room, set the room JUMP\_OK. To allow people to create links that lead *into* your room, set your room LINK\_OK. Note that other players still can't create links that lead *out* of your room—you have to do that.

The TRANSPARENT flag, when set on an exit, causes the description of the next room to be seen when a player looks at that exit. This allows players to see into adjoining rooms.

## Unlinking Exits

You can unlink an exit by using the @unlink command:

> @unlink *exit*

This will unlink the destination of the exit. You can then relink the destination to some other room using the @link command. Exits become owned by whomever links them, so you shouldn't leave an exit unlinked.

## Containers

You can create objects that can contain other objects by setting the object ENTER\_OK. Then, to go inside the object, use the command:

> enter *object*

and to go out of the object, use the command:

> leave

## Enter and Leave Attributes

The normal @desc attribute is the description of the container, and it is displayed to players that look at the object from the *outside*. The @idesc attribute sets the message that is displayed to anyone *inside* the container who types the look command.

There are several attributes that are triggered when an object is entered by something, or when something exits an object. The @enter attribute is displayed to the entering object, the @oenter attribute is displayed to anyone else inside the object, the @oxenter message is displayed to people who are in the room that the player leaves, and the @aenter attribute is a list of actions that are to be executed when an object is entered. The @leave, @oleave, @oxleave, and @aleave attribute correspond similarly for objects that are leaving a container.

> @create phone booth

phone booth created as object #999.

Set .

```
> @desc = You see a telephone booth here.
```

```
Set.
```

```
> @idesc = You are inside a phone booth. There is a coin-operated telephone attached to one side of the booth.
```

```
> @enter phone booth = You enter the phone booth.
```

```
Set.
```

```
> @oenter phone booth = :squeezes into the phone booth with you.
```

```
Set.
```

```
> @oxenter phone booth = :enters the phone booth.
```

```
Set.
```

```
> @leave phone booth = You leave the phone booth.
```

```
Set.
```

```
> @oleave phone booth = :departs from you, leaving the phone booth.
```

```
Set.
```

```
> @oxleave phone booth = :leaves the phone booth.
```

```
Set.
```

## Pronoun Substitutions

Sometimes you want to create sentences in your attribute messages that depend on the gender or name of the player performing some action on an object. The attributes that begin with @a (such as @asucc, @afail, @adrop, and so forth) frequently use special substitutions in their actions.

For instance, perhaps you want to create a cat that meows when it is dropped. You want to display a message to everyone that the cat meows at him when dropped by a male player, and meows at her when dropped by female player. In addition, you want to use the dropping player's name in the message. You can do this using special pronoun substitutions:

```
> @adrop cat = %N drops the cat, which lands on the ground and meows at %o.
```

In the preceding message, the player's name is substituted for the %N portion of the message, and the objective form of the player's name is substituted for the %o part of the message. In this manner, a male player named Linus would trigger a message Linus drops the cat, which lands on the ground and meows at him, while a female player named Lucy would trigger the message Lucy drops the cat, which lands on the ground and meows at her. The mud determines which pronoun to substitute by looking at the gender of the player, which is stored in the @sex attribute.

There are several types of pronoun substitutions:

```
%N, %n
```

Substitutes the player's name. %N is the capitalized form; %n is not capitalized.

```
%S, %s
```

Substitutes the subjective form of the player's name (he/she/it/they). %S capitalizes the first letter; %s does not.

`%O`, `%o`

Substitutes the objective form of the player's name (him/her/it/them). `%O` capitalizes the first letter; `%o` does not.

`%P`, `%p`

Substitutes the possessive form of the player's name (his/her/its/their). `%P` capitalizes the first letter; `%p` does not.

`%A`, `%a`

Substitutes the absolute possessive form of the player's name (his/hers/its/theirs). `%A` capitalizes the first letter; `%a` does not.

## Try It Out!

This information is enough to get you started on creating objects, setting some of their attributes, building some rooms and linking them up with exits, and making your own little virtual world.

The next few sections will describe some specifics for several types of muds: MUCKs, MUSHes, and MUSEs.

## Programming a MUCK

MUCK, which is a pun on the term MUD, is a spin-off of the original TinyMUD. This section describes TinyMUCK version 2.2.

### Popular MUCKs

There are many popular MUCKs on the Internet. Some of them were among the first muds to appear on the Internet and, as a result, have a large population of longtime mud players.

<i>Name</i>	<i>Machine Name</i>	<i>Internet Address</i>	<i>Port</i>
AnimeMUCK	anime.tcp.com	128.95.10.106	2035
CaveMUCK	cave.tcp.com	128.95.44.29	2283
FurryMUCK	sncils.snc.edu	138.74.0.10	8888
NAILS	flounder.rutgers.edu	128.6.128.5	5150

FurryMUCK, one of the earliest muds, harbors a large group of devoted players. Its theme is anthropomorphic animals—characters that are animals with human characteristics.

These addresses are current as of August, 1994. Keep in mind that they may change.

## Things Specific to MUCKs

From the player's standpoint, MUCKs operate similarly to other types of mud on the Net. Beneath the visible text, however, programmers will find that they are significantly different. This section indicates some of the principle features of programming in MUCKs.

### MUF Programming

In order to program more complex objects in MUCK, you have to learn how to program in the MUCK programming language, called Multi-User Forth, or MUF for short.

### Using the MUF Editor and Compiler

MUF programs must be entered into the MUCK and then compiled in order for them to function. To enter the editor, you can type:

```
@program program-name
```

This will put you into the MUCK editor. You can then enter insert mode by typing

```
line-number i
```

which will start inserting what you type before the given *line-number*. To exit insert mode, type a period (.) on a line by itself.

You can then compile the program by typing:

```
c
```

And, finally, you can run the program by typing:

```
????
```

A more detailed listing of editor commands can be found in the MUCK Reference Manual section, later in this chapter. This should be enough to get you started.

## MUF Basics

In MUF programs, comments are enclosed in parentheses:

```
(This is a comment)
```

MUF programs consist of *words*. A word is simply a series of statements. They usually manipulate the stack in some manner. A word begins with a semicolon, followed by the name of the word, the statements of the word, and then ending with a semicolon:

```
: word-name  
  
    statements
```

```
;
```

There are three basic types of values in MUF: the integer, the string, and the DBREF. Strings are enclosed in double-quotes. DBREFs are denoted by putting a pound symbol (#) in front of the DBREF number:

```
7          (This is an integer)  
  
"Test"    (This is a string)  
  
#354      (This is a DBREF)
```

MUF is a stack-based programming language based on Forth. If you are not familiar with the concept of a stack, it is a place to hold information, and it has a "top" element that can be accessed as necessary. When you "push" an element onto the stack, that element becomes the new "top" of the stack, similar to stacking plates on top of one another. The first element to be removed, or "popped" from the stack is the top element. The next element to be removed is the element under the top element, and so forth.

You can put these basic values onto the stack by simply specifying them in your MUF words:

```
: stack_test  
  
    56  
  
    "First string"  
  
    #45  
  
    "Second string"  
  
;
```

In this example, the stack now has four values on it. The number 56 is at the bottom of the stack because it was pushed onto the stack first. The Second string is on the top of the stack, and we can access it if we look at the top of the stack or pop the top

value from the stack.

The indentation did not matter in the previous example. Indentation is simply used to make MUF programs easier to read. The previous example can also be written:

```
: stack_test
    56 "First string" #45 "Second string"
;

```

## Basic MUF Functions

MUF functions operate on the top of the stack. A value may be taken from the top of the stack, processed in some manner, and perhaps put back onto the stack. More commonly, the top two values are taken from the top of the stack and then processed in some manner to result in one new value, which is then placed back onto the top of the stack. The addition function behaves in this manner. It takes the top two values from the stack, adds them together, and then places the result back onto the stack:

```
: simple_add
    5 8 +
;

```

This simple addition would first place 5 onto the stack, followed by 8. The + operator would then pop these two values off the stack, add them together, and push the result, 13, back onto the stack. The number 13 could then be used in some fashion.



**Note:** This is the same as RPN (Reverse Polish Notation), also known as *postfix*. Anyone familiar with the high-end HP calculators will know how to use it already.

---

Another example is

```
: more_add
    4 5 9 + 8 + +
;

```

which first puts 4, 5 and 9 on the stack. It then adds the top two values, 5 and 9, and pushed their sum, 14, back onto the stack. It then pushes 8 onto the stack. The next addition adds 14 and 8 together to get 22, and the last addition adds 4 and 22 together to get 26. Thus, at the end, the stack has one value: 26.

Subtraction, multiplication and division work in a similar manner. For subtraction, the top value is subtracted from the value underneath it. For division, the second-from-the-top value is divided by the top value.

The equality function is an equals sign (=). It pops the top two integers from the stack and compares them. If they are equal, 1 is pushed onto the stack. If they are not equal, 0 is pushed onto the stack. Thus:

```
: equals_test
    4 100 = 7 7 =
;

```

would result in a stack of: 0 1.

Similar to the equality function, the `dbcmp` function also compares two values on the top of the stack. It compares two DBREFs. If they are equal, 1 is pushed, otherwise 0 is pushed.

```
: dbcmp_example

```

```
#54 #54 dbcmp
```

```
;
```

would result in the stack consisting of the number 1.

As mentioned previously, the top value can be removed from the stack by "popping" it. The pop function accomplishes this in MUF programs, appropriately enough:

```
: pop_example
```

```
5 7 pop 8 9 pop
```

```
;
```

This places 5 and 7 onto the stack, pops 7 back off, then places 8 and 9 onto the stack and pops 9 back off. The stack would thus be 5, 8, with 8 being the top of the stack.

Another basic function is the swap function. It switches the top two values on the stack around:

```
: swap_example
```

```
4 5 swap 7 swap
```

```
;
```

This program would place 4 and 5 onto the stack, then swap them to get 5 and 4. The number 7 is then pushed onto the stack, to get 5, 4 and 7. The last swap would swap 4 and 7. The end result is 5, 7, 4, with 4 being at the top of the stack.

One more frequently used function is the duplicate function, dup. It simply pushes a copy of the top of the stack onto the stack:

```
: duplicate_example
```

```
4 dup 7 dup
```

```
;
```

After this function, the stack would be 4, 4, 7, 7. The second 7 would be on the top of the stack.

## Variables

You can use variables in MUF code. First, you must declare the variable by putting:

```
var variable-name
```

at the top of the program. Once declared, the variable can be used in a variety of ways.

The variable storage function is an exclamation point (!). It places the next-to-the-top value into the variable named on the top of the stack.

```
var testvar
```

```
: variable_storage_example
```

```
7 testvar !
```

```
;
```

This puts the value 7 into the variable named testvar.

To retrieve the value of a variable, use the fetch function, which is an "at" symbol (@). It pops the top value on the stack and uses it as a variable name. The contents of that variable are then placed back onto the stack:

```

var testvar

: variable_fetch_example

    7 testvar !

    testvar @

;

```

This first places the value 7 into the variable named testvar. It then accesses testvar and places its contents, 7, onto the stack. The stack, then, would consist only of the number 7.

Three special variables in MUF are me, loc, and trigger. The me variable contains the DBREF of the player, and loc contains the DBREF of the player's current room. The trigger variable contains the DBREF of the player who triggered the program. These variables will be used in later examples.

## Messages

To send a message to an object, you use the notify function. This function sends the string at the top of the stack to the player at the second top element of the stack:

```

: notify_example

    me @ "Hello, there!" notify

;

```

This example first retrieves your DBREF from the me variable. It then pushes a string onto the stack. Finally, it uses the notify function to send that string to you.

## Conditional Statements

MUF has an IF-THEN construct similar to other programming languages. However, it has a major difference with the IF-THEN statement you may be familiar with from other programming languages. In MUF, the IF-THEN statement is more like a compare and jump statement in assembly language.

```

if

    if-statements

then

    following-statements

```

In the preceding syntax, the if keyword pops the top value from the stack. If the value does not equal 0, the *if-statements* are executed, and then the *following-statements* are *also* executed. The *if-statements* consist of all statements up to the next then keyword. If the value popped by if is 0, the *if-statements* are skipped, and only the *following-statements* are executed.

As an example, consider:

```

: if_example

    4 2 2 + =

    if "The values were equal"

    then "Top string"

;

```

First, 4 and the sum of 2 and 2 are compared, using the equality function. Since they are equal, the if statement pushes The values are equal onto the stack. It then pushes the string Top string onto the top of the stack.

The program

```
: if_example
    4 2 3 + =
    if "The values were equal"
    then "Top string"
;
```

would only push the string Top string onto the stack because 4 and 5 are not equal.

As a helpful hint, it may be useful to think of the then keyword as an endif.

## An Example MUF Program

With these concepts, you can create many MUF programs. In addition, you can use the built-in functions that are available in MUF. For a listing of these, consult the MUCK Reference Manual, later in this chapter.

## Programming a MUSH

MUSH, which stands for Multi-User Shared Hallucination, is a spin-off of the original TinyMUD. This section describes TinyMUSH version 2.0.

### Popular MUSHes

There are several MUSHes on the Internet. Many of them have certain themes.

Name	Machine Name	Internet Address	Port
Deep Seas	muds.okstate.edu	139.78.9.1	6250
DuneMUSH	mellers1.berkeley.edu	128.32.243.78	4201
NarniaMush	argo.unm.edu	129.24.9.24	6250
NeverendingStory	jove.cs.pdx.edu	131.252.21.12	9999
PernMUSH	cesium.clock.org	130.43.2.43	4201
Shadowrun	yacht.slip.andrew.cmu.edu	128.2.116.75	4201
TinyCWRU	caisr2.caisr.cwru.edu	129.22.24.22	4201

DeepSeas, previously known as SpaceMadness, DreamScape, Asylum, Chaos, and TinyHELL, is another long-standing mud with a consistent player base. A lot of dino mudders (long-timers) can be found there.

These addresses are current as of August, 1994. Keep in mind that they may change.

## Things Specific to MUSHes

In addition to the commands and objects previously mentioned in the tutorial section, there are a few things that are specific to MUSHes.

### Possessive Pose

You can use a semicolon to possessive-pose. This is the same as posing (acting), except that it adds a possessive *s* after your character name and before the action message

```
;message
```

```
> ;brain hurts.
```

Speedy's brain hurts.

## Puppets

Puppets are special types of objects that seem just like players. Puppets can be controlled by their owners and can see and hear things in a room just like a player would. The controlling player can then use the information returned by the puppet because the puppet echoes the text it sees and hears back to its owner. It does not echo back actions that are performed in the same room as its owner, to prevent message confusion.

To change an object into a puppet, set the object to be a puppet: `@set object = puppet`. The puppet will announce that it has grown ears and can now hear. After it becomes a puppet, you can use its name or its DBREF to have it perform actions that you want.

You can force an object to perform some action by using `@force object = actions`. Or, you can use *DBREF actions*. The object will behave as if it, itself, had typed the action.

```
> @create a little dog
```

```
a little dog created as object #333
```

```
> @set little dog = puppet
```

```
a little dog grows ears and can now hear.
```

```
Set.
```

```
> inventory
```

```
You are carrying:
```

```
a little dog (#333p)
```

```
You have 930 credits.
```

```
> @desc little dog = You see a little puppy.
```

```
Set.
```

```
> drop little dog
```

```
a little dog> Speedy's Room
```

```
a little dog> You are in Speedy's Room.
```

```
a little dog> Contents:
```

```
a little dog> Speedy (#5Pc)
```

```
a little dog> Obvious exits:
```

```
a little dog> north
```

a little dog> Dropped.

Dropped.

> l little dog

a little dog (#333p)

You see a little puppy.

> @force little dog = :wags its tail.

a little dog wags its tail.

> #333 :pants happily.

a little dog pants happily.

> #333 "Yap yap!"

a little dog> You say "Yap yap!"

a little dog says "Yap yap!"

> #333 north

a little dog walks north out of Speedy's Room.

a little dog has left.

a little dog> Hallway

a little dog> You are in a short hallway. Speedy's room is to the south. A stairway can be seen to the north.

a little dog> Obvious exits:

a little dog> north south

> #333 south

a little dog> You walk south into Speedy's Room

a little dog> Speedy's Room

a little dog> You are in Speedy's Room.

a little dog> Contents:

a little dog> Speedy (#5Pc)

a little dog> Obvious exits:

a little dog> north

a little dog walks in from the north.

a little dog has arrived.

```
> @set #333 = DESTROY_OK
```

Set.

```
> @destroy #333
```

a little dog> \*NOTHING\*

You get back your 10 credit deposit for a little dog (#333).

Destroyed.

## Listening and Responding to Messages

Sometimes it's fun to program an object that listens for certain strings and responds to those strings in some manner. The @listen directive will set the pattern string and the @ahear attribute sets the actions that are executed when the listen string is matched.

```
> @listen object = string
```

```
> @ahear object = actions
```

When using @listen, the string frequently has some word or phrase that the object is listening for, plus some wildcard characters to match whatever else might be in a message. If a matching string is seen, the @ahear actions are then executed. For instance, the command

```
> @listen me = * movie*
```

would be looking for a string of anything, followed by a space, followed by the word movie, followed by anything else. This could match something like Speedy went to see a movie today. It could also match What's playing at the movies tonight?.

```
> @listen me = * movie*
```

Speedy - Set.

```
> @ahear me = :wanna direct!
```

Speedy - Set.

Then, later, maybe someone mentions movies. That triggers the response automatically:

Archimedes went to see a movie last night.

Speedy wanna direct!

Mak says, "Which movie did you see?"

Speedy wanna direct!

In addition to the @ahear attribute, there is also an @amhear and an @aahear attribute. @ahear responds only to messages that the listening object did not generate. @amhear only responds to messages that the object itself generated. @aahear responds to all messages.

## Percent Substitutions

Whenever an action occurs, the mud stores information related to that action in a special storage area. These variables can then

be accessed using percent substitutions.

Percent substitutions are simply variables that begin with a percent sign. For instance, in the previous section, you learned about pronoun substitutions. Pronoun substitutions are just special kinds of percent substitutions.

One of the items that the mud stores is the name of the *enactor*—the object that caused an action. As described in the previous section, the name of the enactor is stored in the %N variable. The %N variable is the capitalized name, such as Speedy, and the %n variable is the uncapitalized form, such as speedy. Additionally, the actual DBREF of the enactor can be accessed by using the %# variable.

Another item stored by the mud is the *affected object*—the object on which the action is being performed. The DBREF of this object can be accessed by using the %! variable.

For example, if Speedy dropped a bowling ball, Speedy would be the enactor, and the bowling ball would be the affected object. Speedy's name could be accessed using %N or %n, and Speedy's DBREF could be accessed using %#. The DBREF of the bowling ball could be accessed using %!.

The DBREF of the location of the action can be accessed through the %l variable.

There are several special percent substitutions that are used for text formatting. These formatting characters resemble the special formatting characters in the C programming language: %r converts into a carriage return-newline combination, %t converts into a tab character, %b converts into a blank space, and %% converts into an actual percent sign.

That is why you have to type *two* percent signs when you want to say a line with a percent sign in it. If you say

```
> "24.6% of all statistics are made up on the spot!"
```

```
Speedy says "24.6 of all statistics are made up on the spot!"
```

you notice that the percent sign disappears. The mud is trying to convert it to a percent substitution, and a percent sign followed by a blank space converts into a blank. To get a percent sign into your text and actions, type *two* percent signs

```
> "24.6%% of all statistics are made up on the spot!"
```

```
Speedy says "24.6% of all statistics are made up on the spot!"
```

since the %% is the formatting specification for a percent sign.

Another way to get an actual percent sign is to use the escape character, which is a backslash (\):

```
> "24.6\% of all statistics are made up on the spot!"
```

```
Speedy says "24.6% of all statistics are made up on the spot!"
```

The escape character causes any of the characters following it to be printed as-is.

## Registers

In addition to the various percent substitutions just mentioned, there are 26 registers that you can set using @va through @vz and access by using %va through %vz. These registers are like attributes or variables. They can store just about any mud value—numbers, strings, messages, function results, and so forth. To set a register, use the syntax

```
> @register object = actions
```

and then access it using the corresponding percent-substitution, %register.

As an example, consider creating a water balloon that splats when it is dropped. The va register will be used to print out a message when the balloon is dropped.

```
> @create water balloon
```

```
water balloon created as object #1765.
```

```
> @va water balloon = SPLAT!
```

Set.

```
> @drop water balloon = The balloon falls and goes %va.
```

Set

```
> drop water balloon
```

The balloon falls and goes SPLAT!

## Triggers

You can also use the register to perform an action. To do this, you have to *trigger* the register using the trigger command:

```
> @trigger object/attribute
```

The @trigger command can be abbreviated @tr.

In the water balloon example, a more realistic action would be for the water balloon to destroy itself when it is dropped. You can put the destroy command in a register, and then trigger it using the trigger command when the balloon is dropped. So, adding to the water balloon example:

```
> @vb water balloon = @destroy me
```

Set.

```
> @adrop water balloon = @tr me/vb
```

Set.

```
> drop water balloon
```

The balloon falls and goes SPLAT!

water balloon has left.

You get back your 10 credit deposit for water balloon (#1765).

When the balloon is dropped, the @drop attribute used the va register as a string, and the @adrop attribute triggered the vb register, which destroyed the balloon.

With 26 registers to play around with, you can perform a lot of complex programming. You can use registers to store temporary information. Registers can trigger other registers, too.

## Numbered Variables

You can also make use of the 10 numbered variables, named %0 through %9. These variables are collectively known as the *stack* in mud terms, although it isn't really a stack in the computer science sense of pushing and popping objects. The stack is just a storage area for the 10 numbered variables.

These numbered variables are known as *positional parameters*. They are set consecutively using two methods: the @listen command and the @trigger command.

To illustrate the @listen command and how positional parameters are set, consider a parrot that listens for phrases with the word *Parrot* in them. The parrot will speak the three phrases that come after the trigger word Parrot:

```
> @create parrot
```

parrot created as object #1799.

```
> @listen parrot = * says "Parrot * * * *
```

```
parrot - set.
```

```
Set.
```

```
> @ahear parrot = "SQUAWK %1 %2 %3
```

```
Set.
```

```
> drop parrot
```

```
Dropped.
```

```
> "Parrot can you hear?"
```

```
You say "Parrot can you hear me?"
```

```
parrot says "SQUAWK can you hear"
```

```
> "Parrot tell a story to us."
```

```
You say "Parrot tell a story to us."
```

```
parrot says "SQUAWK tell a story"
```

In this simple example, the asterisks in the @listen command convert to numbered variables %0, %1, %2, %3, and %4. The first asterisk matches the player's name from the spoken text, and this is placed in the variable %0. The three phrases after the string Parrot are matched and placed into numbered variables %1, %2, %3, and %4, respectively. The %1 variable is the first word after Parrot, %2 is the second word, %3 is the third word, and %4 is everything else, including the ending double-quote. The @ahear command then uses some of these variables in its spoken text.

The @trigger command can also be used to set the numbered variables, using the format:

```
> @trigger object/attribute = item, item, item, ...
```

This command sets the positional parameters to the given items, consecutively, starting from %0. The attribute that is triggered can then access these numeric variables.

Consider a globe that has its va attribute programmed as follows:

```
> @va globe = @emit The globe %0, %1, and %2.
```

```
Set.
```

Now, to get %0, %1, and %2 to be shimmers, sparkles, and glows eerily, you can use the @trigger command:

```
> @tr globe/va = shimmers, sparkles, glows eerily.
```

```
Triggered.
```

```
The globe shimmers, sparkles, and glows eerily.
```

Note how the parameters to the @trigger command become the numeric variables for the va attribute.

## Braces

You can use *braces* to surround strings so that commas within them do not get interpreted by the mud as separation characters.

Consider the globe in the previous section. What if you wanted the second phrase to be hums, glistens instead of sparkles? If you type

```
> @tr globe/va = shimmers, hums, glistens, glows eerily.
```

you get

Triggered.

The globe shimmers, hums, and glistens.

This is because the comma separating hums and glistens is taken to be a parameter separator.

In cases like this, you want a comma to be taken as an actual comma, rather than a parameter separator. You can use braces to specify that the string within the braces is to be taken as a whole unit, and any commas within the braces are part of the string:

```
> @tr globe/va = shimmers, {hums, glistens}, glows eerily.
```

Triggered.

The globe shimmers, hums, glistens, and glows eerily.

So remember, when you want to get an actual comma into a string, you may need to throw a pair of braces around the string. Braces are also frequently used in function calls and switch statements (both discussed later in this section), since those programming constructs commonly use commas.

## Using an Object

You can also trigger an object by *using* it. To use an object, simply type the use command:

```
> use object
```

This command triggers the @use, @ouse, and @ause attributes. The object will respond to the use command only if the @ause attribute has been set on the object.

## Functions

The format for a function call is:

```
[ function-name(parameters...) ]
```

The parameters of the function must be enclosed within parentheses, and the function itself must be enclosed in square brackets.

As an example, the name() function returns the name of an object when given its DBREF. If object #5 was the player Speedy, you could determine the name by using the name() function and the object number:

```
> @emit [name(#5)]
```

Speedy

Functions can be nested, and nested functions only require one outermost set of square brackets. You can put the name() function, mentioned earlier, and the Ustrlen() function, which returns the length of a string, together to find out the length of the name of object #5 by typing:

```
> @emit [strlen(name(#5))]
```

6

Sometimes, you may want to evaluate the value of an attribute as though it were a function. In those cases, you may have to add additional square brackets to denote that you are trying to access a variable as a function.

For instance, the get(object/attribute) function returns the value of the named attribute on a given object. If the va and vb attributes on some test object are defined as

```
> @va testobject = vb
```

Set.

```
> @vb testobject = blah blah
```

Set.

then by accessing va you see

```
> @emit [get(testobject/va)]
```

vb

and by accessing vb indirectly through va you see

```
> @emit [get(testobject/[get(testobject/va)])]
```

blah blah

because `[get(testobject/va)]` returns vb, which is substituted into the outermost `get()` function to return blah blah.

There are dozens of predefined functions to choose from. A few important functions are discussed next. For a description of the rest of the functions available, or for quick reference, see the function glossary section, later in this chapter

## The `v()` Function

The `v()` function is simply another way to access the 26 v-registers. The code `[v(va)]` is functionally equivalent to `%va`. However, the percent sign form is evaluated faster than the function call form, so it tends to be used in most cases.

## The `s()` Function

The `s()` function performs pronoun substitution on a string, and returns that string. The pronouns apply to the object that triggers the function.

Thus, if your character has male gender, and you evaluate

```
> @emit [s(%P went down the road.)]
```

you see

He went down the road.

And, if your character had female gender, you would see:

She went down the road.

All the pronoun substitutions, discussed earlier, can be used in the `s()` function.

## The `get()` Function

The `get()` function, mentioned earlier, is another frequently used function. It is used to access various attributes from a particular object. Its syntax is

```
[get(object/attribute)]
```

## The `rand()` Function

This function tends to be used frequently. The syntax is

```
rand(number)
```

and it returns a random integer between 0 and (*number* - 1).

## The Switch Command

You can create conditional expressions on an object by using the switch command. The switch command is similar to a CASE statement in a programming language. There is a conditional expression that is evaluated and, based on the results of that evaluation, different actions are performed.

The format for the switch command is:

```
> @switch test = condition1, action1, condition2, action2, ..., conditionN, actionN,
default
```

The switch command tests the result of the *test* against each *condition*. If there is a match, the corresponding *action* is performed.

As an example, consider programming a dog to obey commands you give it. When you tell the dog to sit, it sits down; when you tell the dog to speak, it barks, and so forth. You can accomplish this by using the switch command. First, set up the dog to listen for commands:

```
> @create Rover
```

Rover created as object #6745.

```
> @listen Rover = *"Rover, *"
```

Set.

Now, use the switch command on the @ahear attribute to perform different actions based on what a player commands Rover to do:

```
> @ahear Rover = @switch %1 = sit , :sits down. , speak , "RARWK!, jump, :does a
backflip., play dead, :lies down on the ground., beg, {:whimpers, cries, and looks
utterly heartbroken.}
```

Set.

Now you can tell Rover what to do:

```
> drop Rover
```

Dropped.

```
> "Rover, sit
```

You say "Rover, sit"

Rover sits down.

```
> "Rover, speak
```

You say, "Rover, speak"

Rover says "RARWK!"

```
> "Rover, jump
```

You say "Rover, jump"

Rover does a backflip.

```
> "Rover, play dead
```

```
You say "Rover, play dead"
```

```
Rover lies down on the ground.
```

```
> "Rover, beg
```

```
You say "Rover, beg"
```

```
Rover whimpers, cries, and looks utterly heartbroken.
```

Note that the conditions and actions are separated by commas. As mentioned previously, if you want to use a comma or a semicolon in one of your conditions or actions, make sure you surround the statement with braces ({}). Otherwise, a comma signifies a new condition or action, and a semicolon signifies the end of the switch command.

The last action in the switch command for Rover is enclosed in braces because the action contains commas within it.

As another example, you can achieve an IF-THEN-ELSE effect by using a switch statement and testing for either 1 or 0 in the conditionals:

```
> @create Parrot
```

```
Parrot created as object #6746.
```

```
> @listen Parrot = * has arrived.
```

```
Set.
```

```
> @ahear Parrot = @switch [eq("Speedy", %0)] = 1, {:flies around squawking, "Speedy is here! Speedy is here!"}, 0, :ignores %0.
```

The test portion of the switch command tests to see if Speedy is the name of the player who is entering the room. If so, it flies around and announces Speedy's entrance; if not, it ignores the entering player:

```
Snowball has arrived.
```

```
Parrot ignores Snowball.
```

```
Speedy has arrived.
```

```
Parrot flies around squawking, "Speedy is here! Speedy is here!"
```

```
Pooka has arrived.
```

```
Parrot ignores Pooka.
```

Again, note the braces around the true part of the command because of the comma within it.

## User-Defined Commands

Perhaps you want to create an object and then define some specialized commands on that object. You can create a stuffed bear that growls when you squeeze it. The squeeze command is not a predefined command in the mud. You have to define it. You can create user-defined commands by setting them on a register:

```
> @attribute object = $command:actions
```

To define the squeeze command on the bear:

```
> @va bear = $squeeze:@emit The stuffed bear grrrowls huskily.
```

Then, when you type

```
> squeeze
```

you see

The stuffed bear grrrowls huskily.

You can define multiple commands on an object. Just make sure that you don't pick a command name that is already used, such as page or get, because predefined mud commands take precedence over user-defined commands.

You can allow your user-defined commands to accept arguments, too. You specify these arguments by using asterisks (\*) in the command definition. These asterisks are converted into the positional parameters for the attribute that defines the command. For instance, you can change the squeeze command to accept an argument:

```
> @va bear = $squeeze *:@emit You squeeze the bear's %0. The bear grrrowls huskily.
Set.
```

```
> squeeze tummy
```

You squeeze the bear's tummy. The bear grrrowls huskily.

## User-Defined Attributes

In addition to user-defined commands on an object, you can also create user-defined attributes on an object. The format is:

```
> &attribute object = anything
```

You can also define an attribute using

```
> @set object = attribute:anything
```

You can set an attribute on an object to anything you want. You can use it for temporary storage of variables, names or DBREFs of objects, strings, and so forth. You can define an unlimited number of attributes on an object.

Make sure that your user-defined attribute does not conflict with an already-defined attribute. As you create more and more complex objects, the chances increase for you to think of an attribute that you have already defined on that object, and you end up redefining something that you have already written!

You can access these attributes using the get() function or the v() function.

```
> @va bear = $squeeze:@emit You squeeze the bear.;&lastsqueeze bear = %N
```

Set.

```
> @vb bear = $last:@emit This bear was last squeezed by [get(me/lastsqueeze)].
```

Set.

```
> squeeze
```

You squeeze the bear.

```
> last
```

This bear was last squeezed by Speedy.

This example sets a user-defined attribute called lastsqueeze whenever the bear is squeezed. The lastsqueeze attribute is set to the squeezing player's name. Then, when the player types last, this attribute is used to recall which player last squeezed the bear.

## The Queue

Each command that the mud executes goes into a queue first. The mud then runs commands out of the queue. Most of the time,

the commands are executed immediately. However, some commands delay action for a few rounds, such as @trigger and @force.

To get a listing of the queue, you can use the @ps command. This command displays four queues. The Player Queue shows actions that you have triggered. The Object Queue shows actions that have been triggered by objects that you own. The Wait Queue displays commands that have been queued up and thus will be executed at some future point in time. And the Semaphore Queue shows semaphores, explained next. The @ps command has options you can use to obtain different process displays, such as a verbose listing, a brief listing, or a quick summary. See the Help on your mud for a description of these options to the @ps command.

## The Wait Command

You can delay commands so that they are executed at a later time by using the wait command:

```
> @wait #seconds=actions
```

This queues up the specified actions and runs them after waiting the specified number of seconds. For instance, you can set up a twilight effect by setting up these attributes on a test object:

```
> @va testobj = @wait 5 = @emit It is starting to get dark.;@tr me/vb
```

Set.

```
> @vb testobj = @wait 10 = @emit It gets darker.;@tr me/vc
```

Set.

```
> @vc testobj = @wait 15 = @emit It's completely dark now.
```

Set.

Then, if you trigger the va attribute, the messages are displayed after each wait delay:

```
> @tr testobj/va
```

A delay of 5 seconds and you see:

```
It is starting to get dark.
```

Then, 10 seconds later:

```
It gets darker.
```

And, finally, 15 seconds after that:

```
It's completely dark now.
```

## The Halt Command

Sometimes, though, the commands that you have queued up will interact with each other in such a way that things get out of control. When these infinite loops occur, you can use the halt command to stop them.

```
> @halt
```

clears out your personal queue,

```
> @halt command
```

clears out your queue and puts the specified command onto the queue, and

```
> @halt object=command
```

clears out the given object's queue and puts the specified command onto the queue.

## Semaphores

One of the more advanced mud concepts is the *semaphore*. A semaphore is a signal that indicates that it is okay for something to perform some action. On many real-life train tracks, semaphores alongside the road indicate that a train has recently passed. The semaphore is initially green, indicating to an approaching train that it is okay for the train to continue. Once the train has passed, the semaphore changes to red. Another approaching train would see the red semaphore and know that another train recently passed through, and it may be necessary for the second train to slow or stop in order to avoid hitting the back end of the first train. A timer on the semaphore eventually changes the red light back to green.

Similarly, a semaphore in a mud is used to signal when certain actions can occur and prevent other actions from occurring until the correct time.

One typical use of a semaphore is to prevent an attribute or object from being triggered until it finishes executing its actions. For instance, you have a vending machine that dispenses magic baseballs. When it is triggered, it starts performing actions to create a new magic baseball and to give that baseball to you. A semaphore can be used to prevent someone from coming along and triggering the machine again while it is in the middle of first action list.

Another common use of semaphores is to force commands to be executed in a specific order. You can create user-defined commands hop, skip, and jump, and force them to be executed in that order through the use of semaphores.

Each semaphore has a *count* associated with it. The count is the number of actions that are currently blocked, waiting for a notification to unblock them and thus allow them to execute. If the count is 0, an action will occur immediately. A positive number indicates that that many actions are currently blocked. A negative number indicates that that many actions in a row will immediately execute.

The @wait command is used to place an action list on a semaphore and then "block" those actions from occurring until a notification occurs. The syntax is

```
> @wait object=actions
```

which will set up a semaphore on the object and enqueue the given actions onto the semaphore's pending action list. If the actions are a list of commands, you have to enclose the list in braces. You can also set up a default execution time by using

```
> @wait object/#seconds=actions
```

Such a wait command will behave like a normal wait command, except that after the given number of seconds, the actions execute anyway, regardless of whether an actual notification has occurred. You can use this to make sure a set of actions is eventually executed, even if no notification ever occurs.

The @notify command is used to notify the semaphore. When notified, the semaphore will execute the first action in the action list. The action item will be "unblocked." A second notification will start the second action, and so forth. To notify a semaphore on an object, use

```
> @notify object
```

which will execute the first action that is pending on the object's semaphore queue. You can also execute a given number of actions by specifying that number

```
> @notify object=#notifications
```

which executes that many actions on the object's semaphore queue.

Finally, the @drain command resets the semaphore count to 0.

Semaphores are usually used on larger, more complex objects, such as vending machines, mud e-mail systems, and so forth. As a simple example of semaphores, consider constructing a trick "ack-in-the-box." The box contains a cat that springs out and coughs up a hairball when the box is wound up. The idea is for the commands relating to the hairball to be executed in a group, without interruption, so a semaphore is used:

```
> @create ack-in-the-box
```

```
ack-in-the-box created as object #1515.
```

```
> @desc ack= This is an ack-in-the-box. Type 'wind' to wind it up.
```

Set.

Now set up the ack-in-the-box to be ready to execute a wait command immediately by setting up a notification for right now, and for when the mud starts up. The startup actions make sure that the ack-in-the-box begins with its semaphore at -1, which means that any action on it will not be blocked.

```
> @notify ack
```

Notified.

```
> @startup ack=@drain me;@notify me
```

Set.

Now the object is ready to execute an action on it. This next section of code sets up a user-defined command, wind, that performs various actions and then, as its last action, renotifies the ack-in-the-box. This will allow the next wind command to immediately execute:

```
> &do_wind ack=$wind:@wait me={@emit The box whirrs and humms. Suddenly, the box
SPROINGS open, and a scruffy-looking orange cat SPRINGS out of the box!;@create
hairball;@desc hairball=You see a slimy cat hairball.;@drop hairball=The slimy cat
hairball splats on the floor.;@adrop hairball=@emit A slimy cat hairball splats onto
the floor.;@set hairball=DESTROY_OK;@emit The cat says "ACK!" and coughs up a
hairball!;drop hairball;@emit The cat is then pulled back into the box, which
closes.;@notify me}
```

Set.

And test it out!

```
> drop ack-in-the-box
```

Dropped.

```
> l
```

Speedy's Room

You are in Speedy's Room.

Contents:

ack-in-the-box (#1515)

Obvious exits:

out

```
> l ack
```

This is an ack-in-the-box. Type 'wind' to wind it up.

```
> wind
```

The box whirrs and humms. Suddenly, the box SPROINGS open, and a scruffy-looking orange cat SPRINGS out of the box!

The cat says "ACK!" and coughs up a hairball!

ack-in-the-box dropped hairball.

The cat is then pulled back into the box, which closes.

A slimy cat hairball splats onto the floor.

> l

Speedy's Room

You are in Speedy's Room.

Contents:

hairball (#1530d)

ack-in-the-box (#1515)

Obvious exits:

out

> wind

The box whirrs and humms. Suddenly, the box SPROINGS open, and a scruffy-looking orange cat SPRINGS out of the box!

The cat says "ACK!" and coughs up a hairball!

ack-in-the-box dropped hairball.

The cat is then pulled back into the box, which closes.

A slimy cat hairball splats onto the floor.

> l

Speedy's Room

You are in Speedy's Room.

Contents:

hairball (#1530d)

hairball (#1531d)

ack-in-the-box (#1515)

Obvious exits:

out

As you can see, every time you wind the ack-in-the-box, a slimy hairball is created, which drops onto the floor! If you were holding the ack-in-the-box, it would drop into your hand (and thus into your inventory).

This simple example illustrates the use of semaphores to make sure that the commands to create the hairball describe it, and set attributes on it are not preempted by another wind command, issued as the ack-in-the-box was working on the first wind command.

## Putting It All Together: Exploding Jawbreaker Example

Putting all the concepts together, this section will go through the programming of an exploding jawbreaker. The jawbreaker will have a command defined on it that will allow a player to eat it. When the player attempts to eat the jawbreaker, various random amusing things happen. The object will also have several user-defined attributes. These attributes store various variables, strings, and messages.

Here goes!

```
> @create jawbreaker
```

```
jawbreaker created as object #4545.
```

```
> @desc jawbreaker = It looks like a giant candy jawbreaker, about the size of a plum. Upon closer inspection, it seems to be made of sugar, nitroglycerin, [name(v(currate))]'s saliva, salt and various corn by-products. Type 'eat jawbreaker' to eat it!
```

Set.

```
> @fail jawbreaker = Type 'eat jawbreaker' to eat it.
```

Set.

This creates the jawbreaker and sets its description. Note the function call within the description. The currate attribute contains the DBREF of the player who last tried to eat the jawbreaker. This attribute is set by other commands later. The name() function uses the value in the currate attribute to generate the name of the player who last ate the jawbreaker.

```
> &comment jawbreaker = This is a trick jawbreaker that explodes when someone tries to eat it.
```

Set.

The comment attribute is simply a note to future programmers. It describes what this object does. On very complex objects, it is helpful to throw in a few comments to describe how the object works, so that other people can figure out what is going on!

Next, program the eat jawbreaker command:

```
> &chew jawbreaker = $eat jawbreaker:@emit [name(##)] picks up the jawbreaker and pops it into %p mouth!:@trigger me/rand[rand(3)];&lastate me=[get(me/currate)];&currate me=##
```

Set.

This sets the eat jawbreaker command and the actions to be performed when eat jawbreaker is typed by a player. There are four actions in the action list.

The first action displays a message to everyone indicating that the player put the jawbreaker in his or her mouth. Note how the %p pronoun substitution is used to print the appropriate possessive pronoun corresponding to the player.

The second action generates a random number from 0 to 2, using the rand() function. This number is appended to the string rand to generate a random attribute name, which is then triggered. One of three attributes will be triggered: rand0, rand1, or rand2.

The third and fourth actions in the action list set the `lastate` attribute to the last player who ate the jawbreaker, and the `currate` attribute to the player who is currently trying to eat the jawbreaker. These two attributes can then be used in later commands and actions.

Now, all there is to do is to define the three random actions that are triggered when the player eats the jawbreaker.

```
> &rand0 jawbreaker=@emit [name(v(currate))] closes [poss(v(currate))] eyes and
smiles from ear to ear as the sugar kick from the jawbreaker sets in. Ahhh. After a
few seconds of sucrose satisfaction, [name(v(currate))] removes the jawbreaker
from [poss(v(currate))] mouth.
```

This first random action is fairly harmless. Nothing bad happens to the player. The `currate` attribute is used to determine both the player's name, using the `name()` function, and the possessive form of the player (his, her, its), using the `poss()` function.

```
> &rand1 jawbreaker=@emit [name(v(currate))]'s eyes get real big as the fire-hot
super jalapeno pepper flavor of the jawbreaker takes effect! [poss(v(currate))]'s
eyes start to water as [subj(v(currate))] screams, "YOW THAT'S HOT!"
Suddenly, [name(v(currate))]'s head EXPLODES with a flash and a smell of hot peppers
and burnt sugar!! *BOOM* !!;kill [name(v(currate))]=100
```

This second random action is a bit more violent. It kills off the player, due to the flavorful effects of the jawbreaker, by performing a kill command on that player. The `subj()` function returns the subjective pronoun of the player.

```
>&rand2 jawbreaker=@emit [name(v(currate))] gasps, "I've been poisoned by
[name(v(lastate))]'s *gag* spit!" In a last-ditch attempt for revenge,
[name(v(currate))] throws the jawbreaker at [name(v(lastate))].! [switch (eq(her,
loc(v(lastate))), 1, The jawbreaker BONKS [name(v(lastate))] on the head and the
unstable nitroglycerides in the jawbreaker EXPLODE!, 0, Luckily, [name(v(lastate))]
is not around, and the jawbreaker merely ricochets off the wall and rolls to a stop
nearby.);kill [name(v(currate))]=100;kill [name(v(lastate))]=100
```

The last random action is somewhat whimsical. It tries to kill off both the current player and the player who last ate the jawbreaker. Note the `switch` statement in the action list. It tests to see if the player who last ate the jawbreaker is in the same room. If so, an appropriate message is printed as the jawbreaker hits that player. If not, the jawbreaker just rolls around to a stop. Finally, a few kill commands are performed. If the person who last ate the jawbreaker is not in the current room, that kill command will have no effect.

This exploding jawbreaker is amusing to play around with in a room full of people! Players who lose out and are killed by the jawbreaker can simply teleport back to the room, so there isn't much harm done.

Thanks to Jen "Tourmaline" Kleiman for providing the idea behind this example.

## Programming a MUSE

MUSE, a variation and elaboration of MUSH, is written in a language called TinyMUSE, another spin-off of the original TinyMUD. TinyMUSE was first implemented in a popular simulated environment called MicroMUSE. This on-line environment was first conceived in 1990 by programmer Stan Lim ("Jin"), who built the original environment in Larry Foard's TinyMUSH programming language. "Cyberion City," as it came to be called, is a beautiful vision of a utopian society in an orbital station high above the Earth, complete with talking robots, spaceships, and otherworldly adventures. As the years went by and the source code was refined and elaborated upon, the newer language became known as TinyMUSE, and a specialized client program, called TinyFugue (also commonly known as *tf*) was developed to interface with the MUSE during play. TinyMUSE worlds come in a variety of flavors, but the language has become a favorite of many educators because of its ease of use, and its firm but modifiable "social structure."

Almost all of the capabilities of MUSHes can also be found in MUSEs. MUSEs also have a few idiosyncracies all their own, such as a heirarchical player class system.

## Popular MUSEs

There are a few thematic MUSEs out there, but the granddaddy MUSE is MicroMUSE.

<i>Name</i>	<i>Machine Name</i>	<i>Internet Address</i>	<i>Port</i>
MicroMUSE	chezmoto.ai.mit.edu	18.43.0.102	4201
Rhostshyl	rhostshyl.cit.cornell.edu	128.253.180.15	4201
TimeMuse	murren.ai.mit.edu	18.43.0.179	4201
TrekMUSE	grimmy.cnidr.org	128.109.179.14	1701

These addresses are current as of August, 1994. Keep in mind that they may change.

MicroMUSE, a popular simulated environment based at MIT, was first conceived in 1990.

TinyFugue, the TinyMUSE client, is available via anonymous FTP to chezmoto.ai.mit.edu and may be found in the pub/muse directory as tf.33a5.tar.gz.

## MuseNet Worlds

There are a growing number of TinyMUSE worlds on the Net, of which those listed here represent merely a sample. A number of educational TinyMUSE worlds were incorporated into a single overarching structure known as MuseNet (the Multi-User Science Education Network); any of these worlds may be reached via telnet to any of the affiliated sites.

With the TinyFugue client, you can instantly open bamf portals between any of these worlds, allowing your character to toggle between worlds at a whim. To see a list of available worlds, simply type /listwords from within TinyFugue. To open a connection to another MuseNet world, type /world *world-name*, and then type ESC-B to bamf through the portal.

The current list of MuseNet worlds includes MicroMuse, BridgeMuse, MariCopa, CyberLib, and EcoMuse. MuseNet worlds can be reached via telnet at any of the following addresses:

musenet.bbn.com

micro.bbn.com

muse.usmee.maine.edu

micro.usmee.maine.edu

bridges.usmee.maine.edu

pico.usmee.maine.edu

femto.usmee.maine.edu

cyberion.usmee.maine.edu

nano.usmee.maine.edu

atto.usmee.maine.edu

## Things Specific to MUSEs

The MUSE environment allows for *classes* of programming capability to be distributed by the MUSE Directors, according to the talent and drive of individual players. The learning curve is therefore broken into a series of *plateaus*, allowing players to progress at their own speed. The following things are specific to MUSEs.

## Commands

There are a few common commands that MUSEs have, in addition to those described in the "Overview of Common Commands" section, earlier in this chapter.

The money command shows your current number of credits and the value of all your assets.

The news command displays the current newsfile, which you should check regularly.

## Classes

In TinyMUSE worlds, programming capabilities (powers) and virtual civic duties are conjoined rather formally in a hierarchic structure of roles: *Visitor*, *Citizen*, *Guide*, *Official*, *Corporation*, *Administrator*, and *Director* (the specific duties and powers of each class are determined by the Directors of any given MUSE, and may be set to affect all *lower-ranked* users, all *equally ranked* users, or *all* users). This system tends to ensure that those players entrusted with powerful characters act in a responsible and accountable manner.

Moving around in the virtual environment and playing with other people's creations is an enjoyable pastime, but the real action begins when you decide to become a MUSE Citizen. At this point, you receive a real character (having been known as guest up until now), and some building privileges. Your character, provided by the officials of the MUSE, will have a name and an associated password, and probably very little else.

The function

```
class(player)
```

can be used to access a player's class in MUSE programs.

You can also use functions that access player's powers:

```
controls(player1, player2, power)
```

returns 1 if *player1* has the specified power over *player2*. Otherwise it returns 0.

```
haspow(object, power)
```

returns 1 if the object has the specified power. Otherwise it returns 0.

## Zones

MUSEs have specialized rooms called *zones*. A zone is a group of rooms acting as if it were one big room. Messages that are sent to a room in the zone can be heard throughout the zone. For instance, you can create a long hallway, composed of three rooms, such as the West End of the hallway, the Middle of the hallway, and the East End of the hallway. Players can be located in any of these three rooms. Then, you can group the three rooms into a zone room. Certain messages that are sent to the zone can be heard by everyone in the hallway rooms. In this manner, you can create large rooms that seem to have a depth to them. You can create a ballroom or a large dance floor using a zone.

To link a room into a zone, use the command:

```
@zlink room = zone-object
```

And to emit a message to everyone in the zone, use:

```
@zemit zone-object = message
```

There are also a few functions you can use in MUSE code that relate to zones. The function

```
zone(object)
```

operates on a given object. If the object is a room, the room's zone is returned. If the object is an exit, thing, or player, the DBREF of the zone that the object is in is returned.

Additionally, the function

```
zwho(zone-object)
```

returns a list of the DBREFs of all the players in the zone, and the function

```
inzone(zone-object)
```

returns a list of all the rooms that have been z-linked to the zone.

## Interactive Narration: "Monster Island" Example

In an object-oriented environment, directing players where you want them to go can be tricky. If done inartistically, linearity can be easily overdone, making things seem like a slow-moving sequence of still images. The trick is to balance the decision to "opt out" of a storyline with various "lures in" to the storyline. It also doesn't hurt to make every room alive with incidental action.

The following environment is excerpted from Tod Foley's "Monster Island," a 1920's pulp adventure MUSE narrative with a fairly strong overarching plotline. The story begins in a dark, secluded bay...

```
Blackmoor Bay(#248Rpv)
```

The waters of Blackmoor Bay are a deep indigo. Dark waves lap the sandy shore in a mesmerizing rhythm, and a chilly seabreeze sends a shiver up your spine. Rumors claim that this Cove is home to a host of lost seafaring souls, sailors whose ships were destroyed by the merciless might of The Great Sea...

Contents:

```
ship(#252SeoK)
```

Obvious exits:

```
Shore Sea
```

A combination of locks were used to keep people from traveling straight out into the sea. The @fail message for this exit describes the biting wind and stinging spray. Although there are a few clues lying around on the shore, the objective is for players to get out to sea.

```
enter ship
```

```
ship(#252SeoK)
```

The ship is stocked and ready to sail, despite the fact that no crew can be found. The hull is old and the sails slightly torn, but she seems seaworthy enough. Care to set sail?

Obvious exits:

```
bay climb mast
```

Typing climb mast leads you to the crow's nest, where you can see the entire surrounding area (and perhaps discern another clue); it also allows you to jump straight into the cold waters of Blackmoor Bay, just for fun. The desire was to structure this adventure so that one party could come along fairly soon after the previous one, so the ship had to stay here. If you examine the code, you see that that's exactly what it does. The @va register waits to hear the magic words "set sail" to issue you the exit name:

```
ex here
```

```
ship(#252SeoK)
```

This dilapidated old vessel looks like she's weathered a few storms, but she's rigged and ready to sail. You see no movement onboard. Dare you enter?

```
Owner: Speedy Credits: 1
```

```
Type: Thing      Flags: sticky enter_ok opaque key
```

```
Zone: Universal Zone
```

```
Created: never
```

Modified: never

Idesc:The ship is stocked and ready to sail, despite the fact that no crew can be found. The hull is old and the sails slightly torn, but she seems seaworthy enough. Care to set sail?

Va:\$set sail:@PEmit %N=You take the helm, piloting the ancient vessel northward to the mouth of the Bay. The Sea looks especially rough for this time of year... What's your call - `sea' or `bay'?

Contents:

Speedy(#5Pevcn)

Home: Blackmoor Bay(#248Rpv)

Location: Blackmoor Bay(#248Rpv)

Exits:

bay(#372Ev)

sea(#371EDv)

climb mast;up;climb(#251Ev)

First we head out to sea...

sea

You boldly set forth into the fierce and mysterious Sea...

The Great Sea(#250RvJ)

You head out into The Great Sea, fierce winds blowing in your hair, salty spray hitting your face... you begin to fear for your safety as the sky darkens and the waves grow higher... you have no idea which direction you're facing anymore...

Obvious exits:

<I>sland <B>ay

After a second or two, we see...

You fear a storm is approaching...

And after another second or two...

The sky begins to draw dark as the clouds above gather in disturbing thickness...

The pressure's really getting to you by now, isn't it? Well, the clue in the crow's nest is a scrawled note reading "Ye better pray, matey." And, in true literal fashion, players who type anything beginning with p-r-a-y will be rewarded by the following line of code (from register @va of The Great Sea):

Va:\$spray\*:@PEmit %N=Your prayers have been answered, %N. The sky begins brightening as the clouds part, and a lovely rainbow arches overhead. You see a single white dove heading toward an island that wasn't there a minute ago... ;@tr me/vb

All in all, it really hasn't taken very much work—two rooms and two things—to get the players fairly immersed in the adventure. They've experienced a little thrill and a small miracle, and feel as though they're far from "home." At this point, we could allow players a chance to move on through other exits and travel to other realms. By linking exits to the rooms of fellow

MUSErs, the Sea becomes Great indeed. Once the players commit to debarking upon the island, however, the narrative picks up in full swing:

island

River Mouth(#373R)

The river is narrow, the water dark, and the smell is horrid. You push aside some of the hanging vines that obscure your view, and peer ahead. Overhead, several carrion-birds circle slowly, patiently waiting...

Obvious exits:

<S>wamp <J>jungle Sea

One easy way to create the feeling of "life" in environments like these is by using @wait commands in the @adesc line, like this:

```
Adesc:@Wait 5=@Emit A mutant rodent on the north bank is startled by your passage. It freezes, limbs locked, staring at you.;@Wait 15=@Emit A slimy tentacle suddenly erupts from the water. It grasps the small mammal, which quickly sinks its teeth into the attacker's squishy flesh. Thus released, the little rat-thing scampers away into the underbrush, and the tentacle retreats silently into the brackish water.
```

You can also make use of the rand() function to vary the local wildlife a little bit. After picking up a machete from the swamp to the north (you'll understand why in a moment), we'll move south into the jungle:

Jungle(#376R)

The vegetation is immense and unusual, and the sounds you hear around you are like no animals you can identify. You feel as though you are being watched by some lurking malevolence...

Obvious exits:

<E>ast <N>orth

## After a few seconds, we see...

The jungle is getting too thick to push your way through... Branches tear your clothes and scratch your face... You have to stop and rest for a minute...

Now there are two moves players can make. The first is to use the machete we got from the swamp (note that if the player came here first, this area would seem pretty useless, causing them to go across the river to the swamp room, where they'd find the machete). Used here, the machete will hack a way through the trees for us. But there's another clue hidden in the words above; let's see what happens if we type rest:

rest

You settle back against a tree and catch your breath. This jungle-cutting is no easy feat! The smooth curve of the trunk behind you feels comfortable, and you close your eyes, drifting into a much-deserved slumber.

...

You dream of a better life...

...

One with much less excitement and regular paychecks...

...

Dream on...

You are awakened by a deafening roar from the east. What could you have been thinking? You scramble to your feet and scan

the surrounding area, looking for anything you'd hope not to see...

Scary! But if the player waits just a little while longer, they'll be rewarded again:

The roar is not repeated, and you cannot pinpoint its

source, but there's a tall tree nearby with a man's jacket stuck in the highest branches.

Huh -- what's this? Only one way to find out:

climb tree

You pull yourself to the top of the tall tree, trying not to look down too often. At last you reach a point where you can climb no further.

Top of Tall Tree(#379R)

You are hanging in a precarious position from the top of one of the tallest trees on the island. From here you can see your ship and most of the river valley. You notice that many of the ancient trees are bent or broken at odd angles...

Contents:

Jacket(#382S)

Obvious exits:

down

Where's that jacket?

look jacket

This khaki jacket has been shredded by some amazingly powerful attacker. It's covered with dark brown stains, and seems to have something in the left pocket...

take jacket

You can't pull the jacket loose from the branches, but you manage to reach into the pocket... It feels like a small notebook...

take notebook

You grab the notebook quickly, and a few pages scatter to the wind.

read notebook

Small Notebook(#383SD)

It's the battered journal of an adventurer named (something illegible) Westcott. Most of the pages are either missing or too badly damaged by the elements to read, but the last entry is fairly clear. "Most of my crew have deserted me," writes Westcott, "and I cannot say I blame them, for it is true what they say of this island—it is inhabited, nay, RULED, by a gigantic reptilian creature. I have therefore decided to name this place Monster Island."

You get the picture. We'll leave off right there, just to keep you wondering.

## MUCK, MUSH and MUSE Reference

This section provides a detailed reference to MUCKs, MUSHes, and MUSEs. It lists at-commands, common functions, flags, and other mud commands and structures.

## Concept Reference

As an aid for your memory, this section summarizes some basic mud concepts.

## Control

In general, you can examine, modify, or use any object you control, and you control only the objects you own (unless you're a Wizard or an Official). The single exception to this rule is an unlinked exit; anyone can link to it, causing the exit to be @chown'ed to them, and effectively stealing the exit from its original owner. Builders should be aware of this.

## Costs

Certain commands cost credits to perform.

<i>Command</i>	<i>Cost</i>
page	1 credit each
@dig	1 credit each
@link	1 credit if you did not own the exit, plus 1 to the previous owner
@open	1 credit, plus 1 more if the exit is linked at the same time
@create	10 credits or more (up to 505 credits)
@wait	10 credits each
@find	10 credits each
@search	10 credits each
@announce	250 credits.
@trigger	Anywhere from 1/64th to 1/16 of a credit
@force	Anywhere from 1/64th to 1/16 of a credit

Certain commands demand deposits, which are refunded when the object is destroyed.

## Credits

In the enlightened words of Moulton (Barry Kort), "Credits are really chits for CPU cycles." In other words, don't take anything about them too personally—they're not even really *fake* money. You receive an allowance of credits every day you connect with the mud. You may also borrow money, or sell the objects you create.

## Drop-To

When an @link is used on a room, it also creates a drop-to location. From then on, any object dropped in the room will go to that location (unless the room is STICKY). If the room is STICKY, the drop-to will be delayed until the last person has left.

## Enactor

For any action that occurs, the enactor is the object that causes a given action to occur. If an object initiates an action by an @force or @trigger, it is the enactor of the specified action. If one of your objects initiates an action by an @asucc, @afail, or @aenter, you are the enactor; the action will occur just as if you had typed it yourself. Interactive Design Note: When an action begins running, the name of its enactor is plugged into the %N and %n variables.

## \$ and ! Events

These extremely useful events enable you to define new verbs and "automatic" commands. Stored in any register of any class of object, they wait until their specific text pattern comes along, and then they execute their listed actions.

*\$ events* wait until their pattern is matched by any typed command (by any player in the room):

```
$pattern:action1[;action2][;...]
```

*! events* are similar, except that they listen for "spoken" words instead of typed commands:

```
!pattern:action1[;action2][;...]
```

In either case, the pattern may include wild cards. If it does have wild cards, the strings matching them is plugged in as temporary variables, and may be referenced in the action part of the command. The first such variable will be called %0 or v(0), the second %1 or v(1), and so forth.

## Exits

Exits are created by the @open command, but they will only be automatically linked to a given room if this is specified. If unspecified (or if the exit is created from within another type of object), you must then use an @link command to link the exit to where it goes. If you ever find yourself in a place with no obvious exits, you may be inside a thing. The leave command will get you out.

## Failure

Failure has different meanings in different contexts. You fail to use a thing when you cannot take it (due to locks, for instance). You fail to use an exit when you cannot go through it. You fail to use a room when you cannot look at it.

## Home

To teleport instantly to your home, type home at any time. To set a new one, use the @link command (the target room must either be yours or have its LINK\_OK flag set). All players and things have homes; this is the place where STICKY objects go when you drop them. For things, the default home is the room in which they were created. To @link an exit that sends players back to their own homes:

```
> @link exit-name=home
```

## Linking

You can only @link to rooms you control, and rooms that are set LINK\_OK. An @link performed on a thing sets the object's home.

## me

You can use the term me to refer to your character.

## Names

To have an object refer to a players' name, use the variable %N in your command statement. If, for some strange reason, you wish not to capitalize the name, use %n instead.

## NULL Exits: Bogus Commands

It's easy to come up with "bogus commands" by creating unusable exits. Here's an example that seemingly grants the power of flight:

```
> @open fly
```

```
> @link fly=here
```

```
> @lock fly=me&!me
```

```
> @fail fly=You suddenly rise, flying high above the ground!
```

```
> @ofail fly=rises up overhead, flying through the air!
```

The exit was linked so that no one could steal it, and locked so that no one could successfully use it (since it's impossible to be both "me" and not "me"). Since nobody can go through this "exit," it always fails. The @fail message is displayed to the player, and the @ofail message (preceded by the player's name) is shown to everyone else.

## Numbers

All objects in the mud are known by a database reference number, called a DBREF for short. The DBREF of any object you own shows up whenever you look at the object; others do not see it. To have an object refer to the DBREF of another object, use the variable %# in your command statement.

## Objects

There are only four types of objects in muds: *rooms*, *exits*, *things*, and *players*.

The first letter following an object's ID number indicates its type (*R*, *E*, *T*, or *P*). *Rooms* are objects that can contain other objects and linked exits. *Exits* are the means by which objects move between rooms and things. *Things* are inanimate objects that can be carried. *Players* are animate objects that can move and carry (contain) other objects.

## Pronouns

Mud objects can be made to refer directly to players, by pronoun as well as by name. This is one of the areas where normal English grammar becomes a little tricky to program, so a number of special variables has been set aside as pronoun-substitution characters. There are five "sexes" available in most mud worlds: male, female, neuter, plural, and "spivak" (a genderless system that lops off S's and H's). Objects without sex of any kind will have their names used instead of pronouns; but this makes for inelegant narrative. For instance: "Speedy shines the apple on Speedy's chest and bites into it with a gleam in Speedy's eye." Bleagh. The results of the shown variables for all five sexes are as follows:

<i>Variable</i>	<i>Type</i>	<i>Example</i>
%S, %s	Subjective pronoun	he, she, it, they, e
%O, %o	Objective pronoun	him, her, it, them, em
%P, %p	Possessive pronoun	his, her, its, their, eir
%A, %a	Absolute possessive pronoun	his, hers, its, theirs, eirs

## Stack

The stack for any object consists of the contents of the numbered stack variables: v(0) through v(9). You can set these contents in various ways; they may be edited directly, entered as actions in an @trigger command, or replaced by wild card substitutions.

In MUCK, the stack is the an actual stack of things. You create MUF programs that manipulate the stack.

## Substitutions

Several types of % variables (variable substitution characters) have already been discussed (see "Names," "Numbers," and "Pronouns"). There are a few others reserved for use, which make programming quite a bit easier. The values of an object's V-registers may be accessed by the variables %va through %vz.

## Success

Success has different meanings in different contexts. You succeed in using a thing when you take it. You succeed in using an exit when you go through it. You succeed in using a room when you look around.

## V-Registers

V-registers are the main means of programming objects. Each object has 10 numbered stack variables, v(0) through v(9), and 26 V-registers, named v(va) through v(vz). Each V-register can hold a number, a character string, or an executable command.

## Verb Commands

Most verb commands come in three forms:

verb (what the enactor sees)

overb (what others in the room see)

averb (the action triggered when the verb event happens)

## Tips For Programmers

You can easily spend a few years using the commands given in the last section, and create thousands of masterpieces of textual art (and many have done just that!). There are, however, a good number of advanced features that prove invaluable when it comes to fine-tuning the behavior of your mud objects, both as individual units and—even better—as interconnected parts of an *interactive environment*. This section provides a reference for many of those features and functions.

Your first interactive environment may consist of nothing more than one densely coded room full of chatty objects, or you may choose to bite off a dozen connected rooms. In either case, you still have a little way to go.

When you create a room and place a couple of objects in it, you tend to begin developing a theme of sorts. This theme may be as loose as "fantasy," or as tightly focused as "a rowdy roadside bar located in a traveling tent, sometime during the early seventies." Follow these thematic threads and pick out the most important items, places, and characters. As you continue creating these initial elements of your environment, certain objects will, by their very nature, imply others. Be careful—these other objects will imply others beyond them, and the process may begin to exponentiate beyond reckoning (resulting in nothing but a bad case of programmer's shock!). If you don't develop things a little at a time, you're bound to get trampled under your own ideas. This can lead to hastily constructed mud environments that are of poor quality.

So as not to draw the ire of the wizards, you might stick to a small series of related rooms at first, adding objects and attributes a little bit at a time. Having kept a list of related objects and ideas, begin adding one or two objects at a time into your area. After creating and describing these new objects, go around your entire area again and add a level of detail.

Here are some guidelines to help:

1. Limit yourself to a certain number of initial objects, say 10 or 12.
2. @create them and @desc them, and @link your exits.
3. Add @ofails, @adescs, and such to enhance the narrative.
4. Jot down any other ideas you have on a nearby piece of paper.
5. Think about how these objects might interact with players, and with each other.
6. Take a break! Then add more to your creation in an iterative fashion.

It is also important to remember that all your objects take up space in the database, so be considerate and thoughtful when you create new areas and objects. Avoid creating low quality objects that just clutter up the database. Strive for interesting rooms and innovative objects that mesh with the theme and tone of the mud.

## At-Commands Reference

In this section, the term *action* is frequently used. Any basic command or at-command may be used as an action. These actions occur sequentially if separated by semicolons. For example:

```
> @wait 5=say Hi!;!@Emit %N sounds funny.;say Do you think I sound funny?
```

will wait for 5 seconds and then display:

```
Speedy says "Hi!"
```

```
Speedy sounds funny.
```

```
Speedy says "Do you think I sound funny?"
```

Note that you may have to enclose your action list in braces if your messages contain commas or semicolons.

### At-Commands

Some commands in this section are only available on certain types of muds. These commands are listed as such. Certain commands may also only be available to wizards or MUSE officials, and they are likewise denoted.

```
@aahear object=actions
```

At All Hear. Sets the actions triggered when the @listen string is matched by a pose or utterance from the object itself or from a player.

MUSH and MUSE only.

```
@aclone object=actions
```

Sets the actions triggered by the newly cloned copy when the object is cloned.

MUSH and MUSE only.

```
@aconnect object=actions
```

Sets the actions triggered when someone connects into the mud. On a room or object in the room, it's triggered by connection within the room.

MUSH and MUSE only.

```
@addparent object=parent
```

Gives the object a new parent. The parent must be controlled by you or be set BEARING. The object inherits attribute definitions from its parents. It also inherits attribute values, if the attributes in question have the INHERIT option set.

MUSE only.

```
@adesc object=actions
```

Sets the actions triggered when the object is looked at.

MUSH and MUSE only.

```
@adisconnect object=actions
```

Sets the actions triggered when someone disconnects from the mud. On a room or an object in the room, it's triggered by a disconnection within the room.

MUSH and MUSE only.

```
@adrop object=actions
```

Sets the actions triggered when the object is dropped.

MUSH and MUSE only.

```
@aefail room|object|player=actions
```

Sets the actions triggered when someone fails to enter the room, object, or player.

MUSH and MUSE only.

```
@aenter room|object|player=actions
```

Sets the actions triggered when the room, object, or player is entered.

MUSH and MUSE only.

```
@afail object=actions
```

Sets the actions triggered when an attempt to use the object fails.

MUSH and MUSE only.

```
@ahear object=actions
```

Sets the actions triggered when the object's @listen string is matched.

MUSH and MUSE only.

```
@aleave room|object=actions
```

Sets the actions triggered when a player or other object leaves the specified room or object.

MUSH and MUSE only.

```
@alfail room|object=actions
```

Sets the actions triggered when someone fails an attempt to leave the room or object.

MUSH and MUSE only.

```
@alias player=alias
```

Sets the alias by which to reference the player.

MUSH and MUSE only.

```
@allquota number
```

Sets the builder's quota for all players to the number entered. Individual players can check their own quotas with the @quota command (see @quota later in this section).

Wizards and Officials only.

MUSE only.

```
@amhear object=actions
```

Sets the actions triggered when the object's @listen string is matched by a pose or utterance from the object itself.

MUSH and MUSE only.

```
@amove object[=message]
```

Sets the message emitted if the object moves by any means.

MUSH and MUSE only.

```
@announce message
```

Broadcasts the specified message to every player connected. Costs 250 credits. Players can block out announcements by setting themselves NO\_WALLS. Abbr.: @ann.

MUSE only.

```
@apage player=actions
```

Sets the actions triggered when you are on-line and receive a page.

MUSE only.

*@apay object=actions*

Sets the actions triggered when someone or something gives credits to the object.

MUSH and MUSE only.

*@apemit object=actions*

Sets the actions triggered when the object gets an @pemit.

MUSH and MUSE only.

*@asfail room=actions*

Sets the actions triggered when someone fails to use an @slock.

MUSE only.

*@asucc object=actions*

Sets the actions triggered when the object is successfully used.

MUSH and MUSE only.

*@aufail object=actions*

Sets the actions triggered when someone fails to use the object due to @unlock.

MUSH and MUSE only.

*@awhisper object=actions*

Sets the actions triggered when the object receives a whisper.

MUSH and MUSE only.

*@broadcast message*

Broadcasts the given message throughout the mud. Nonblockable.

MUSE only.

**Wizards and Officials Only.**

*@boot player*

Disconnects the player from the game.

Available on MUCK, MUSH and MUSE.

**Wizards and Officials Only.**

*@charges object=#charges*

Limits the number of times an action can be used.

MUSH and MUSE only.

*@chown object=player*

Changes the ownership of the object to the specified player, turns on the object's HAVEN flag, and turns off the object's INHERIT flag.

Available on MUCK, MUSH and MUSE.

*@chownall player1=player2*

Transfers the ownership of all of player1's possessions to player2.

MUSH and MUSE only.

### **Wizards and Officials Only.**

`@clone object`

Creates a duplicate of the object in the current room.

MUSH and MUSE only.

`@columns #columns`

### **Sets the number of columns that WHO prints. The default is 80.**

MUSE only.

`@cost object=#credits`

Sets the number of credits that must be given to the object to trigger @pay, @opay, and @apay.

MUSH and MUSE only.

`@cpattr new-object/new-attribute=old-object/old-attribute`

Copies an *attribute definition* from the old object to the new object. The names of the two attributes used need not be the same.

MUSE only.

`@create object-name[=#credits]`

Creates a new object whose name is the specified *object-name*. The command costs #*credits* or 10 credits, whichever is greater. The object's "actual" value will become (*#credits* / 5) - 1.

Available on MUCK, MUSH, and MUSE.

`@dbck`

Shows any disconnected rooms and clears the free list.

MUSE only.

`@dbtop item`

Shows the top list for the specified item in the database. Do @dbtop (without an argument) to see what objects are available to list.

MUSE only.

`@decompile`

Outputs a series of commands that can be fed back into the mud to redefine and reset all of an objects' registers and flags. This command can be used to save an object to disk, edit it, and read it back in via the TinyFugue /quote macro. When using @decompile for this purpose, be sure to set /wrap off in TinyFugue so that lines are not broken every 80 characters. @decompile displays several commands, including @set (to set flags), @defattr (to define attributes), and @nset (to set attribute values).

MUSH and MUSE only.

`@defattr object/attribute[=option1 option2 ...]`

Creates or modifies a user-defined attribute on the object. Any options specified will dictate the behavior of the attribute. The new attribute may be set by typing: @set *object=attribute:value*, or with the usual shorthand: @*attribute object=value*. For example, @defattr Book/title=osee defines an attribute called title which anyone can see upon doing an examine command (this is the meaning of osee--see @osee), and @title Book=Muse Tutorial places the words Muse Tutorial there. Now the function [get(book/title)] returns Muse Tutorial.

MUSE only.

`@delparent object=parent`

Removes the parent from the object. The parent must be controlled by you, or set BEARING.

MUSE only.

`@describe object [=text]`

Sets the text seen when the object is looked at. Abbr.: @desc.

Available on MUCK, MUSH, and MUSE.

`@destroy object`

Sets the object "non-protected" and returns the creator's investment. This command may be aborted by an @undestroy within 10-15 minutes.

MUSH and MUSE only.

`@dig room-name [ ; alternate-names ] [=exits [ , return-exits ]]`

Creates a room, giving it the specified room name. The optional portions allow alternate names and auto-linked exits.

Available on MUCK, MUSH, and MUSE.

`@does object=actions`

Like a v-register, but removes 1 charge from the object when triggered.

MUSE only.

`@doing message`

Sets the player's @doing message, which is displayed in the WHO list.

MUSH and MUSE only.

`@doomsday`

Set by @destroy, @undestroy, @protect, and @unprotect, doomsday is a number in xtime() format, specifying the system time before which an object should not be destroyed. Objects with the GOING flag set will be destroyed during the first @dbck after the time given.

MUSE only.

**Only Officials may set the doomsday attribute directly.**

`@drop object [=message]`

Sets the message seen when someone drops the object. If entered without a message, it clears any existing message.

Available on MUCK, MUSH, and MUSE.

`@dump`

Dumps the database for storage (vital to the continued survival of the mud).

Available on MUCK, MUSH, and MUSE.

`@edit object/attribute={old-string}, {new-string}`

Replaces the first occurrence of the old-string by the new-string. The attribute can be any attribute on the object that holds a string value (for example, @desc, @succ, @adrop, @listen, @ahear, @va, @vb, and so forth). If the strings contain only alphabetic characters, the curly braces may be omitted.

Available on MUCK, MUSH, and MUSE.

`@efail object=actions`

Sets the message shown to a player who fails to enter the object.

MUSH and MUSE only.

`@elock object=lock`

Sets the enter lock for the object. Only players or things satisfying the lock may then enter the object.

MUSE only.

`@emit message`

Emits the message to everything in the room. The message can't start with a player's name (in order to prevent spoofing).

MUSH and MUSE only.

`@enter object[=message]`

Sets the message seen when a player enters the object.

MUSH and MUSE only.

`@entrances room|object`

Lists all the entrances into the room or object. You must control the room or object.

Available on MUCK, MUSH, and MUSE.

`@fail object[=message]`

Sets the message seen when an attempt to use the object fails. If entered without a message, it clears any existing message.

Available on MUCK, MUSH, and MUSE.

`@find [name]`

Displays the name and number of every object you control whose name matches the specified name. Costs 10 credits.

Available on MUCK, MUSH, and MUSE.

`@force player|object=actions`

Forces the specified player or object to perform the actions, as though the player or object entered the actions itself. Objects you own can `@force` you if set INHERIT.

Available on MUCK, MUSH, and MUSE.

**Only Wizards or Officials may `@force` players.**

`@foreach list=actions`

Executes the actions (which may be a semicolon-separated list of commands) once for each item in the list (whose elements are separated by spaces). If a %0 occurs in the actions, `@foreach` substitutes each element of the list, in order, for the %0. Since the %0 is evaluated each time the command passes through the command queue, you will need several %'s if the %0 is embedded in function calls.

MUSE only.

`@getalias player`

Tells you the player's alias, if any.

MUSE only.

`@giveto player=#credits`

Privately gives an amount of money to the player.

MUSE only.

`@halt object`

Stops a process or a runaway machine. To stop one particular object, type: `@force object=@halt`.

MUSH and MUSE only.

@hide

Removes you from the WHO list this session. Costs 10 credits.

MUSE only.

@huhto *player*[=*E-Mail-Address*]

Initiates or terminates the automatic logging of failed commands attempted by players in rooms owned by player. The log is periodically mailed to the specified E-Mail-Address. The use of HuhLogs enables a Builder to improve the choice of exit-names and trigger verbs in the Builder's realm, in order to better respond to commands most frequently attempted by visitors.

MUSE only.

**Wizards and Officials only.**

@idescribe *object=message*

Sets the message seen when the object is entered or looked at from the inside.

Available on MUCK, MUSH, and MUSE.

@leave *object*[=*message*]

Sets the message shown to a player upon leaving the object.

MUSH and MUSE only.

@lfail *object*[=*message*]

Sets the message shown to a player who fails to leave the object.

MUSH and MUSE only.

@link *object=room*

For things and players, makes the specified room Home. For rooms, makes the specified room the drop-to room. For exits, makes the specified room the exit's target room (the exit must be unlinked and in your control, or LINK\_OK). Costs 1 credit.

Available on MUCK, MUSH, and MUSE.

@listen *object=string*

Listens for the given string to trigger @ahear, @amhear, and/or @aahear. If the string is matched, the object's contents also hear the message.

MUSH and MUSE only.

@llock *object=lock*

Sets the leave lock for the object. Only players or things satisfying the lock will be able to leave the object.

MUSE only.

@lock *object=lock*

Sets the lock on the object. Only players or things satisfying the lock will be able to "succeed" with object (i.e., pick up a thing, go through an exit, trigger @succ, @osucc, @asucc registers, and so forth).

Available on MUCK, MUSH, and MUSE.

@move *object*[=*message*]

Sets the message shown to the object itself when it moves by any means.

MUSH and MUSE only.

@name *object=new-name* [*password*]

Changes the object's name. For a player, it requires the player's password.

Available on MUCK, MUSH, and MUSE.

*@necho message*

Shows a message to you only, but also allows n-phrasing, so you can type multiple spaces and other characters that don't normally come out with *@echo*. If this command is used in an attribute, it still must be *@nset*.

MUSE only.

*@nemit message*

Emits the given message, but does not parse its argument. This can be useful for emitting strings with spaces, brackets, or parentheses in them, without worrying about functions being evaluated or spaces being removed.

MUSE only.

*@newpassword player=password*

Changes the player's password. The player is informed of the change. If you forget your password, log on as a guest and talk to a Wizard or an Official.

Available on MUCK, MUSH, and MUSE.

**Wizards or Officials only.**

*@npage player=message*

Pages the player without parsing the message.

MUSE only.

*@npemit object=message*

Emits the specified message to the player, but does not parse its argument. Useful for emitting strings with spaces, brackets, or parentheses in them, without worrying about functions being evaluated or spaces being removed.

MUSE only.

*@nset object=attribute:actions*

Sets the attribute to the given actions, but does not parse its argument. May be used for setting attribute values with leading or embedded spaces, or with brackets or parentheses, without worrying about functions being evaluated or spaces being removed.

MUSE only.

*@nuke player*

Recycles the player's character, after which it no longer exists.

MUSE only.

**Wizards and Officials only.**

*@oconnect object=message*

Sets the message shown to everyone but the specified object, when a person connects to the mud.

MUSE only.

*@odescribe object[=message]*

Sets the message that is seen by the other players in the room when a player looks at the specified object. The message is prefaced by the name of the triggering player. Without a message, the command clears any existing message. Abbr.: *@odesc*.

MUSH and MUSE only.

*@odisconnect object=message*

Sets the message shown to everyone but the specified object when a person disconnects from the mud.

MUSE only.

`@odrop object [=message]`

Sets the message that is seen by the other players in the room when a player drops the specified object. The message is prefaced by the name of the triggering player. If the object is a room, this sets the message that is displayed to other people when someone enters the room. Without a message, it clears any existing message.

MUSH and MUSE only.

`@ofail object [=message]`

Sets the message shown to the other players in the room if a player fails to enter the specified object.

MUSH and MUSE only.

`@oemit object=message`

Sets the message to be emitted to everything in the room except the object itself.

MUSH and MUSE only.

`@oenter object [=message]`

Sets the message that is seen by the other players in the room when a player enters the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message.

MUSH and MUSE only.

`@ofail object [=message]`

Sets the message that is seen by the other players in the room when a player fails an attempt to use the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message.

Available on MUCK, MUSH, and MUSE.

`@oleave object [=message]`

Sets the message that is seen by the other players in the room when a player leaves the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message.

MUSH and MUSE only.

`@olfail object [=message]`

Sets the message that is seen by the other players in the room when a player fails to leave the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message.

MUSH and MUSE only.

`@omove object [=message]`

Sets the message seen by all other objects in the room when the specified object moves by any means.

MUSH and MUSE only.

`@opay object [=message]`

Sets the message that is seen by the other players in the room when a player or thing pays the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message.

MUSH and MUSE only.

`@open direction[;other-directions] [=#room]`

Creates an exit in the specified direction(s). If a room number is specified, the exit is linked to that room. Otherwise, the exit remains unlinked. Anyone may use `@link` to specify where an unlinked exit leads. Costs 1 credit, plus 1 credit more if the room number is also specified.

Available on MUCK, MUSH, and MUSE.

`@osfail room=message`

Sets the message shown to everyone in the room, except the enactor, when the enactor fails to use an @slock.

MUSE only.

`@osuccess object [=message]`

Sets the message that is seen by the other players in the room when a player or thing succeeds in using the specified object. The message is prefaced by the name of the triggering player. Without a message, it clears any existing message. Abbr.: @osucc.

@osucc.

Available on MUCK, MUSH, and MUSE.

`@password old-password=new-password`

Changes your password. Abbr.: @pass.

Available on MUCK, MUSH, and MUSE.

`@pay object [=message]`

Sets the message shown to the player who pays the specified object. Without a message, it clears any existing message.

MUSH and MUSE only.

@pbreak

Shows list of players, broken down into classes.

MUSE only.

**Wizards and Officials only.**

`@pemit player=message`

Emits the message to a specified player. The message may not start with the name of a player or of an object in the room.

MUSH and MUSE only.

@poof object

Teleports the object into itself and then destroys it (if possible). The object disappears from the scene immediately, though it is not actually destroyed for several more minutes.

MUSE only.

@poor

Resets all players' credits to 0.

MUSE only.

**Wizards and Officials only.**

`@powers player|thing`

Lists the authorized powers of the player or thing.

MUSE only.

@protect object

Sets the @doomsday attribute on the object to -1, in order to keep the object from being destroyed with @destroy.

MUSE only.

@ps

Lists the queued commands that have not yet been executed.

Available on MUCK, MUSH, and MUSE.

`@quota`

Displays your current builders' quota.

MUSH and MUSE only.

`@quota *player=#quota-amount`

Resets the player's builders' quota to the specified quota amount.

MUSH and MUSE only.

**Wizards and Officials only.**

`@recycle object`

Recycles the object, which saves space in the database. Remember your cyber-ecology!

Available on MUCK, MUSH, and MUSE.

`@relink exit=#room`

Relinks the exit to a different room number.

MUSE only.

`@remit object=message`

Emits the message to the contents of the object. The object is generally a room.

MUSE only.

`@runout object=actions`

Sets the actions triggered when the object's charges reach 0.

MUSH and MUSE only.

`@rwho start|stop|status`

Starts, stops and shows the status of the RWHO server. When stopped, the RWHO command will not operate.

MUSE only.

`@search [player] [class=restrictions]`

Searches the mud database and lists the objects that meet the player's search criteria. If the player argument is supplied, only objects owned by that player are listed. If the class argument is supplied, only objects of a certain class are listed. Costs 10 credits. The command `@search flags=RHD` lists all rooms (the R flag) set HAVEN and DARK (the R and H flags). The command `@search type=rooms` lists all rooms owned by the player.

MUSH and MUSE only.

`@searchobj object=text`

Searches an entire object and its attributes for the specified text. It then lists out each matching attribute. You may include wild card characters.

MUSE only.

`@selfboot`

Automatically performs an `@boot` on you three times, to clear up double logins into the mud. This means that you and your doubles (only) will be disconnected. Afterwards, you can reconnect to the mud.

MUSE only.

`@set`

Used to set attributes and flags on objects. May be used in various ways:

```
@set object=flag
```

Sets the specified flag on the object.

```
@set object=! flag
```

Resets the specified flag on the object.

```
@set object=attribute:value
```

Sets an attribute value on the object.

```
@set object=! attribute:value
```

Resets an attribute value on the object.

```
@attribute object=value
```

Short form of: @set *object attribute:value*.

```
@new-attribute new-object=old-object/old-attribute
```

Copies an attribute from one object to another object.

Available on MUCK, MUSH, and MUSE.

```
@sex player=gender
```

Sets the gender of the player. The player's gender is used for pronoun substitution. Options: unassigned, male, female, neuter, spivak, plural. The default is unassigned. Pronouns cannot be used in reference to players unless their gender is set to one of these choices. Visible to all.

MUSH and MUSE only.

```
@sfail room=message
```

Sets the message shown only to the player who fails to use the @slock. The player's name is not prefixed to the message.

MUSE only.

```
@shutdown
```

Shuts down the mud.

Available on MUCK, MUSH, and MUSE.

### **Wizards and Officials only.**

```
@slock room=arguments
```

Speech lock. Sets the speech lock on the room, where only the players specified in the arguments may speak in the room. The owner may talk in the room, and those with the power to spoof, even if they are not on the list. This command only affects rooms. The room that the @slock is on must be set AUDITORIUM: @set room=A.

MUSE only.

```
@startup object=actions
```

Sets the actions to be performed in the event that the mud crashes and gets rebooted. By using @startup, you can retrigger objects that need to be running continuously.

MUSH and MUSE only.

```
@stats [player]
```

Displays all the objects in the game broken down by types.

Available on MUCK, MUSH, and MUSE.

**Officials can supply a player name to count only objects owned by that player.**

```
@success object [=message]
```

Sets the success message for the specified object, which is displayed to the player whenever a player successfully uses the object. Without a message, it clears any existing message. Abbr.: @succ.

Available on MUCK, MUSH, and MUSE.

```
@sweep
```

Lists all the objects and players that are listening in the room you are currently in, as well as in the objects you are carrying. In these listings, player denotes a connected player that hears all occurrences in the room, puppet denotes a puppet belonging to a connected player, relaying all occurrences in the room to the owner, messages denotes an object that is listening for specific occurrences in the room, and commands denotes an object waiting for a specific command.

MUSH and MUSE only.

```
@switch
```

Evaluates a conditional expression and performs different actions based on the results of that evaluation. May be used in two ways:

```
@switch condition={pattern1},{yes-actions},{no-actions}
```

Corresponds to the IF-THEN-ELSE programming structure.

```
@switch  
condition={pattern1},{yes-actions1},{pattern2},{yes-actions2},...,{default-actions}
```

Corresponds to the CASE or SWITCH programming structure.

MUSH and MUSE only.

```
@teleport thing=#room
```

```
@teleport #room
```

Teleports the given thing to the specified room. If the thing is omitted, the command teleports you to the specified room. You must own or control the thing or its current location. You can only teleport objects into rooms or objects you own or that are set JUMP\_OK. If the target room has a drop-to, the object will go to the drop-to room instead. Abbr.: @tel.

Available on MUCK, MUSH, and MUSE.

**Officials can teleport objects into players' inventories.**

```
@trigger
```

Passes control and data (on the stack) between items. If you create attributes that are triggered by other commands or actions, this command can be used to trigger them. Many tricky things can be done with simple command combinations. Abbr.: @tr.

MUSH and MUSE only.

```
@tz player|object=#hours:y|n
```

Sets a player's time zone. The #hours is a numerical value representing the number of hours the player's time is ahead of or behind Greenwich Mean Time (GMT) (also referred to as Universal Coordinated Time (UTC), or London Time). The y or n refers to whether or not U.S. style Daylight Savings Time is used in your area. Objects that do not have a @tz set on them will use their owner's @tz. However, a player may set the @tz register on any object individually. If neither the object nor the object's owner has a @tz set, the mud will default to the local time at the mud's server site.

MUSE only.

```
@ufail object=actions
```

Sets the message shown to a player who fails to use an object due to an @unlock.

MUSH and MUSE only.

```
@unlock object=lock
```

Sets the Use Lock for the object. Only players or things satisfying the lock will be able to trigger \$ and ! events on the object.

MUSE only.

```
@undefattr object/attribute
```

Removes the specified attribute definition from the object, if it was set with @defattr. If object has children with the attribute, they lose the definition and any stored values, so use this command with great caution.

MUSE only.

```
@undestroy object
```

Aborts an @destroy process on the object if entered within 5-10 minutes of the original @destroy command.

MUSE only.

```
@unhide
```

Restores your name in the WHO list. Players are reset to @unhide each time they connect.

MUSE only.

```
@unprotect object
```

Removes the @doomsday attribute on the object, allowing it to be @destroyed.

MUSE only.

```
@unlink exit
```

```
@unlink here
```

Removes links from exits. The first version shown removes a link from the specified exit. The second removes the drop-to on the room. Be careful, though; anyone can relink an unlinked exit, thereby becoming its new owner (you will, however, be reimbursed for your 1 credit).

MUSH and MUSE only.

```
@unlock object
```

Removes the lock on the object.

MUSH and MUSE only.

```
@va object=actions
```

```
@vb object=actions
```

```
.
```

```
.
```

```
.
```

```
@vz object=actions
```

Sets the v-register on the object to the specified actions. Every object has 26 built-in registers, va through vz. Anything may be stored on a v-register. If a v-register contains a semicolon-separated list of actions (for example, @va *object=action1;action2;action3*), these actions will be executed in order when the register is triggered.

MUSH and MUSE only.

```
@wait #seconds=actions
```

Queues actions. The actions are placed on the queue and are executed no earlier than #seconds from the time it is queued. The actions may be a list of commands in curly braces.

MUSH and MUSE only.

`@whereis player`

Displays the location of the specified player. If you want to remain unlocatable, set yourself DARK. The located player receives a message informing him or her that you have successfully or unsuccessfully located them.

MUSE only.

`@who_flags object=flags`

Controls the output of WHO commands. Capital letters indicate "full" output; lowercase letters indicate "truncated" output:

A, a = Alias

C, c = Class

F, f = Flags (h: hidden, H: haven, N: set NO\_WALLS)

I, i = IdleTime

N, n = Name

O, o = OnTime

H, h = Hostname.

The default is NfoiC.

MUSE only.

`@who_names list-of-names`

Restricts the WHO search and display to the specified names.

MUSE only.

`@wipeout player`

Destroys all the player's possessions.

MUSH and MUSE only.

**Wizards and Officials only.**

## Flag Reference

In the mud, you can change the states, qualities, or abilities of other objects. Although many of these states can be mimicked at textual levels (by the creation of triggered text strings, for example), many are so far-reaching in importance that they have been removed to the level of *flags*. Flags may be thought of as switches on the object, which may be set in one direction or the other. The on position is indicated by the appearance of the flag's associated letter in the object's DBREF. The flag is off if the flag's letter doesn't appear.

All flags may be set and unset in the same way:

`@set object=<flag>`

Sets the flag.

`@set object=!<flag> to reset (unset)`

Resets, or unsets, the flag.

## Flags

## Abode

Flag: A

```
@set here=ABODE
```

Allows players to set their homes to that location. Players can also set object's homes to that location.

MUCK only.

## Bearing

Flag: b

```
@set object=BEARING
```

Allows players to @addparent or @delparent their objects to or from the specified object. If not set, the specified object can neither bear new children nor lose old ones.

MUSE only.

## Builder

Flag: B

On some muds, only players that have this flag set are allowed to build anything.

## Change Ownership Allowed

Flag: C

```
@set object=CHOWN_OK
```

Allows players to transfer ownership of the specified object to another player. To set it, you must be carrying the object. To set this flag on a room or exit, you must be in the room. Once set, the new player may gain ownership of object by using @chown.

Available in MUCK, MUSH, and MUSE.

## Connected

Flag: c

```
@set player=CONNECTED
```

Shows whether player is connected or not. This flag only applies to players.

MUSH and MUSE only.

## Dark

Flag: D

```
@set object=DARK
```

Sets the object Dark. If a room is DARK, a look sees the room's description, but does not display the objects that the room contains. If a thing is DARK, look does not list that thing in the room's contents. Players who don't want to have their location found via @whereis may set themselves DARK. Puppets cannot be DARK, but an object set puppet and DARK enters "debug" mode, and relays to its owner everything it does and hears.

Available in MUCK, MUSH, and MUSE.

## Destroyable

Flag: d

```
@set object=DESTROY_OK
```

Allows any player to destroy the specified object as long as the object is not locked against them. This is good for creating things like notes, allowing the recipient to destroy the note after reading it, instead of having to look for you to destroy it.

MUSH and MUSE only.

Enterable

Flag: e

```
@set object=ENTER_OK
```

Allows players to enter the specified object. Players must be ENTER\_OK if they want to be able to receive things from other players via the give command.

MUSH and MUSE only.

Exit

Flag: E

Available in MUCK, MUSH, and MUSE.

Floating

Flag: f

```
@set room=FLOATING
```

When set on a disconnected room, this flag suppresses the disconnected room message. Should be used only by experienced builders because floating rooms can easily be forgotten or misplaced.

MUSH and MUSE only.

Going

Flag: G

```
@set object=GOING
```

Used internally; automatically set on objects that are @destroyed. If you change your mind about @destroying an object, use @undestroy to unset it quickly!

MUSE only.

Haven

Flag: H

```
@set object=HAVEN
```

Sets the object to be a haven. A haven player ignores all messages from other players. A haven object is halted, and its register commands is not triggered. If a room is a haven, you cannot kill in that room.

Available in MUCK, MUSH, and MUSE.

Hide

Flag: h

```
@set object=HIDE
```

Used to mark players who do not want their names to show up on the WHO list. Set by @hide and reset by @unhide.

MUSE only.

Inherit

Flag: I

@set object=INHERIT

Causes the object to inherit any special powers of its owner. Otherwise, the object has no special powers.

MUSE only.

Jump Allowed

Flag: J

@set room=JUMP\_OK

Allows players to @teleport into the room.

Available in MUCK, MUSH, and MUSE.

Key

Flag: K

@set object=KEY

Sets the object to be a key. Puppets may not take the object, and the object returns to its home when its holder teleports away (except that the owner of the object can teleport with it).

Available in MUCK, MUSH, and MUSE.

Light

Flag: l

@set object=LIGHT

Allows the object to be seen in a room that is set DARK.

MUSH and MUSE only.

Link Allowed

Flag: L

@set object=LINK\_OK

Allows anyone to link exits *to* the object (but not *from* it). Anyone can also set the object as their home.

Available in MUCK, MUSH, and MUSE.

Mucker

Flag: M

@set object=MUCKER

When set on a player, it allows the player to create MUF programs in MUCK.

Available in MUCK.

## **Wizards only.**

Opaque

Flag: o

```
@set player=OPAQUE
```

Prevents other players from seeing what you are carrying in your inventory. Players can still see objects you are carrying that they own.

MUSE only.

Player

Flag: P

Available in MUCK, MUSH, and MUSE.

Puppet

Flag: p

```
@set object=puppet
```

Causes the object to grow eyes and ears; the object will relay all it sees and hears to its owner.

MUSH and MUSE only.

Quiet

Flag: q

```
@set me=QUIET
```

Blocks out all the set and triggered messages that you get when your objects are set or triggered. This is useful if you have a lot of machines running continuously. It will also keep you from hearing shake and crumble and deposit back messages when you destroy objects.

MUSE only.

Room

Flag: R

```
@set object=room
```

Identifies the object as a room; automatically set when you @dig the room. Once set, it cannot be changed.

Available in MUCK, MUSH, and MUSE.

Slave

Flag: s

```
@set player=SLAVE
```

Disables the player from doing anything except to pose; player cannot talk, page, build, get, or drop objects.

MUSE only.

## **Officials only.**

Sticky

Flag: S

@set object=STICKY

Sets an object or room to be sticky. A thing that is sticky goes home when dropped. On room that is sticky, the drop-to is delayed until the last person leaves.

Available in MUCK, MUSH, and MUSE.

Terse

Flag: t

@set object=TERSE

Sets an object or player to be terse. An object will not shake and crumble when destroyed. On a player, room descriptions won't be displayed when the player enters a new room (contents and exits are still displayed). This is good when you want to move about quickly, or don't want to read the room description. Beware: You might miss important information in some descriptions.

MUSH and MUSE only.

Transparent

Flag: T

@set exit=TRANSPARENT

Allows players to look through the exit: they will first see the exit's description, followed by the description and contents of the room on the other side.

MUSE only.

Vehicle

Flag: V

@set object=vehicle

Allows an object to contain players. Vehicles have certain restrictions placed on them, to prevent players from using vehicles to get around locks or go places they shouldn't.

Available in MUCK.

Visible

Flag: v

@set object=VISIBLE

Allows any player to examine the object and see all the object's attributes as if they owned it. They cannot make any changes to object.

MUSE only.

Visual

Flag: v

@set object=VISUAL

Allows any player to examine the object and see all the object's attributes as if they owned it. They cannot make any changes to object.

MUSH only.

Wizard

Flag: W

Denotes a player that is a wizard. Wizards cannot be killed. They can use special wizard commands, and they are not subject to the same restrictions as normal players.

Available in MUCK, MUSH, and MUSE.

Zombie

Flag: Z

```
@set object=zombie
```

Allows objects in MUCK to have the @force and the FORCE MUF primitive used on them.

Available in MUCK.

## Attribute Options Reference

Mud objects have attributes, which are registers in which you can place text or commands. The basic and at-commands generally impact directly upon one or more of the built-in attributes, such as @succ, @osucc, @asucc, and so forth. The @defattr command can be used to create entirely new attributes.

But there is still another level of mud code we can explore! A broad set of functional modifiers known as *attribute options* may be applied to @defattr commands. These options allow you to change how your objects' attributes respond to their environments in subtle and far-reaching ways.

To create or modify a user-defined attribute and/or install an option, type:

```
@defattr object/attribute=attribute-option
```

Some of these attribute options are only available on MUSEs.

### Attribute Options

Dark

A dark attribute is not visible in an examine command, and cannot be obtained by a get function.

Date

A date attribute stores date and time information, if the information is entered in xtime() format. Anyone who examines the object will see the correct date and time for their own timezone.

Function

A function attribute can create a user-defined function. You can enter any text or function into the attribute register, and it is executed whenever someone attempts to call the function (the function must first be defined on the object). For example, assume the object named testobj has already been created. You can define functions on it as follows:

```
> @defattr testobj/mult60=function inherit
```

Set.

```
> @mult60 blee=[mul(60,v(0))]
```

Set.

```
> say [mult60(3)]
```

180

```
> @defattr testobj/foo=function inherit
```

Set.

```
> @foo testobj=Everyone with [v(0)]: go to the [v(1)]!
```

Set.

```
> say [foo(coats,closet)]
```

You say "Everyone with coats: go to the closet!"

Inherit

An inherit attribute will pass its value down from parent to child. Child objects may have these values overwritten, but if these new values are erased, the original inherited value will appear again. This option is set on most of the built-in attributes.

Osee

An osee attribute can be seen by anyone who examines an object on which the attribute has a value, even if the object is not set to be visible. Attributes with a built-in osee option include last, sex, and race.

Unsaved

An unsaved attribute will not retain its contents when the mud is rebooted. The IT attribute carries the unsaved option.

Wizard

A wizard attribute can be @set only by the object's owner (or by a Wizard or Official). If you define an attribute with the wizard and inherit options, its child objects may be given to other players—even allowing other players to @chown them—and the new owners will not be able to change that attribute.

## MUF Function Reference

This section describes the functions available for use in the MUF programming language (used on MUCKs). It is based on the MUF Reference Manual.

### Notation

A special function notation is used to denote the values used on the stack, and any variables that are pushed back onto the stack. The notation

```
function(value1 value2 | result)
```

means that the given function first pops *value2* from the top of the stack and then pops *value1* from the stack. It then processes these values in some fashion and produces a result. This *result* is then pushed onto the stack. The value pushed onto the stack is denoted by the item following the vertical bar, if any. Using this syntax, the addition function would be:

```
+ (integer1 integer2 | sum)
```

A function that does not put anything back onto the stack will not have anything after the vertical bar:

```
pop (val |)
```

### Parameter Types

Throughout this section, the following terminology is used to indicate parameter types:

<b>Terminology</b>	<b>Indicates</b>
int	integer
str	string
dbref	DBREF
var	variable
boolean	either 0 or 1
value	any value
result	any value

## MUF Functions

### Arithmetic Functions

The following functions are available for arithmetic:

<b>Function</b>	<b>Description</b>
+ ( <i>int1 int2   sum</i> )	Adds int1 to int2
- ( <i>int1 int2   difference</i> )	Subtracts int2 from int1
* ( <i>int1 int2   product</i> )	Multiplies int1 by int2
/ ( <i>int1 int2   integer-result</i> )	Integer-divides int2 by int1
% ( <i>int1 int2   remainder</i> )	Modulo of int2 divided by int1

### Comparison Functions

The following functions are available for comparison. They push a boolean result of either 0 or 1 back onto the stack.

<b>Function</b>	<b>Description</b>
< ( <i>int1 int2   boolean</i> )	Less than
<= ( <i>int1 int2   boolean</i> )	Less than or equal to
> ( <i>int1 int2   boolean</i> )	Greater than
>= ( <i>int1 int2   boolean</i> )	Greater than or equal to
= ( <i>int1 int2   boolean</i> )	Equal to

@ (*var | value*)

Accesses var and places its value onto the stack.

! (*var value |* )

Places the given value into the variable var.

addprop(*item-dbref property-str property-value-str int |* )

Sets the given property on the given item. The property is set to the given property value. If this string is null, the property is set to the given integer. All four parameters to this function must be on the stack (for example, you cannot omit certain ones).

and(*value1 value2 | boolean*)

Performs Boolean AND on value1 and value2 and pushes the Boolean result.

```
atoi(str | int)
```

Converts a string into an integer. If the string is not a number, 0 is pushed onto the stack.

```
call(program-name)
```

Starts executing a different MUF program. The program must have been compiled already. The called program inherits the value of me, loc, and trigger.

```
contents(container-dbref | first-item-dbref)
```

Pushes the DBREF of the first item contained in the container onto the stack. You could then use the next function to scan through the rest of the items in the container.

```
copyobj(original-dbref | copy-dbref)
```

Creates a copy of the original thing and pushes the copy's DBREF onto the stack.

```
dbcmp(dbref1 dbref2 | boolean)
```

Compares two DBREFs and pushes the Boolean result.

```
dbref(int | dbref)
```

Changes the integer into a DBREF. dbref(54) is the same as #54.

```
desc(item-dbref | desc-str)
```

Pushes the description attribute (@desc) of the item onto the stack.

```
drop(item-dbref | drop-str)
```

Pushes the drop attribute (@drop) of the item onto the stack.

```
dup(val | val val)
```

Duplicates the top of the stack.

```
exit
```

Exits from the MUF word being executed.

```
exit?(dbref | boolean)
```

Checks if the item is an exit object, and pushes the Boolean result.

```
exits(item-dbref | exit-dbref)
```

Pushes the DBREF of the first exit object in the exit list of the given item onto the stack. You could then use the next function to scan through the rest of the exits in the exit list.

```
explode(str1 str2 | str1 ... strN int)
```

Breaks str1 into individual strings, delimited by the delimiter in str2. The individual strings are pushed in reverse order onto the stack, and then an integer indicating the number of individual strings is pushed onto the stack. For example, if str1 was "This is a test" and str2 was a space character, " ", the stack would be "test" "a" "is" "This" 4, after this function. The null string, "", cannot be used as str2.

```
fail(item-dbref | fail-str)
```

Pushes the fail attribute (@fail) of the item onto the stack.

```
flag?(item-dbref str | boolean)
```

Uses the given string parameter as the name of a flag to check if the given flag is on or off for the item. Pushes 1 if it is on and 0 if it is off.

```
getlink(item-dbref | link-dbref)
```

Pushes whatever the item is linked to onto the stack. If the item is an exit, it pushes whatever the item is linked to. If the item is a player, it pushes the DBREF of the player's home. If the item is a room, it pushes the room's drop-to, if any. If the item is not

linked to anything, #-1 is pushed.

```
getpropstr(item-dbref prop-name-str | prop-str)
```

Looks up the property named *prop-name-str* in the item, and pushes the string corresponding to that property. Pushes the null string if the property does not have a value.

```
getpropval(item-dbref prop-name-str | prop-int)
```

Looks up the property named *prop-name-str* in the item, and pushes the integer corresponding to that property. Pushes 0 if the property does not have a value.

```
if (boolean) statements [ else statements ] then
```

Tests the Boolean value. If the value is true, the statements after the if, up until the next then (or the next else, if the else is present), are executed. If the value is false, the statements between the else and next are executed. Execution then resumes after the word then. The if statement pops the Boolean value from the stack. As a helpful hint, it may be better to think of the then keyword as an endif.

```
instr(source-str pattern-str | index-int)
```

Searches for the pattern string in the source string. Pushes the index of the first occurrence of the pattern in the source, or -1 if the pattern is not found.

```
int(item | int)
```

Converts the variable item or object item into an integer.

```
intostr(int | str)
```

Converts an integer into a string. The integer could also be a DBREF.

```
location(item-dbref | location-dbref)
```

Pushes the DBREF of the location of the given item onto the stack.

```
match(str | item-dbref)
```

Checks the given string against the player's objects, the objects in the room, and the exits in the room. Pushes the first object that contains the given string onto the stack. Pushes #-1 if the string is not matched in any object.

```
moveto(item-dbref location-dbref | )
```

Moves the item to the given location.

```
name(item-dbref | name-str)
```

Pushes the name attribute (@name) of the given item onto the stack.

```
next(item-dbref | next-item-dbref)
```

Pops the given item and pushes the next item in the list of contents (if the item is an object) or list of exits (if the item is an exit) onto the stack.

```
not(value | boolean)
```

Performs the Boolean NOT on the value, and pushes 1 if the value evaluates true and 0 if the value evaluates false.

```
notify(player-dbref message-str | )
```

Sends the given string to the given player.

```
notify_except(room-dbref player-dbref message-str | )
```

Sends the given message to everyone in the given room except for the specified player. The excepted player can be #-1, in which case the message is sent to everyone in the room.

```
number?(str | boolean)
```

Pushes 1 if the string consists of a number, 0 otherwise.

```
odrop(item-dbref | odrop-str)
```

Pushes the odrop attribute (@odrop) of the item onto the stack.

```
ofail(item-dbref | ofail-str)
```

Pushes the ofail attribute (@ofail) of the item onto the stack.

```
ok?(value | boolean)
```

Pushes 1 if the value is a valid object, 0 otherwise.

```
or(value1 value2 | boolean)
```

Performs Boolean OR on value1 and value2 and pushes the Boolean result.

```
osucc(item-dbref | osucc-str)
```

Pushes the osucc attribute (@osucc) of the item onto the stack.

```
over(value other-value | value other-value value)
```

Copies the second-to-top value on the stack.

```
owner(item-dbref | owner-dbref)
```

Pushes the DBREF of the owner of the specified item onto the stack.

```
pick(value1 ... valueN ith-int | value1 ... valueN ith-value)
```

Copies the ith value from the top of the stack and pushes it onto the stack.

```
player?(item-dbref | boolean)
```

Pushes 1 if the item is a player, 0 otherwise.

```
pop(value | )
```

Pops the top of the stack.

```
program?(item-dbref | boolean)
```

Pushes 1 if the item is a program, 0 otherwise.

```
pronoun_sub(item-dbref str | new-str)
```

Uses the given item to convert pronoun substitutions (%n, %p, %o, and so forth) in the given string to the pronouns corresponding to the item. The new string is then pushed onto the stack.

```
random( | int)
```

Pushes a random integer from 0 to MAXINT onto the stack.

```
read( | str)
```

Reads a string from the user and pushes it onto the stack.

```
remove_prop(item_dbref property-str | )
```

Removes the given property from the given item.

```
rinstr(source-str pattern-str | index-int)
```

Searches for the pattern string in the source string. Pushes the index of the last occurrence of the pattern in the source, or -1 if the pattern is not found.

```
rmatch(item-dbref str | item-dbref)
```

Checks the given string against all the objects associated with the given item. Pushes the object that matches the given string onto the stack. Pushes #-1 if no object matches the string.

```
room?(item-dbref | boolean)
```

Pushes 1 if the item is a room, 0 otherwise.

```
rotate(value-i value-i+1 ... valueN ith-int | value-i+1 ... valueN value-i)
```

Moves the *ith* value to the top of the stack.

```
set(item-dbref flag-str | )
```

Turns on the given flag on the given item.

```
setdesc(item-dbref str | )
```

Sets the description attribute (@desc) of the given item to the given string.

```
setdrop(item-dbref str | )
```

Sets the drop attribute (@drop) of the given item to the given string.

```
setfail(item-dbref str | )
```

Sets the fail attribute (@fail) of the given item to the given string.

```
setname(item-dbref str | )
```

Sets the name attribute (@name) of the given item to the given string.

```
setodrop(item-dbref str | )
```

Sets the odrop attribute (@odrop) of the given item to the given string.

```
setofail(item-dbref str | )
```

Sets the ofail attribute (@ofail) of the given item to the given string.

```
setosucc(item-dbref str | )
```

Sets the osucc attribute (@osucc) of the given item to the given string.

```
setsucc(item-dbref str | )
```

Sets the succ attribute (@succ) of the given item to the given string.

```
strcat(str1 str2 | str1str2)
```

Concatenates the top two strings on the stack and pushes the result.

```
strcmp(str1 str2 | int)
```

Pushes 0 if the two top strings are equal, or a non-zero number if they are not equal. Case-sensitive.

```
strcut(source-str index-int | first-str second-str)
```

Breaks the given source string into two parts, and pushes these two parts back onto the stack. The string is broken at the character after the given index integer. Example: information 6 strcut would push inform ation back onto the stack.

```
stringcmp(str1 str2 | int)
```

Pushes 0 if the two top strings are equal, or a non-zero number if they are not equal. NOT case-sensitive.

```
strlen(str | int)
```

Pushes the length of the given string onto the stack.

```
strncmp(str1 str2 numchars-int | int)
```

Compares the first *numchars* characters of the two strings on the top of the stack. Pushes 0 if the two top strings are equal, or a non-zero number if they are not equal. Case-sensitive.

```
subst(source-str pattern-str replacement-str | new-str)
```

Takes the source string and changes all occurrences of the pattern string to the given replacement string. The resulting new string is then pushed onto the stack.

```
succ(item-dbref | succ-str)
```

Pushes the succ attribute (@succ) of the item onto the stack.

```
swap(value1 value2 | value2 value1)
```

Switches the two top values on the stack.

```
thing?(item-dbref | boolean)
```

Pushes 1 onto the stack if the item is a thing, otherwise pushes 0.

```
time( | seconds-int minutes-int hours-int)
```

Pushes the time onto the stack in the form of three integers representing the seconds, minutes, and hours at that current time.

```
variable(int | variable)
```

Converts the given integer into a variable. There are three predefined variables: me corresponds to variable number 0, loc is variable number 1, and trigger is variable number 2. Thus the code: 0 variable is equivalent to: me.

## MUSH and MUSE Function Reference

This section presents functions that are available for use in MUSH and MUSE programs.

A large number of intricate operations can be performed by mud objects, including many basic and advanced mathematical operations. You can access these operations by using *function calls*. Function calls can also be used to perform character-string and logical operations, and to generate information on the state of the database and its contents. They return their data almost instantly, allowing you to use it in your commands statements and action lists. By incorporating a few of these function calls in your objects, you can really create some impressive stuff.

### Functions

```
abs(number)
```

Returns the absolute value of the number.

```
add(a, b)
```

Returns the sum of a and b.

```
arccos(number)
```

Returns the arccosine of the number.

```
arcsin(number)
```

Returns the arcsine of the number.

```
arctan(number)
```

Returns the arctangent of the number.

```
art(word)
```

Returns the appropriate article of the following word; returns a if the initial letter of the next word is a consonant, and an if it's a vowel.

```
band(a, b)
```

Returns the bitwise AND of a and b (which are signed decimal integers).

```
base(number, oldbase, newbase)
```

Converts the number from the oldbase to the newbase. Bases must be between 2 and 36. Digits above 9 are represented by letters a = 10, through z = 35. For example, base(14,10,2) returns 111,. base(aa,16,10) returns 170.

```
bnot(number)
```

Returns the bitwise NOT of the number (which is a signed decimal integer)

```
bor(a, b)
```

Returns the bitwise OR of a and b (which are signed decimal integers).

```
bxor(a, b)
```

Returns the bitwise XOR of a and b (which are signed decimal integers).

`change(string1, string2, index)`

Changes the character at the given index in string1 to string2. Characters are numbered starting at 1.

`children(object)`

Returns a list of the object's children, in a format suitable for @foreach.

`cjust(string, width)`

Returns the string, center-justified, in a field of the given width.

`comp(n, m)`

Returns 0 if n = m, -1 if n < m, and 1 if n > m. Non-numeric strings all equal 0.

`con(object)`

Returns the first thing in the contents list of the object. Similar to the Lisp car() function.

`cos(number)`

Returns the cosine of the given number.

`cr(player)`

Returns the number of credits belonging to the player.

`delete(string, index, numchars)`

Returns the remainder of the given string after deleting numchars characters beginning with the character at the given index. Characters are numbered starting at 0.

`dist2d(x1, y1, x2, y2)`

Returns the distance between (x1, y1) and (x2, y2) as if plotted on a 2-dimensional graph.

`dist3d(x1, y1, z1, x2, y2, z2)`

Returns the distance between (x1, y1, z1) and (x2, y2, z2) as if plotted on a 3-dimensional graph.

`div(a, b)`

Returns a divided by b (this is integer division with no remainder).

`entrances(object)`

Returns a list of exits leading into the object, in a format suitable for use by @foreach.

`even(argument)`

Returns 1 if the argument is an even number, otherwise 0.

`exit(object)`

On a room, returns the first exit in the room. On a thing, returns its home.

`exp(number)`

Returns the number raised to the power of e (reverse ln).

`extract(string, wordnumber, numwords)`

Returns the substring from string starting at word number wordnumber and containing numwords words. Words are numbered starting at 1.

`fadd(a, b)`

Returns a + b, where a and b are floating point numbers.

`fcomp(a, b)`

Returns 0 if a = b, -1 if a < b, and 1 if a > b. Non-numeric strings equal 0.

`fdiv(a, b)`

Returns  $a/b$ , where  $a$  and  $b$  are floating point numbers.

`first(string)`

Returns the first word of the string.

`flags(object)`

Returns a string consisting of the current flags on the object.

`flip(string)`

Returns the string with its characters reversed.

`fmul(a, b)`

Returns  $a * b$  where  $a$  and  $b$  are floating point numbers.

`fsqrt(number)`

Returns the floating-point square root for the given number. If the number is negative, 0 is returned.

`fsub(a, b)`

Returns  $a - b$ , where  $a$  and  $b$  are floating point numbers.

`get(object, attribute)`

`get(object/attribute)`

Returns the specified attribute on the given object.

`gt(number1, number2)`

Returns 1 if  $number1$  is greater than  $number2$ , otherwise returns 0.

`hasflag(object, flag)`

Returns 1 if the object has the specified flag, otherwise 0.

`host(user)`

Returns the hostname of user. If the user is not connected, nonexistent, or hidden from you, -1 is returned.

`idle(user)`

Returns the time in seconds that the user has been idle. If the user is not connected, nonexistent, or hidden from you, -1 is returned.

`if(test, action)`

Evaluates the test, and then performs the given action if the test is true; otherwise, it does nothing. The test is false if it evaluates to 0, -1, or to the null string. Otherwise it's true.

`ifelse(test, action1, action2)`

Evaluates the test, and then performs  $action1$  if the test is true and  $action2$  if test is false. The test is false if it evaluates to 0, -1, or to the null string. Otherwise, it's true.

`isdigit(argument)`

Returns 1 if the argument is a digit, 0 if the argument is a letter or a symbol.

`land(a, b)`

Returns the logical AND of  $a$  and  $b$ .

`lattr(object)`

Returns a list of the attributes on the given objects that have a non-null value.

`lattrdef(object)`

Returns a list of the attributes defined (with @defattr) on the object.

`lcon(object)`

Returns a space-separated list of items from the object.

`lexit(object)`

Returns a space-separated list of exits on the object.

`link(object)`

On an exit, returns the exit's destination; on a player or thing, returns its home; and on a room, returns the room's drop-to room.

`linkup(object)`

Returns a list of DBREFs of objects linked to the given object.

`ljust(string, width)`

Returns the string, left-justified, in a field of the given width.

`ln(number)`

Returns the natural log of the number.

`lnot(number)`

Returns the logical NOT of the number.

`lnumber(number)`

Returns a list of consecutive numbers from 0 to (number-1). The number cannot be larger than 250.

`loc(object)`

Returns the current location (room number) of the object.

`log(number)`

**Returns the logarithm base 10 of the number.**

`lor(a, b)`

Returns the logical OR of a and b.

`lt(number1, number2)`

Returns 1 if number1 is less than number2, otherwise returns 0.

`lwho()`

Returns a list of DBREFs of connected users. It costs 50 credits for an object to use this function.

`lxor(a, b)`

Returns the logical XOR of a and b.

`match(string1, string2)`

Returns 1 if string1 matches string2; otherwise returns 0. String2 may contain wild cards.

`mid(string, index, numchars)`

Returns numchars characters from the string, starting at the specified index.

`mod(a, b)`

Returns a mod b (the remainder after integer division).

`mul(a, b)`

Returns a multiplied by b.

`name(object)`

`name (#DBREF)`

Returns the name of the object. The object itself or a DBREF can be used as the argument.

`nearby(object1, object2)`

Checks to see if object1 and object2 are nearby each other; and returns 1 if they are, 0 if they aren't, and -1 if there is a problem identifying the objects. Two objects are considered nearby if they are in the same room, or if one is inside the other one.

`next(object)`

Returns the next thing in a list.

`number(object)`

Returns the DBREF of the object.

`objectlist(object)`

Returns a list of space-separated DBREFs taken from the list of objects beginning with object. For example, here's a way to tell everyone in the room what you're carrying: @foreach objectlist(con(me))=say I've got my %0!

`objectmem(object)`

Returns number of bytes used by the object in the database.

`objectn(object)`

Returns the correct objective pronoun for the object (him, her, or it). If the object is neither male nor female, the function returns it.

`odd(argument)`

Returns 1 if the argument is an odd number, 0 if even.

`onfor(user)`

Returns the amount of time in seconds that the user has been on-line in the mud. If the user is not connected, nonexistent, or hidden from you, -1 is returned.

`owner(object)`

Returns the DBREF of the object's owner.

`parents(object)`

Returns a space-separated list of the object's parents, in a format suitable for @foreach.

`pi()`

Returns pi.

`playmem(player)`

Returns the total number of bytes used in the database by all objects owned by the player.

`plural(number, suffix)`

Returns the given suffix if the number does not equal 1; otherwise returns a null string.

`pos(string1, string2)`

Returns the position in string2 where string1 first appears.

`poss(object)`

Returns the correct possessive pronoun for the object (his, her, or its). If the object is neither male nor female, the function returns its.

`pow(a, b)`

Returns a to the b'th power, where a and b are floating point numbers.

`quota(player)`

Returns the builder's quota of the player.

`quota_left(player)`

Returns the unused builder's quota of the player.

`rand(number)`

Returns a random number between 0 and (number-1).

`remove(string, wordnumber, numwords)`

Returns the remainder of the string after removing numwords words starting at word number wordnumber. Words are numbered starting at 1.

`rest(string)`

Returns everything but the first word of string.

`rjust(string, width)`

Returns the string, right-justified, in a field of the specified width.

`room(object)`

Checks to see where the object is, and returns the room number. This works no matter how deeply nested the object is within rooms and things.

`rtime([seconds])`

Returns a time readout that can be altered by the specified amount of seconds. You can add the optional argument to `xtime()` to alter the current rtime. If the seconds argument is omitted, the current time is displayed.

`s(string)`

Performs pronoun substitution on the string.

`scomp(string1, string2)`

Compares the ASCII values of string1 and string2. Returns 0 if string1 = string2, -1 if string1 < string2, and 1 if string1 > string2.

`sgn(a)`

Returns 1 if a > 0, -1 if a < 0, and 0 if a = 0.

`simp(floating-point-number)`

Returns a rounded-off version of a floating-point number.

`sin(number)`

Returns the sine of the given number.

`spc(number)`

Returns a string consisting of the given number of spaces.

`strcat(string1, string2)`

Returns the concatenation of the two strings.

`string(character, number)`

Returns a string made up of the given character, repeated the given number of times.

`strlen(string)`

Returns number of characters in the string.

`sqrt(number)`

Returns square root of the number, rounded down to the nearest integer. If the number is negative, the function returns the square root of the number's absolute value.

`sub(a, b)`

Returns a minus b, where a and b are integers.

`subj(object)`

Returns the correct subject pronoun for the object (he, she, or it). If the object is neither male nor female, this function returns it.

`tan(number)`

Returns the tangent of the given number.

`time()`

Returns the current system time and date in alphanumeric format (although the exact value returned depends on your @tz setting). time() first checks the @tz value on the object calling the function, and if none is present it checks the @tz value of the object's owner.

`time(tz)`

Returns the current system time and date, using tz as the timezone value. The tz argument represents a number of hours before or after GMT, and can include an optional :Y or :N suffix to indicate whether or not U.S. style daylight savings time should be considered.

`truth(a)`

Returns 0 if a is a null string or 0, otherwise returns 1.

`type(object)`

Returns the object type of the object (room, exit, thing, or player).

`v(argument)`

Returns the value of the variable specified as its argument. Some of the possibilities include:

`v(0) . . . v(9)`

Returns the appropriate stack value.

`v(va) . . . v(vz)`

Returns the contents of the appropriate object register.

`v(#)`

Returns the DBREF of whoever caused the action.

`v(N)`

Return the object name of whoever caused the action, with its first letter capitalized.

`v(n)`

Returns the object name of whoever caused the action.

`v(!)`

Returns the DBREF of the object calling v().

`v(attribute)`

Returns the value of the appropriate attribute.

`wcount(string)`

Returns the number of words in the given string.

`wmatch(string, argument)`

Locates the position of the given argument in the string, where the given string is a series of words and/or numbers, separated by spaces. Returns 0 if the argument is not in the string. Not case-sensitive.

## Lock Reference

Locks are used to make the successful use of objects possible or impossible, depending upon various things (for example, who's doing the attempting, what they're carrying, what room the attempt is occurring in, and so forth). There are many types of locks, including regular locks (@lock), enter-locks (@elock), leave-locks (@llock), and use-locks (@ulock). You can apply the full variety of *lock options* to any one these lock types.

Lock options are applied by the use of punctuation in lock command statements. The following examples refer to the @lock, but the same options can be applied to all types of locks.

Any lock may be combined with any other in a Boolean expression, by using & (and), | (or), and ! (not) symbols. You can also use parentheses () to group objects into logical elements. The lock in question will be "satisfied" if the Boolean expression evaluates to true.

## Locks

```
@lock thing=object
```

Ordinary Lock. Checks against the player or object that tries to pick up the locked object (or walk through the locked exit), but also checks against all objects carried by the enactor.

```
@lock thing=+object
```

```
"Carry Lock."
```

Checks only against objects carried by the enactor, not against the enactor itself.

```
@lock thing=@object
```

```
"Indirect Locks."
```

Checks exactly as specified by the other object's lock.

```
@lock thing==object
```

```
"Is Lock."
```

Checks against the enactor only, not against objects the enactor carries.

```
@lock thing=@(object=attribute:value)
```

```
"Object Attribute Lock."
```

Checks the value of an attribute on another object (rather than on the enactor).

```
@lock thing=*player
```

```
"Player Lock."
```

Locks object for a specific player.

```
@attribute object=$pattern:/lock/actions
```

```
"Register Lock."
```

Locks a specific \$ or ! event. Different from an @ulock, which locks all \$ and ! events on the object.

## Object Design Worksheet

The following worksheet might come in handy for your complex objects.



VE \_\_\_\_\_  
VF \_\_\_\_\_  
VG \_\_\_\_\_  
VH \_\_\_\_\_  
VI \_\_\_\_\_  
VJ \_\_\_\_\_  
VK \_\_\_\_\_  
VL \_\_\_\_\_  
VM \_\_\_\_\_  
VN \_\_\_\_\_  
VO \_\_\_\_\_  
VP \_\_\_\_\_  
VQ \_\_\_\_\_  
VR \_\_\_\_\_  
VS \_\_\_\_\_  
VT \_\_\_\_\_  
VU \_\_\_\_\_  
VV \_\_\_\_\_  
VW \_\_\_\_\_  
VX \_\_\_\_\_  
VY \_\_\_\_\_  
VZ \_\_\_\_\_

FLAGS (circle):

Chown\_ok Dark destroy\_ok enter\_ok Haven Inherit Jump\_ok Link\_ok No\_walls  
opaque puppet(of:\_\_\_\_\_) visible

ATTRIBUTE OPTIONS (circle):

dark date function(\_\_\_\_\_) inherit lock osee unsaved wizard

MISCELLANEOUS: \_\_\_\_\_

## How To Program a MOO

*MOO* stands for *MUD Object-Oriented*. The original MOO was *TinyMOO*, developed by Steven White. *TinyMOO* was then expanded and developed further by Pavel Curtis, who called his version *LambdaMOO*. Almost all of the MOOs running today are *LambdaMOOs*.

MOOs are similar to *TinyMUD*-based muds, such as *MUCK*, *MUSH*, and *MUSE*, but they have enough differences to warrant their own section. Some of the differences are fairly cosmetic, such as certain commands having slightly different names and syntax. More importantly, though, MOOs differ from other *TinyMUD*-based muds in the underlying paradigm: MOOs are object-oriented.

In procedural environments, the data and the functions that operate on that data are separate. In object-oriented environments, objects contain both the data and the functions that operate on the data. This allows functionality to be grouped into more cohesive units. It also allows objects to be aware of "themselves." In MOO, an object contains attributes that store the data and verbs that define functions that operate on the data.

Objects can also inherit functionality from other objects in MOO. You can create generic objects that behave in a certain manner, and then create specific sub-types of these generic objects. You can then write functions on the specific objects, and perhaps override functions defined on the generic objects.

### Popular MOOs

Here are some popular MOOs.

<i>Name</i>	<i>Machine Name</i>	<i>Internet Address</i>	<i>Port</i>
Diversity University	erau.db.erau.edu	155.31.1.1	8888
Jay's House MOO	jhm.ccs.neu.edu	129.10.111.77	1709
LambdaMOO	lambda.parc.xerox.com	192.216.54.2	8888
MediaMOO	purple-crayon.media.mit.edu	18.85.0.48	8888
ZenMOO	cheshire.oxy.edu	134.69.1.253	7777

*LambdaMOO* is the flagship MOO. It resides at Xerox PARC. Its chief maintainer, Pavel Curtis, maintains the MOO server code.

*MediaMOO* is a MOO devoted to media research. If you work in the media field—perhaps as an English professor, a journalist, or someone researching virtual reality—you may want to sign up on this MOO.

These addresses are current as of August, 1994. Keep in mind that they may change.

### Basic Concepts

This section on MOOs assumes that you have read the previous sections on *MUCKs*, *MUSHes*, and *MUSEs*, since many of the concepts in *MUCKs*, *MUSHes*, and *MUSEs* can also be found in MOOs. Most of the commands available in *TinyMUD*-based muds have corresponding analogues in MOOs.

### Basic Values

MOOs, like other types of muds, have the basic values of numbers and strings. Numbers are always integers, and strings are enclosed in double-quotes. MOOs also have *lists*, which can be a list of any type of value. MOOs also have special values called *errors*. These errors are simply codes that can be used to check to see if some action is performed successfully. In the event of an error, the error code can be used in some manner. Lastly, MOOs have *objects*. Objects are a combination of functionality (verbs) and data (properties).

### Object Numbers

Each object in a MOO has a database reference number, just like each object in a TinyMUD-based mud, except that instead of it being called a DBREF, it is called an *object number*. Object numbers are completely synonymous with object names; in fact, you can use object numbers instead of object names. For instance, if a potted plant was object number 735 and a table was object number 675, you could type `put #735 on #675` instead of `put potted plant on table`.

## Inheritance

Objects can *inherit* values from other objects. For instance, you can create a car, and define various actions on it, such as `drive`. Then you can create a sports car that derives from the generic car, and performs special actions, such as `revving the engine`. It can also override the functionality of the object it is derived from, such as overriding the `drive` function to do something special that pertains to sports cars.

In this example, the generic car is called the *parent* object, and the specialized object that derives from it, the sports car, is called a *child* object. Child objects can override the functionality of their parent objects. If they don't, the default parent object's functionality is used. Objects can have children, and those children can have other children objects, and so forth.

## Verbs

*Verbs* are the functions that objects can have that enable them to perform operations on their attributes, calculate things, and respond to actions and events. They are similar to attributes that define action lists, but they are more powerful because you can inherit from them.

Verbs are actually little programs that are defined on an object. The generic thing object has several predefined verbs already written on it, and you can add to these predefined functions, or override them to perform specialized tasks. Verbs defined on a parent object are also defined on their children, unless they are overridden.

Verbs are denoted using a colon with their object:

```
object-name:verb-name
```

```
object-number:verb-name
```

So, if you have a chair that has a `sit` verb defined on it, you can refer to that verb as:

```
chair:sit
```

If the chair is object #788, you can also use:

```
#788:sit
```

## Properties

A *property* is a value that is stored on an object. It is similar to an attribute in TinyMUD, except that TinyMUD attributes can contain actions, whereas MOO properties are simply slots to store information about the object. Properties are typeless; that is, they can store any type of data field: numbers, strings, lists, and even other objects.

You refer to properties by using a period with their object:

```
object-name.property
```

```
object-number.property
```

You set these properties with the `@property` command:

```
@property property-name value
```

For example:

```
@property deck.numcards 52
```

## Messages

*Messages* are properties that end with the suffix `_msg`. These messages are special kinds of properties. These messages

correspond to certain kinds of verbs. For instance, a message corresponding to a fly verb might be called a fly\_msg. Most of the string attributes, such as an object's description, are actually messages that have been defined on those objects.

## Inspecting Objects

You can see an object's description in MOO in the same manner as in TinyMUD, by using the look command:

```
look object
```

If you examine the object, you can obtain the description, verbs, and properties on the object:

```
@examine object
```

Further information can be obtained using the show command. This command displays the object's name and number, its parent object's name and number, its owner's name and number, and a list of verbs and properties defined on the object:

```
@show object
```

The messages on an object can be viewed using:

```
@messages object
```

The dump command lists the programs associated with each verb defined on an object. It does not check for inherited verbs.

```
@dump object
```

To get the program listing for a specific verb, use:

```
@dump object:verb-name
```

You can also use:

```
@list object:verb-name
```

## Basic Commands

Let's digress for a moment and present commands in MOO that have equivalent commands in TinyMUD-based muds.

Many MOO commands are similar to TinyMUD-type commands. Here is a list of basic TinyMUD commands and their MOO equivalents. If you haven't read the "Overview of Common Commands" section in this chapter, and the sections on MUCKs, MUSHes, and MUSEs, you may want to go back and read those sections first.

<b>Command</b>	<b>TinyMUD command</b>	<b>MOO Equivalent</b>
acting	: (colon)	same
dropping	drop <i>object</i>	same
examining	examine <i>object</i>	@examine <i>object</i>
getting	get <i>object</i>	same
going home	home	same
inventory	inventory	same
killing	kill <i>object</i>	not present
listing players	WHO	@who
looking	look <i>object</i>	same
moving	go <i>place</i>	same
paging	page <i>player = message</i>	page <i>player "message"</i>
quitting	QUIT	@quit
talking	" (double-quote)	same
whispering	whisper <i>player = message</i>	whisper " <i>message</i> " to <i>player</i>

## At-Command Equivalencies

For a quick introduction, here is a list of TinyMUD at-commands and their MOO equivalents. For more detail, consult the "MOO Reference Manual" sections on "At-Commands" and "Messages," given later in this chapter.

TinyMUD At-Command	MOO Equivalent
@create object = cost	@create object named "name"
@desc object = description	@describe object as "description"
@dig room-name	@dig room-name
	@dig exit-name to place
@drop object = message	@arrive exit is "message"
@fail object = message	@take_failed object is "message"
	@nogo exit is "message"
@find	@audit
@link	None. Use @dig.
@name	@rename
@ofail object = message	@otake_failed object is "message"
	@onogo exit is "message"
@open	None. Use @dig.
@osucc object = message	@otake_succeeded object is
	"message"
	@oleave exit is "message"
	@oarrive exit is "message"
@succ <i>object = message</i>	@take_succeeded <i>object</i> is " <i>message</i> "
	@leave <i>exit</i> is " <i>message</i> "
	@arrive <i>exit</i> is " <i>message</i> "

## MOO Programming

Now that you have had an introduction to basic MOO concepts, and you have seen which other TinyMUD-based commands have equivalencies in MOOs, you can delve into MOO programming.

If you have programmed in the C programming language, you will find yourself quite familiar with the programming constructs in MOOs, because they are almost exactly the same.

This section assumes you have had some prior experience using the C programming language. As such, it presents the programming concepts in a quick overview fashion, rather than an extensive tutorial on computer programming.

## Expressions

Expressions evaluate to either true (non-zero) or false (zero). They are used for comparisons, conditional statements, and so forth.

The simplest expressions are literal values:

*number*

# *object-number*

"*string*"

*error-code**variable*

Next, you can create lists of expressions using:

```
{ expr, expr, ... expr }
```

These values can then be put together in more complex expressions:

<b>Expression</b>	<b>Meaning</b>
<i>expr.name</i>	Property access.
<i>expr.(name)</i>	Property access, where <i>name</i> is a string denoting the property.
<i>expr.name(arguments)</i>	Call a verb.
<i>expr.(name)(arguments)</i>	Call a verb, where <i>name</i> is a string denoting the verb.
<i>function(arguments)</i>	Call one of the built-in MOO functions.
<i>expr[expr]</i>	Index into a string or list.
<i>expr[expr .. expr]</i>	Get a slice of a string or list.
<i>name = expr</i>	Set a variable.
<i>expr.name = expr</i>	Set a property.
<i>expr + expr</i>	Add two expressions.
<i>expr - expr</i>	Subtract expressions from another.
<i>expr * expr</i>	Multiply two expressions.
<i>expr / expr</i>	Divide one expression by another.
<i>expr % expr</i>	Return the remainder of one expression divided by another.
<i>- expr</i>	Add two expressions.
<i>expr == expr</i>	Compare two expressions for equality.
<i>expr != expr</i>	Compare two expressions for inequality.
<i>expr &lt; expr</i>	See if one expression is less than another.
<i>expr &lt;= expr</i>	See if one expression is less than or equal to another.
<i>expr &gt; expr</i>	See if one expression is greater than another.
<i>expr &gt;= expr</i>	See if one expression is greater than or equal to another.
<i>expr &amp;&amp; expr</i>	Determine the logical AND of two expressions.
<i>expr    expr</i>	Determine the logical OR of two expressions.
<i>! expr</i>	Determine the logical NOT of an expression.
<i>expr IN expr</i>	Test for set inclusion.
<i>b-expr ? t-expr   f-expr</i>	Test b-expr. If true, return t-expr. If false, return f-expr.

Most of these are the same as in the C programming language. Note that the tertiary expression (given last) contains a vertical line (|) in MOO, as opposed to a colon (:) in C.

## Statements

You can use the expressions to create statements.

The simplest statement is the null statement, a semi-colon (;), which does nothing.

The conditional statement has the following syntax:

```

if (expr)

    statements

elseif (expr)

    statements

else

    statements

endif

```

The else-if and else clauses are optional. You need to use them only when you want to test further conditions or perform an action if the expression is false, respectively.

The list iteration statement has the following syntax:

```

for name in (expr)

    statements

endfor

```

and the numeric iteration statement is:

```

for name in [expr .. expr]

    statements

endfor

```

In this regard, the iteration statements are closer to the Pascal programming language than to the C programming language.

The indefinite iteration statement is:

```

while (expr)

    statements

endwhile

```

To return a value from a verb, use:

```

return expr;

```

or

```

return ;

```

Finally, to fork a task:

```

fork name (seconds-expr)

    statements

endfork

```

This last statement evaluates the expression to a number of seconds. After waiting that many seconds, a separate task will execute the given statements. Execution simultaneously continues with the statements, if any, after the endfork.

## Entering Verbs

To program a verb on an object, use the @program command:

```
@program object:verb
```

This puts you into the MOO editor. You can type in your verb program, one statement per line. The program is interpreted as a list of strings. At the end, you type a period on a line by itself

.

to tell the MOO that your verb program is complete. The MOO then error-checks it. If it is correct, the program becomes the new program for that verb on that object.

If you are entering a verb program and you want to abort the editing, type

```
@abort
```

on a line by itself. The editor will terminate, and the verb will remain unchanged.

## MOO Reference Manual

This section is a reference to MOOs. It lists at-commands, common functions, messages, and other information pertaining to MOOs.

### At-Command Reference

```
@addalias alias to object
```

```
@addalias alias to object:verb
```

Adds an alias to the name of the object or the verb on an object, respectively. For example, if you want a Corvette that you create to also be known as a car, you can use this command to add the alias "car" to the Corvette.

```
@add-entrance exit-obj-number
```

Links the given exit object to the current room as an entrance.

```
@add-exit exit-obj-number
```

Links the given exit object to the current room as an exit.

```
@add-feature feature-object
```

Adds the specified feature object to you. Features are objects that provide additional functionality to you. For example, one common feature is one where you can begin a line with a minus symbol (-), followed by a name and some text, and it will perform a directed say command at that person:

```
> -George Hey there!
```

```
You say [to George] "Hey there!"
```

You can find out the object number of this feature on your MOO by using the @features command, and thus add it to your feature list.

```
@args object:verb direct-obj
```

```
@args object:verb direct-obj preposition
```

```
@args object:verb direct-obj preposition indirect-obj
```

These three commands set the direct object, preposition, and indirect object specifiers on the given object's verb.

```
@audit
```

Prints out a list of everything that you own.

```
@check-chparent object to new-parent-obj
```

Sees if the specified parent object is already the parent of the given object.

```
@check-property object.property
```

Sees if the specified property already exists on the given object.

```
@chmod object permissions
```

```
@chmod object:verb permissions
```

```
@chmod object.property permissions
```

Changes the permissions on the object, verb, or property, respectively. The permissions are given by a string. The valid object permissions are r, w, and f. The valid verb permissions are r, w, x, and d. The valid property permissions are r, w, and c.

```
@chparent object to new-parent
```

Changes the parent of the specified object to the new parent object. The object will inherit all the verb and property definitions of the parent object.

```
@classes
```

Lists the classes available for you to use. You might want to check this out when you are first starting.

```
@clear-property object.property
```

Changes the property to use value of the parent object's property instead of itself.

```
@contents object
```

Lists the contents of the given object.

```
@copy object:verb to new-object:new-verb
```

Copies the arguments, code, and other associated information of the specified object's verb to the new object's verb.

```
@count
```

Displays the number of objects that you own. Remember that your character itself is an object that you own.

```
@create class-name named "object-names"
```

```
@create parent-obj named "object-names"
```

Creates a new object. The first form specifies one of the generic classes to use as the new object's parent. The second form uses the given parent object as the new object's parent. The object names string is a list of comma-separated aliases for the name of the object. The first item in the string will be used as the name of the new object, and the rest of the items in the string are aliases for that object.

```
> @create $thing named "Vehicle"
```

```
> @create Vehicle named "Dodge Viper,viper,car"
```

```
@describe object as description
```

Sets the description on the object.

```
@dig "place"
```

```
@dig exits-to|exits-from to "place"
```

```
@dig exits-to|exits-from to room-obj-number
```

Creates a room. The first form simply creates a room with that name, and tells you the object number of the newly created room. The second form also creates exits to and from the newly created room. The exits-to links the current room to the new room, and the exits-from links the new room back to the current room. You must own the current room in order to link to it. If you do not own the current room, you should take note of the exit object numbers and get the owner of the current room to

`@add-exit` and `@add-entrance` the exit objects to their room.

`@disinherit` *child-object* [ *from parent-object* ]

Removes a child object from the parent object. You must own the parent, but you do not have to own the child object.

`@display` *object*

Displays information about the object in a formatted manner. See the MOO help on your system for a detailed description of the options available.

`@display-option`

Allows you to customize the formatting of the `@display` command. See the MOO help on your system for a detailed description of the options available.

`@dump` *object*

This command displays the verbs and properties that have been defined on the given object.

`@edit` *object:verb*

Enables you to edit the verb on the given object. This command puts you into the MOO editor.

`@edit-option`

Enables you to customize the editor that is invoked when you enter a verb program using `@program`, or compose a mail message, and so forth. See the MOO help on your system for a detailed description of the options available.

`@eject` *object* [ *from room* ]

Removes the object from the current room, or the specified room if that is given. Objects thus removed are sent home.

`@entrances`

Lists the entrances into the current room. You must own the current room in order to see this.

`@examine` *object*

Displays a more detailed description of the given object. The description contains the object's full name, object number, aliases, owner's name, and object number, its description, its contents, and readable verbs defined on the object.

`@exits`

Lists the exits from the current room. You must own the current room in order to see this.

`@features`

Lists the features on your system that you can use. You can add these features to your player object by using the `@add` feature, and you can remove features from your player object by using the `@remove` feature.

`@forked`

Lists the forked tasks that you own.

`@gag` *object*

Prevents you from seeing messages or text from the specified object, which is usually a player. This is useful when you want to ignore someone who is being annoying or generating a lot of output.

`@gaglist`

Lists the objects you currently have gagged.

`@gender` *object*

Sets the gender of the specified object.

`@go` *place*

`@go` *room-obj-number*

Teleports you to the specified place or room number. On some MOOs, a route is computed and you walk room by room to the

destination rather than being magically teleported there.

`@hide`

Removes your character from appearing in the WHO list. However, you will still show up if someone specifically uses your name in a command such as `@whois`.

`@kids object`

Lists the children of the specified object.

`@kill task`

Enables you to kill off a task. See the MOO help on your system for a more detailed description of this command.

`@lastlog [ player ]`

Displays the last login times of the players in the database. If you specify a player name, it displays the last time that player was connected to the MOO.

`@linelength number`

Sets the length of a line on your screen. The MOO will use this length to break up text that is displayed to you. However, it is better to use a client program that provides this functionality than for you to have the MOO do it for you, since the MOO does it slower.

`@list object:verb`

Displays the code for the given object's verb.

`@more`

Sets your display to print out long messages in a page-by-page fashion. This command uses the `@pagelength` value to determine how long your screen is. If a displayed message is longer than your page length, the display is halted and you will see an indicator that there is more to follow.

`@opacity object is number`

Sets the opacity for the given object, usually a container. The opacity number is one of three values. If the number is 0, the container is transparent, and you can see into it. If the number is 1, the container is opaque, and you cannot see into it when the container is closed. If the number is 2, you cannot see into it even if it is open.

`@pagelength`

Sets the number of lines you want to have displayed to you at one time when long messages are sent to you. Also see `@more`.

`@parents object`

Displays the name and object number of all of the given object's ancestors.

`@password old-password new-password`

Changes your password from the old value to the new value.

`@program object:verb`

Allows you to enter code for the given object verb. This command puts you into the MOO editor. For details on the commands available for use in the MOO editor, consult the editor help on your MOO system.

`@property object.property [ initial-value ]`

Sets the given object's property to the specified initial value. If the initial value is omitted, the property is initially set to 0.

`@quit`

Exits you from the MOO.

`@quota`

Displays your builder's quota, which indicates how many objects you can create.

`@recreate old-object as parent-obj named "names"`

Changes the given old object to a new object with a new parent and a new set of names and aliases. The aliases are a comma-separated list of names, the first of which is taken to be the name of the new object. This command enables you to recycle older, unused objects you own into new ones.

`@recycle object`

Destroys the given object. You must own the object.

`@remove-entrance entrance`

Removes an entrance from the current room.

`@remove-exit exit`

Removes an exit from the current room.

`@remove-feature feature-object`

Removes a feature from your features list.

`@rename object to name,alias,...,alias`

Changes the name of the given object to a new name. The aliases are a comma-separated list of names, the first of which is taken to be the name of the new object.

`@rename object:verb to new-verb-name`

Renames a verb on an object to a new name.

`@rmalias alias from object`

`@rmalias alias from object:verb`

Removes an alias from an object or an object verb.

`@rmproperty object.property`

Removes a property from an object.

`@rmverb object:verb`

Removes a verb from an object.

`@sethome`

Sets your home to the current room.

`@show object`

`@show object:verb`

`@show object.property`

Displays detailed information about the given object, verb, or property.

`@suggest`

Makes a suggestion to the owner of the current room.

`@ungag object`

Removes an object, usually a player, from the list of objects you have gagged.

`@unlock object`

Removes the lock on the given object.

`@verb object:verb`

Creates a verb on the given object.

`@version`

Displays the version number of the MOO.

```
@whereis player
```

Displays the current location of the specified player.

```
@who [ player ... player ]
```

Displays the list of people that are currently using the MOO. If one or more specific players are given, information for only those players is displayed. If a specified player is not connected, the last connection time of that player is displayed.

```
@whois player
```

Displays the real-life name and e-mail address of the specified player.

```
@wrap [ on|off ]
```

Turns line wrapping on or off, as specified.

## Message Reference

MOOs have a message implementation that is similar to the attribute systems in MUCKs, MUSHes, and MUSEs (such as @drop and @odrop). These at-commands set the messages displayed when certain events happen. In MOO, these messages are a bit more consistent than in the older TinyMUD-based muds.

This section lists the messages that can be set in MOO.

To set one of these messages, use the following syntax:

```
@message-name object is "text"
```

For example:

```
> @drop vase is "You drop the vase. It shatters into a million pieces."
```

Messages that begin with @o are displayed to other players (besides the enacting player), with the name of the enacting player included in the text of the @o message.

The messages are listed alphabetically as pairs of messages—the primary message and its corresponding @o message. For each message, if the message has a default, the default message text is given for that particular message. This default message is given in parentheses after the message name.

## Messages

```
@arrive
```

Displayed to the player right after successfully using the exit.

```
@oarrive ( %N has arrived. )
```

Displayed to others in the destination room when the player successfully uses the exit.

```
@close ( You close %d. )
```

Displayed to the player who closes a container.

```
@oclose ( %N closes %d. )
```

Displayed to others in the room when the player closes a container.

```
@drop_failed ( You can't seem to drop %t here. )
```

Displayed to the player who fails to drop something.

```
@odrop_failed ( %N tries to drop %t but fails! )
```

Displayed to others in the room when the player fails to drop something.

@drop\_succeeded ( You drop %t. )

Displayed to the player who drops something.

@odrop\_succeeded ( %N drops %t. )

Displayed to others in the room when the player drops something.

@ejection ( You expel %d from %i. )

Displayed to the player who ejects something from the room.

@oejection ( %D is unceremoniously expelled from %I by %n. )

Displayed to others in the room when the player ejects something from the room.

@empty ( It is empty. )

Displayed as the container's contents when the container is empty.

@leave

Displayed to the player right before successfully using the exit.

@oleave ( %N has left. )

Displayed to others in the source room when the player successfully uses the exit.

@nogo ( You can't go that way. )

Displayed to the player who fails to use an exit.

@onogo (no default)

Displayed to others in the room when the player fails to use an exit.

@open ( You open %d. )

Displayed to the player who opens a container successfully.

@oopen ( opens %d. )

Displayed to others in the room when the player opens a container successfully.

@open\_fail ( You can't open that. )

Displayed to the player who opens a container unsuccessfully.

@oopen\_fail (no default)

Displayed to others in the room when the player opens a container unsuccessfully.

@put ( You put %d in %i. )

Displayed to the player who puts something into a container.

@oput ( puts %d in %i. )

Displayed to others in the room when the player puts something into a container.

@put\_fail ( You can't put %d in that. )

Displayed to the player who fails to put something into a container.

@oput\_fail

Displayed to others in the room when the player fails to put something into a container.

@remove ( You remove %d from %i. )

Displayed to the player who removes something from a container.

@oremove ( removes %d from %i. )

Displayed to others in the room when the player removes something from a container.

@remove\_fail ( You can't remove that. )

Displayed to the player who fails to remove something from a container.

@oremove\_fail

Displayed to others in the room when the player fails to remove something from a container.

@take\_failed ( You can't pick that up. )

Displayed to the player who fails to take something.

@otake\_failed

Displayed to others in the room when the player fails to take something.

@take\_succeeded ( You take %t. )

Displayed to the player who succeeds in taking something.

@otake\_succeeded ( picks up %t. )

Displayed to others in the room when the player succeeds in taking something.

@victim\_ejection ( You have been expelled from %i by %n. )

Displayed to the player who is ejected from a room.

## Error Reference

<b>Error</b>	<b>Meaning</b>
E_NONE	No error
E_TYPE	Type mismatch
E_DIV	Division by zero
E_PERM	Permission denied
E_PROPNF	Property not found
E_VERBNF	Verb not found
E_VARNF	Variable not found
E_INVIND	Invalid indirection
E_RECMOVE	Recursive move
E_MAXREC	Too many verb calls
E_RANGE	Range error
E_ARGS	Incorrect number of arguments
E_NACC	Move refused by destination
E_INVARG	Invalid argument
E_QUOTA	Object ownership quota exceeded

## Built-In Function Reference

MOOs come with a set of built-in functions you can access in your MOO programs. These built-in functions are described in this section. For more detailed information, consult the specific topic help documentation on your MOO.

In addition to these basic built-in functions, many MOOs have packages of utility functions that are also very helpful, such as math utilities, string manipulation, list manipulation, e-mail systems, news systems, democracy and voting systems, and so forth. You may want to look at your MOO help for such utilities by typing `help utilities` or `help core-index`.

Throughout this section, the following terminology is used to indicate parameter types and return values.

<i>Terminology</i>	<i>Indicates</i>
<i>num</i>	integer
<i>str</i>	string
<i>obj</i>	an object
<i>list</i>	a list of values
<i>void</i>	no return value
<i>value</i>	any value
<i>result</i>	any value

*num* `abs(num)`

Returns the absolute value of the given number.

*void* `add_property(obj, property-name-str, initial-value, info-list)`

Adds the specified property to the given object and all of its descendants. The property will have the given initial value, and its property information will be set to the given property information list. If any of the arguments are invalid, the error E\_INVARG is returned. If the programmer does not have write permission on the object, E\_PERM is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

*void* `add_verb(obj, info-list, args-list)`

Creates a new verb on the given object. The new verb's owner, permissions, and name (or names) is specified by the information list. The new verb's direct object, preposition, and indirect object are specified in the argument list. If any of the arguments is invalid, the error E\_INVARG is returned. If the programmer does not have write permission on the object, E\_PERM is returned.

*list* `callers()`

Returns a list of the verbs that have called the current task. Similar to a trace of the verbs that have successively called other verbs, resulting in the current task being executed.

*obj* `caller_perms()`

Returns the currently-executing verb's caller's permissions.

*void* `clear_property(obj, property-name-str)`

Clears the specified property on the given object. Clear properties return the value of the parent's property when accessed, as opposed to their own value. If the programmer does not have write permission on the object, the error E\_PERM is returned. If the object is not valid, E\_INVARG is returned. If the given object defines the specified property, as opposed to inheriting it, E\_INVARG is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

*list* `children(obj)`

Returns a list of the children of the given object.

*void* `chparent(obj, new-parent-obj)`

Changes the parent of the given object to the given new parent object. If the given object is not valid, E\_INVARG is returned. If the new parent object is not fertile (has its "f" bit set) or is not owned by the owner of the given object, E\_PERM is returned.

```
str connection_name(obj)
```

Returns the name of the connection being used by the player.

```
list connected_players()
```

Returns a list of the object numbers of the players currently connected to the MOO.

```
num connected_seconds(player-obj)
```

Returns the number of seconds the given player object has been connected to the MOO.

```
obj create(parent-obj [, owner-obj])
```

Creates a new object that has the given parent object as its parent. If the owner object is not specified, the owner is the programmer. If the owner object is #-1, the object is its own owner. If the owner object is a valid object, it becomes the owner of the newly created object. The new object's name is the empty string, its location is initialized to nothing (-1), and it has no contents. The various property flags are set to zero. In addition, the new object inherits other properties from its parent object. The parent must be fertile (have its "f" bit set) or must be owned by the programmer. Otherwise, the error E\_PERM is returned. If the player creating the object cannot do so because of a quota limitation, the error E\_QUOTA is returned.

```
str crypt(str)
```

Encrypts the given string and returns the encrypted string.

```
str ctime([time-num])
```

Uses the given time number to create a string that specifies the time. If the time number is not given, the current time is used. The format of the string is *DDD MMM XX HH:MM:SS YYYY TZN*, as in Tue Jul 19 12:00:00 1994 EST.

```
void delete_property(obj, property-name-str)
```

Deletes the specified property from the given object and all of its descendants. If the programmer does not have write permission on the object, the error E\_PERM is returned. If any of the arguments are not valid, E\_INVARG is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

```
void delete_verb(obj, verb-name-str)
```

Deletes the specified verb from the object and all of its descendants. If any of the arguments are invalid, the error E\_INVARG is returned. If the programmer does not have write permission on the object, E\_PERM is returned.

```
list eval(str)
```

Evaluates the given string as though it were a piece of MOO code. The returned value is a list. The first item in the list is either 1, indicating that the code did not have compilation errors, or 0, indicating that it did. The second item in the list is either the result of the evaluation if there were no compilation errors, or a string denoting the compiler error messages if there were such errors.

```
num idle_seconds(player-obj)
```

Returns the number of seconds the given player object has been idle on the MOO.

```
num index(subj-str, pattern-str [, CASE-MATTERS])
```

Finds the first occurrence of the pattern string in the given subject string, and returns the index of this first occurrence of such pattern in the subject. Case is significant only if the optional CASE-MATTERS specifier is given.

```
num is_clear_property(obj, property-name-str)
```

Returns 1 if the property on the object is clear, 0 otherwise. Clear properties return the value of the parent's property when accessed, as opposed to their own value. If the programmer does not have read permission on the object, the error E\_PERM is returned. If the object is not valid, E\_INVARG is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

```
num is_player(obj)
```

Returns a non-zero value if the given object is a player, otherwise returns 0. If the given object is not valid, E\_INVARG is

returned.

```
void kill_tasks(task-id-num)
```

Stops the given task by removing it from the list of queued tasks.

```
num length(list)
```

Returns the number of items in the given list.

```
num length(str)
```

Returns the length of the given string.

```
list listappend(list, value, [, index-num])
```

Inserts the given value into the given list at the end of the list. If the optional index number is specified, the value is inserted after the value in the list with that index. Lists are always indexed beginning at 1.

```
list listdelete(list, index-num)
```

Deletes the index-th item from the list, and returns the new list. If the given index is out of range of the list, the error E\_RANGE is returned.

```
list listinsert(list, value, [, index-num])
```

Inserts the given value into the given list at the beginning of the list. If the optional index number is specified, the value is inserted before the value in the list with that index. Lists are always indexed beginning at 1.

```
list listset(list, value, index-num)
```

Sets the index-th item in the given list to the given value.

```
list match(subj-str, pattern-str [, CASE-MATTERS])
```

Searches for the first occurrence of the pattern in the subject. If found, a list of information pertaining to the match is returned. The pattern can contain special MOO regular expressions. For more details, consult the MOO help.

```
num max(num1, num2, ..., numN)
```

Returns the maximum of the given set of numbers. If any argument is not a number, E\_TYPE is returned.

```
obj max_object()
```

Returns the highest object number of the currently existing objects.

```
num min(num1, num2, ..., numN)
```

Returns the minimum of the given set of numbers. If any argument is not a number, E\_TYPE is returned.

```
void move(obj, location-obj)
```

Moves the given object to the given location. Various permissions are checked to make sure that the object has permission to move to the specified location.

```
void notify(player-obj, str)
```

Displays the given string to the given player.

```
obj parent(obj)
```

Returns the parent object of the given object.

```
result pass(args, ....)
```

Calls the verb defined on the parent object with the same name as the current verb. This enables programmers to define verbs that add onto the actions of parent object verbs.

```
list players()
```

Returns a list of the object numbers of all the players that are in the database.

```
list properties(obj)
```

Returns a list of the names of the properties defined on the given object. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have permission to read the properties, E\_PERM is returned.

```
list property_info(obj, property-name-str)
```

Returns a list containing information about the specified property on the given object. The list that is returned has the form { owner, permissions }. The owner is the owner of the property, and permissions is a string denoting if the property is readable, writable, or child-owner-inheritable (r, w, and c, respectively). If the programmer does not have read permission on the object, the error E\_PERM is returned. If the object is not valid, E\_INVARG is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

```
list queued_tasks()
```

Returns a list denoting the tasks owned by the programmer, whether they are forked, suspended, or reading user input.

```
num random(num)
```

Returns a random number from 1 to the given number. If the argument is a negative number, the error E\_INVARG is returned.

```
str read([player-obj])
```

Suspends the current task, waits for the user to enter a line of text, then resumes execution by returning the text line as a string. Fails if the task has already been suspended.

```
void recycle(obj)
```

Recycles the given object. The object is permanently destroyed. If the programmer does not own the object, the error E\_PERM is returned. If the object is not valid, E\_INVARG is returned. The given object's parent becomes the parent of the given object's children.

```
num rindex(subj-str, pattern-str [, CASE-MATTERS])
```

Finds the last occurrence of the pattern string in the given subject string, and returns the index in the subject that matches the first character of the pattern. Case is significant only if the optional CASE-MATTERS specifier is given.

```
list rmatch(subj-str, pattern-str [, CASE-MATTERS])
```

Searches for the last occurrence of the pattern in the subject. If found, a list of information pertaining to the match is returned. The pattern can contain special MOO regular expressions. For more detail, consult the MOO help.

```
num seconds_left()
```

Returns the number of seconds left before the currently-executing task will be terminated due to time constraints.

```
str server_version()
```

Returns a string denoting the version of the MOO server.

```
list setadd(list, value)
```

Adds the given value to the given list. The list is considered to be a set of values—thus, the value is not added to the list if it already exists in the list.

```
list setremove(list, value)
```

Removes the given value to the given list. If the value occurs more than once in the list, only the first occurrence of the value is removed from the list.

```
void set_player_flag(obj, value)
```

Sets the player status of the given object to the specified value. If the value is non-zero, the object is a player. If the value is 0, the object ceases to be a player. If the object is not a valid object, E\_INVARG is returned.

```
void set_property_info(obj, property-name-str, info-list)
```

Sets the specified property on the given object to the specified property information list. This list has the format { owner, permissions }, where the owner is the owner of the property and permissions is a string denoting if the property is readable, writable and child-owner-inheritable (r, w, and c, respectively). If the programmer does not have write permission on the object, the error E\_PERM is returned. If the object is not valid, E\_INVARG is returned. If the object does not have a property with the specified name, E\_PROPNF is returned.

```
void set_task_perms(obj)
```

Changes the currently-executing verb's permissions to be the same as the given object's.

```
void set_verb_args(obj, verb-name-str, info-list)
```

Sets the argument list of the specified verb on the given object. This list has the form {*direct-obj*, *preposition*, *indirect-obj*}. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have write permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

```
void set_verb_code(obj, verb-name-str, code-list)
```

Sets the code listing associated with the specified verb on the given object. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have read permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

```
void set_verb_info(obj, verb-name-str, info-list)
```

Sets the information list associated with the specified verb on the given object. The list has the form { *owner*, *permissions*, *names* }, where *owner* is the owner of the object, *permissions* denote whether the verb is readable, writable, executable, or in debug mode (r, w, x, and d, respectively), and *names* denote synonymous names of the verb. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have write permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

```
num sqrt(num)
```

Returns the square root of the given number. If the given number is negative, the error E\_INVARG is returned.

```
num strcmp(str1, str2)
```

Compares the two strings. If *str1* is lexicographically less than *str2*, a negative number is returned. If *str1* is lexicographically greater than *str2*, a positive number is returned. If the two strings are equal, 0 is returned.

```
str strstr(subj-str, pattern-str, replacement-str [, CASE-MATTERS])
```

Replaces the given pattern string in the given subject string with the given replacement string. If the optional CASE-MATTERS specifier is given, case is significant when the pattern matches substrings in the subject.

```
void suspend(seconds-num)
```

Suspends the currently-executing task for at least the specified number of seconds. Afterwards, the task resumes execution.

```
num task_id()
```

Returns the task identifier for the currently executing task.

```
num ticks_left()
```

Returns the number of seconds left before the currently-executing task will be terminated due to time constraints.

```
num time()
```

Returns the number of seconds elapsed since January 1, 1970.

```
obj toobj(value)
```

Converts the given value to an object number. Numbers are converted to object numbers. Strings that represent numbers are converted to object numbers, also. Such strings can begin with a # symbol. Lists return the E\_TYPE error.

```
num tonum(value)
```

Converts the given value to a number. Objects are converted to their object numbers. Strings are converted to their numeric values. If the value is a string, but does not represent a number, 0 is returned. If the value is a list, the error E\_TYPE is returned.

```
str tostr(value)
```

Converts the given value to a string. If the given value is a list, the string {list} is returned.

```
num typeof(value)
```

Returns the type of the given value. This number will be one of the following constants: NUM, STR, LIST, OBJ, or ERR.

```
num valid(obj)
```

Returns a non-zero number if the given object is a valid object and 0 if the given object is not a valid object. A valid object is one that has been created and not recycled.

```
list verbs(obj)
```

Returns a list of the names of the verbs defined on the given object. If the object is not a valid object, E\_INVARG is returned. If the programmer does not have permission to read the object, E\_PERM is returned.

```
list verb_args(obj, verb-name-str)
```

Returns the argument list of the specified verb on the given object. This list has the form {*direct-obj*, *preposition*, *indirect-obj*}. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have read permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

```
list verb_code(obj, verb-name-str)
```

Returns a listing of the code associated with the specified verb on the given object. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have read permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

```
list verb_info(obj, verb-name-str)
```

Returns a list of the information associated with the specified verb on the given object. The list has the form { *owner*, *permissions*, *names* }, where *owner* is the owner of the object, *permissions* denotes whether the verb is readable, writable, executable, or in debug mode (r, w, x, and d, respectively), and *names* denote synonymous names of the verb. If the object is not valid, the error E\_INVARG is returned. If the programmer does not have read permission on the object, E\_PERM is returned. If the specified verb name does not exist on the given object, E\_VERBNF is returned.

## Summary

You are now completely prepared to create environments of astounding complexity and lifelike responsiveness. All the tools in the world, however, won't help you if you wander around from area to area without completing anything coherent. The creation of coherent, consistent, and entertaining interactive environments is a nascent artform, one that requires not only programming capability but artistic sensibility as well.

What is a mud? In a strict sense, it is a highly structured imaginative exercise—that is, a game—which involves the simulation of a fictional reality on a literary and dramatic level. The goal of interactive narrative is to allow the participants to project their various alter-egos into your world-specific simulations, discovering problems and resolutions with a high degree of internal consistency.

A fledgling devotion somewhere between art and science, interactive narrative design relies upon a tricky sort of relativistic thinking—the attainment of an object-and-pattern-oriented skill set, which in itself is a new and exciting development. This mental skill set draws its inspirations from the fields of game theory and computer programming, as well as the more traditional narrative arts. But unlike the traditional novelist or playwright, whose stories progress in a linear fashion from event to event, a mud builder cannot simply decide which interesting turn the plotline "should" take next. Here in the living electronic world, players are free to move through your story in any variety of ways, and you should be ready to give that freedom to them. Consider all your possible "branch-points" and combinations of story elements, and draw a practical but flexible conceptual framework to encompass their interplay.

With a little brainstorming and a few handy commands, it's possible to flesh your design out in short time; creating intricate mazes of interacting elements. Try to alternately tighten and release your control over the plotline as the players proceed; rely instead on your descriptive and programming skills; you don't want players to feel restricted within the conceptual confines of your narrative. By concentrating on the quality and detail of your interactive elements, you will shift the players' attention—and more importantly, your own—from the *structural mechanics* of interactive stories to the stories themselves.

After spending a bit of time as a mud programmer, you should really be getting the hang of it. You should be able to settle into a groove that feels natural and economic for you, using these pages as your handbook. Soon after that, you'll be interacting *with* the text, *on* it, and *in* it!





- [12](#)
  - [Listservs and Mailing Lists](#)
    - [--by Philip Baczewski](#)
      - [Finding and Configuring Mailing List Software](#)
        - [An Overview of the Mercury NLM](#)
        - [List Management Features of the Mercury NLM](#)
        - [Setting up a Mercury Mailing List](#)
        - [More on Mercury](#)
      - [Introducing MajorDomo](#)
        - [Majordomo Commands](#)
        - [Defining a Majordomo List](#)
        - [Summing up Majordomo](#)
      - [Listproc](#)
        - [Listproc Commands](#)
        - [Installing and Configuring Listproc](#)
        - [One More Word on Listproc](#)
      - [Paco Xander Nathan](#)
        - [--by Tod Foley](#)
      - [Listserv](#)
        - [User Commands](#)
        - [Mailing List Owner Commands](#)
        - [Listserv Installation and Management](#)
        - [A Closing Word on Listserv](#)
      - [Some Aspects of Setting up Your Own List](#)
        - [Choosing a Forum](#)
      - [Mailing Lists and Usenet](#)
        - [Usenet Digests](#)
        - [Usenet Echoes](#)
      - [For More Information\[el\]](#)

# 12

## Listservs and Mailing Lists

[--by Philip Baczewski](#)

So, you want to run your own mailing list. You too can join a mere 5000-8000 other mailing list owners who provide a unique service to the Internet. Sarcasm aside, it is getting easier to set up and run a mailing list. Not only is the supporting software more sophisticated and readily available, but the choice of host computing platforms has also become more varied. A potential world-wide pool of subscribers provided courtesy of the Internet now makes it possible to sustain mailing list on quite esoteric topics. More and more mailing lists are being created to support specific professional organizations or clubs. It is now likely, then, that if you are involved with using or supporting use of Internet facilities, you may be called upon to start and maintain an on-line mailing list (just like when the person who owned the computer used to end up doing the club newsletter).

At its simplest, a mailing list can just be a list of e-mail addresses which allows a single message to be sent to a group. To provide automated interaction between multiple subscribed individuals, however, usually requires some piece of software to accept subscription and sign-off requests, redistribute messages, maintain message archives, and possibly maintain a file distribution service. For many years, the Revised Listserv software written by Eric Thomas, was the only program which performed all of the above tasks and more. Listserv was primarily associated with the Bitnet educational and research network, and ran on IBM mainframe systems. The program utilized the IBM RSCS protocol supported on IBM operating systems and emulated by various software offerings on other non-IBM Bitnet nodes. More recently, a version of Listserv has been produced for a diverse set of computer platforms, and several additional programs are available to manage the tasks associated with running a mailing list.

The first step to starting a mailing list is finding a host to support it. This may be an individual computer workstation that you manage, or it could be an Internet-connected multiuser host system. It may even be a remote system where other mailing lists are maintained. If mailing list software is not installed, then you will either need to install one of the programs that are available, or request that it be installed by your system manager. In either case, you will need to carefully evaluate your requirements for mailing list management and select and acquire the software which best supports those requirements. The following discussion may serve as a head start on the process by offering some information on the most popular mailing list packages currently available.

### Finding and Configuring Mailing List Software

There are several factors to keep in mind when "shopping" for mailing list software. You must know what kind of computer will be available to run the software. You should have some idea of the size and purpose of your intended mailing list or lists. You need to evaluate what other supporting features are required from your mailing list software (archives, file distribution, and so on). Often, question number one overrides any others. If your only available host is a Novell file server, then that may make the decision for you. However, all of the programs discussed below provide, at least, basic automated list management features and all of them are currently in use at a number of Internet sites.

### An Overview of the Mercury NLM

David Harris, the programmer from New Zealand who provides many Novell file server installations with an e-mail program named Pegasus Mail (P-Mail), has also written an SMTP mail gateway to go along with that

program. His name for that mail delivery gateway is Mercury (surprise) and it runs as a Netware Loadable Module (NLM). Novell types will remember that an NLM is a dynamically loadable extension to the Netware operating system which first appeared with Netware 3.0.

The Mercury NLM provides an interconnection between a Netware queue and a UNIX mail delivery agent (for example, sendmail), and allows P-Mail users to send Internet mail, if that UNIX system is on the Internet. An additional feature of the Mercury NLM is that it has automated mailing list capabilities built in. This is quite a handy feature for maintaining local discussion lists, but it will also just as easily support Internet mailing lists. Mercury is distributed along with Pegasus mail and is primarily available from risc.ua.edu in the /pub/network/pegasus directory.

## List Management Features of the Mercury NLM

The Mercury NLM supports the basic commands needed to maintain an automated Internet e-mail discussion list. The part of the software that performs these tasks is referred to as Maiser. The supported commands operate in a similar fashion to some familiar and more established mailing list software packages. Subscribe *listname* and Unsubscribe *listname* commands are used to add or delete your name to a list. Mail to be distributed to the list is sent to *listname@internet.node*. Commands such as Subscribe and Unsubscribe are usually sent to *maiser@internet.node* (*maiser* is short for *mail server*). A mail alias can also be defined in place of the name Maiser, if, for example, you wished people to send subscription requests to *listmng@your.internet.node*.)



### Is it Listserv or isn't it?

In the good old days, there was just one major mailing list software package; it was named Listserv, and it ran on Bitnet. Now a number of different software packages are available and they may be addressed as listserv but they are not always the "real" Listserv. The Revised Listserv, written by Eric Thomas, was, as its name implies, a revision and extension of a mailing list server program written by EDUCOM staff and installed at BITNIC, the Bitnet network information center, under the User ID LISTSERV. Apparently, no one holds a legal trademark on the name "listserv," but the program written by Thomas has clearly utilized that name for the longest period of time.

So, you say, "Listserv is to mailing lists what the Hershey bar is to chocolate, and Kleenex is to tissue, and Scotch tape is to cellophane tape, and the Internet is to networks. Does it really make a difference?" The answer is a resounding "probably." Because Listserv is the most established of the mailing list software—and in some ways, one of the most documented—people expect certain commands to work when they address a message to an entity named listserv.

In order to 1) be honest with people, and 2) give them the most efficient use of mailing list facilities, it is probably best to avoid using the name listserv as the alias for one of the non-Listserv packages. This is not a value judgement in relation to any of the software, but more of a practical comment. If people know they are not dealing with a familiar Listserv installation, they are more likely to request a help file (or consult some other reference) before trying to utilize that facility. This will save the headache (not to mention, the possible expense) of having to receive and deal with rejection messages for unsupported commands. While most packages support your basic subscribe command, in some cases the similarity ends there.

Even Brent Chapman (of Majordomo fame) says the following in the readme file distributed with his program: "I used to suggest using listserv, but I don't any more; Majordomo is not compatible with Bitnet Listserv, and calling it listserv confuses people." The bottom line, from my point of view, is if it's not Listserv, don't call it listserv.

Maiser supports a number of different list configuration options. Lists can be moderated (only the moderator can post messages), restricted (only members of the list may post), or fully public. Open subscription as well as moderator-controlled subscription options are also available. One installation can support multiple lists, moderated by different people. MAISER supports a number of list user and list management commands, as shown in the following list. You will notice that some commands have one or more variations.

<b><i>User Command</i></b>	<b><i>Enables You to</i></b>
SUBSCRIBE <i>listname</i> SUB <i>listname</i>	Subscribe to the indicated mailing list
UNSUBSCRIBE <i>listname</i> UNSUB <i>listname</i> SIGNOFF <i>listname</i>	Sign off from the mailing list
ENUMERATE <i>listname</i> REVIEW <i>listname</i>	Receive a listing of mailing list members
LIST	Receive an inventory the mailing lists available at this host
HELP	Receive a copy of the help file defined in the MAISER section of the MERCURY.INI file
LOOKUP <i>string</i>	Search the associated host's NetWare bindery for usernames matching <i>string</i> —the string can contain the wild cards * and ? (the file server manager can disable lookup by excluding the LOOKUPFILE entry in the MAISER section of the MERCURY.INI file)
INDEX	Receive a copy of an index to any files associated with this installation (this command sends the file INDEX.TXT from the "files to send" directory of the Mercury installation)
SEND <i>filename</i>	Receive a copy of a file listed in the index file (the file server manager can disable this feature by excluding the SEND_DIR entry in the MAISER section of the MERCURY.INI file)
<b><i>Moderator Command</i></b>	<b><i>Enables You to</i></b>
ADD <i>listname address</i>	Add a new user to the indicated list (for moderators only)
REMOVE <i>listname address</i>	Remove a user from the indicated list (for moderators only)
BOUNCE	Return a message to the sender, with the headers intact
VERIFY <i>address</i>	Returns a message indicating
whether the address specified is valid on the local host.	

## Setting up a Mercury Mailing List

Once the Mercury NLM is installed on the file server, creating a list is an easy task. This will usually need to be done by the NetWare file server manager or someone with an equivalent level of access. List service configuration statements are added to the mercury.ini file (Mercury's configuration file) and a "list of lists" file will need to be created and reference in the mercury.ini file. The list file contains the definitions for the mailing lists supported by

that Mercury installation.

The following examples show how a mailing list could be defined using the Mercury NLM. This mercury.ini example appears courtesy of Eriq Neale (neale@acs.unt.edu), Academic Computing Services file server manager at the University of North Texas. The Maiser section of the file is seen at the end. Statements preceding the Maiser section define the environment for performing the SMTP gateway function.

Anything after a # to the end of the line is a comment, and # is stripped out before parsing. Trailing and leading white space is also stripped before parsing.

#

[General]

```

myname:      acs.unt.edu      # Canonical name for this server
timezone:    GMT+6           # Time Zone to add to date fields
mailqueue:   MAILQ           # Where mail should be put for delivery
smtpqueue:   MAILQ           # Where the SMTP client should look for
                                     # mail
                                     # note: smtpqueue and mailqueue can be the
                                     # same

```

[Mercury]

```

failfile:    SYS:SYSTEM/MERCURY/FAILURE.MER # Delivery failure notification
                                                    # template
confirmfile: SYS:SYSTEM/MERCURY/CONFIRM.MER # Delivery confirmation template
aliasfile:   SYS:SYSTEM/MERCURY/ALIAS.MER   # System-wide alias file
synfile:     SYS:SYSTEM/MERCURY/SYNONYM.MER # User synonym database
listfile:    SYS:SYSTEM/MERCURY/LISTS.LST   # List of lists
logfile:     SYS:SYSTEM/MERCURY/MERCURY.LOG # Traffic logging file
bitnethost:  ricevm1.rice.edu      # Relay host for ".bitnet" rewrites
poll:        10                    # Seconds between queue polling cycles
gullible:    0
scratch:     SYS:SYSTEM/MERCURY      # Where we can write temp files

```

```
switch:      1          # number of ms to yield per op on heavy I/O
returnlines: 15        # How many lines of failed messages to
                        # return
postmaster:  NEALE     # NetWare UIC of postmaster
swapids:     1
[MercuryC]
host:        129.120.1.1 # mail mail host which relays for us
scratch:     SYS:SYSTEM/MERCURY # Where we can write temp files
poll:        30         # Seconds between queue polling cycles
switch:      1          # number of ms to yield per op on heavy I/O
returnlines: 15        # How many lines of failed messages to
return
failfile:    SYS:SYSTEM/MERCURY/FAILURE.MER # Delivery failure template
timeout:     300
[MercuryS]
switch:      1
debug:       1          # Whether or not to show session progress
discard:     20000
[MercuryP]
scratch      :          SYS:SYSTEM/MERCURY
switch       :          2
[Groups]
# Alias for group          Actual NetWare group name
[Domains]
# NetWare Server          Domain name
```

```

ACS          :      acs.unt.edu

ACS          :      ACS

[Maiser]

Maiser       :      Maiser

Helpfile     :      SYS:SYSTEM/MERCURY/MAISER.HLP

Lookupfile   :      SYS:SYSTEM/MERCURY/MAISER.LKP

Send_dir     :      SYS:SYSTEM/MERCURY/SENDABLE

Logfile      :      SYS:SYSTEM/MERCURY/MAISER.LOG

Notify       :      SYS:SYSTEM/MERCURY/TMP

Local_only   :      Y

```

The following is an example mailing list definition (gcba-l). "Green Chili Burp and the Aftertaste" are a band playing in a style the author has coined "facetious rock." Eriq Neale happens to be the band leader (the Burp, as it were).

```
; Example entry from a "list of lists"
```

```
;
```

```
gcba-l
```

```

file:          sys:system/mercury/gcbal.lst

title:         Green Chili Burp and the Aftertaste List

welcome:       sys:system/mercury/gcbalw.txt

moderated:     N

Public:        Y

Reply_To_List: Y

enumerate:     Y

```

## More on Mercury

Mercury and Maiser work only on Novell file servers, but there are a lot of those around. Mercury provides a robust mailing list management solution within the bounds of a Netware environment. Now when you see "Maiser" referenced in those mailing list announcements, you will know from what kind of system that list originates.

## Introducing MajorDomo

Majordomo, written by Brent Chapman, is a relative newcomer to the set of mailing list software offerings. Because it is written primarily in Perl, it is usable on UNIX systems only. The primary source for distribution is the site <ftp.greatcircle.com>, in the directory `pub/majordomo`. Majordomo is simple to install (on most systems) and is flexible enough to support several mailing list configuration variations.

Majordomo will need to be installed on a UNIX system by someone with root (su) privilege. The Majordomo manager (who creates the lists and so on) need not be the system manager, however. Each list created will have at least one owner. Majordomo supports both closed and open lists. Being added to a closed list requires the approval of the list owner.

Majordomo supports a number of different list types, some of which can be utilized in combination. An open list is one to which anyone can subscribe with no restrictions. Addition to a closed list requires approval of the list owner. A private list's members can be listed only by other list members. Messages to a moderated list can be posted only by the list owner. An auto list allows subscription and other commands to be automatically processed without needing any approval by the owner. You can also restrict lists to a specific Internet domain. These various configurations make it possible to define the type of list needed for the particular purpose.

### Majordomo Commands

The commands supported by Majordomo are similar in concept to those of other mailing list software. They can be sent via an e-mail message to the address `majordomo@listsite.internet.site` (or to whatever alias has been defined for Majordomo). At the heart of any such software are the subscribe and unsubscribe commands, and Majordomo is no different. It does provide a slight variation on these commands, however.

The following shows a summary of Majordomo user and list owner commands. The parameters in braces ({} ) are optional. A vertical bar (|) stands for OR.

To add your name to a Majordomo list you can use the format:

```
subscribe listname {address}
```

To remove your name from a list, use the following command:

```
unsubscribe listname {address}
```

Here, *listname* is the name of the mailing list and *address* is an optionally specified address different from the one from which the mail is sent.

Several other commands can provide you with information about a list or about the lists maintained on a Majordomo installation:

<code>info <i>listname</i></code>	Returns information about the specified list
<code>which {<i>address</i>}</code>	Returns the listnames to which you are subscribed or to which the optionally specified address is subscribed;
<code>who <i>listname</i></code>	Returns a list of all addresses subscribed to the specified list name
<code>lists</code>	Returns the names of all mailing lists maintained on that particular Majordomo installation
<code>index <i>listname</i></code>	Returns a list of files associated with a particular mailing list.
<code>get <i>listname filename</i></code>	Returns a copy of the requested file name

Two additional commands are available to all users:

help	Returns a informational help file;
end	Used to terminate a sequence of one-line Majordomo commands, so that subsequent lines of a message will not be processed (like a signature, for example)

In addition to these "general user" commands are some available only to list owners:

`approve password subscribe|unsubscribe listname address`

enables a list owner to approve a request to subscribe to, or sign off from, a closed list; the password (set initially by the Majordomo manager) is necessary to validate that the command is coming from the list owner;

`passwd listname old_value new_value`

enables list owners to change their password;

`newinfo listname password`

enables the list owner to set or change the information returned about list as the result of an info command.

This may seem like a limited set of commands, but they are enough to maintain your basic Internet mailing list.

## Defining a Majordomo List

Installing Majordomo involves doing some UID and directory management, compiling some code, customizing a configuration file, and defining some alias values, the details of which are found in the documentation distributed with the package. Once Majordomo is installed, defining a list is relatively simple. At least four aliases are defined for each list: a pointer to the list of subscribers, a pointer to the list owner, a pointer to a handler for list requests, and a pointer to where requests are forwarded for any necessary approval. Sample aliases for a list might appear as follows:

```
list_name: :include:/majordomo/lists/list_name
```

```
owner-list_name: owner@somenode.com
```

```
list_name-request: "|/majordomo/wrapper request-answer list_name"
```

```
list_name-approval: owner@somenode.com
```

In addition to the aliases, several files must be created. In the previous example, `list_name` would be a file whose name matches the name of the list. This file will ultimately contain the addresses of the list's members. Along with it would be files named `list_name.passwd`, containing the approval password, and `list_name.info`, containing the list's welcoming information. Other variations are described in Majordomo's documentation, and sample configurations are included.

## Summing up Majordomo

Majordomo is a fairly straightforward package and is not at all overwhelming to install or configure. As with any e-mail operation, care must be taken to avoid generating e-mail loops and other nasty conditions. So far on the Internet, Majordomo seems to be proving itself as a useful list-management option.

## Listproc

Listprocessor (*listproc* for short) is a UNIX-based Listserv alternative that was developed by Anastasios Kotsikonas ([tasos@cs.bu.edu](mailto:tasos@cs.bu.edu)). Its command set is similar but not identical to Listserv. Listproc was a freely

distributed program through version 6, and you may still find version 6 on anonymous FTP sites (cs.bu.edu, for one). In 1994, the Corporation for Research and Educational Networking (CREN) purchased the ownership rights to Listproc and eventually issued version 7.0. This version is available at no charge to CREN members, with a yearly licensing fee for other nonprofit or for-profit organizations. More information about the availability of this software can be attained via e-mail from listproc-info@listproc.net or cren@cren.net.

## Listproc Commands

Listproc is a full-featured mailing list package that includes some list digest features and limited file serving. Both unmoderated and moderated lists are supported. Lists can be private or open. The commands shown in the following list indicate a higher level of complexity (as well as utility) than is seen in the previously discussed alternatives. The commands subscribe and unsubscribe once again show up in this command set. Notice that some of the Listproc commands will be familiar if you have any experience with Listserv (not a coincidence).

The following is a summary of Listproc 7.0 commands. The parameters in braces ({} ) are optional. A vertical bar (|) stands for OR. Where the meaning of a command is not necessarily evident, a brief description is provided in parentheses.

### Commands to join a list or modify your access:

subscribe *listname your name* -or- join *listname your name*

unsubscribe *listname* -or- signoff *listname*

purge *password* (Removes your address from all lists)

which (Finds out to which lists you are subscribed)

set *listname {option arguments}*

mail ACK|NOACK|POSTPONE|DIGEST

password *old\_password new\_password*

address *password new\_address*

conceal YES|NO

preference *preferences*

query *listname* (Returns your settings)

<b>Commands to retrieve list or general information:</b>	
help <topic>	
help listproc	(Receives information on Listproc)
help live	(Receives information on interactive connection)
lists {local global {keywords}}	
release -or- version	(Listproc version)
information {listname}	
recipients <i>listname</i>	(Lists subscribers)
review <i>listname</i> {short description subscribers}	

statistics listname {subscriber_addresses   -all}
---

run listname {password cmd {arguments}}
---

(Runs a special command)
--------------------------

### Commands to list or retrieve files:

index {archive|path} {password} {-all}

search archive|path} {password} {-all} pattern

(pattern can be a regular expression)

get archive|path file {password} {# of parts}

view archive|path} {password} {# of parts} (Views files on the screen when connected interactively)

fax FAX# archive|path file {password} {# of parts}

(FAXes files to specified number)

### Commands for list management:

{quiet} add listname password address name

{address name} ...

(Adds user or users—quiet specifies no notification to address)

{quiet} delete listname password address {address} ...

{quiet} set listname {option arguments} for address

{address} ...

purge password address {address} ... (Removes addresses from mailing list)

approve listname password msg tag # {msg tag #} ... (Approves messages for a moderated list)

discard listname password msg tag # {msg tag #} ...

configuration listname password {option {arguments}} ... (Sets list configuration options—send help configuration for a list of options and arguments)

edit listname password file {-nolock} (Retrieves a list file for editing)

hold listname password (suspend list mail delivery)

free listname password (resume list mail delivery)

lock listname password (suspend list requests)

unlock listname password (resume list requests)

initialize password {alias list\_address config\_options}

-or-

new-list password alias list\_address config\_options

put listname password keyword {arguments}

reports listname password

system *listname password user\_address commands* (Issues commands for users)

## Installing and Configuring Listproc

Installing Listproc does require system manager privileges to accomplish some tasks. Other than setting up the appropriate accounts, aliases must be set up, software must be compiled and configured, and file permissions must be set. A config file defines the parameters of the Listproc system, including elements like organization name, server definition and options, Listproc manager, and list definitions. The total number of parameters is quite numerous.

Lists are defined in the config file while the server is not running. Elements to define are a list alias (listname), list e-mail address, list owner's address, and a password. After they are defined, lists can be maintained by the Listproc manager through use of a list command supplied with the program distribution. Lists can be changed to moderated, message reply options can be set, message forwarding can be set, and several other parameter changes are possible.

## One More Word on Listproc

For a number of years, Listproc was the alternative system for UNIX sites that wanted to run a mailing list server. With recent releases the software seems to be coming into its own, and CREN is certainly hoping to establish it in a manner similar to the proliferation of Listserv. As the following sections show, however, it is not without competition, and continual addition of a features may be necessary for this package to catch up to the more firmly established mailing list system.

## Paco Xander Nathan

--by Tod Foley

"Overall, Rover represents an ongoing AI project, intended to explore the use of automated e-mail to help obviate most needs for a marketing department :)

Seems like OTHER magazines have even copied our research efforts, but we won't mention any names now, ahem, will we WiReD??!?!?!?!?"

--Paco Xander Nathan; speaking of the FringeWare Infobot "Rover"

Paco Xander Nathan, also known as pacoid@io.com, is a programmer, writer, lecturer, and exemplary entrepreneur from 15 minutes into the future. He is perhaps best and most widely known as the cofounder of FringeWare Inc., a decidedly alternative hardware/software vendor and publishing company that does a majority of its business online. FringeWare's customers, clients, and cohorts keep in touch with Paco and his partner Jon Lebkowsky via an e-mail list, an automated file server, an anonymous FTP site, a Gopher site, a conference on The WELL, and a newly-created Mosaic site. But that isn't enough for these fabulous furry fringe brothers—they're currently creating their own virtual building in Steve Jackson Games' "Metaverse" MOO (telnet metaverse.io.com 7777).

Paco is a master of UNIX and telecommunications programming, and a well-known cyberjournalist as well. He is the creator of a wide variety of programs, from the rudimentary AI "Pets" that serve his online interests to a digital lunar calendar designed to assist in the prediction of menstrual cycles. His written works have appeared in the pages of *BOING-boING*, *Fringe Ware Review*, *Mondo 2000*, *PIX-Elation*, *Whole Earth Review*, *Wired*, and *2600* magazines.

**TF:** How did FringeWare Inc. begin, and how is it structured?

**PXN:** Well, Jon and I noticed a fundamental shift in the nature of media + communities + entreprenuring + art, so we created FW as an experiment/survival-measure. Over the following two years, we've collected a nifty community of talented, like-minded folks—and, very importantly, alienated others whom we felt needed to be alienated. So, on the foundation, we're a Net-savvy publisher. That lends us cachet to maintain "property" within what might otherwise be considered "public" areas: a kind of "news service" (i.e., the e-mail list) with over 1000 primary and perhaps 5000 secondary subscribers worldwide. Plus, we have the worldwide circulation of our magazine, *Fringe Ware Review*, and these projects have prospered based on our formulation of Net-based economics and behavior. From this base, we've built another pillar: Our publications are registered in the Library of Congress and archived at an anonymous FTP site. On top of that, we've built means for access via Gopher and the Infobot. Atop that, we have an even niftier, more interactive method for access: our WWW pages. Altogether, we'll call this a "structure" :), at least until we come up with some better name. P'haps it's all the Infobot....

**TF:** So you've got a built-in base, which covers all available Internet services. How do you make use of this "structure" in day-to-day business?

**PXN:** We manage to slip in our product catalog at each level, along with public access to product info and discussions. This is all based on info-on-demand and not broadcast or direct marketing, so I don't see it as "evil commercialization of the Net." Rather, it's a new kind of marketplace; a formalized, nurtured version of what's been going on since the dawn of e-mail anyway. We also manage to slip in feedback loops: (1) the various intelligent agents that comprise our infobot also generate files within our archives; (2) the business itself generates part of the archives, since our product and vendor and store databases are automatically converted into WWW pages; (3) the community develops a narrative about the business and the infobot; and (4) the infobot's agents analyze and publish the narratives and participation within the community. So, due to the feedback and automation issues, this Thing, this Structure falls under the domain of cybernetics; for example, the use of the term "Infobot."

**TF:** The InfoBot—a "pet" you refer to as "Rover"—exhibits a high level of interactivity that some might describe as a rudimentary form of "artificial intelligence." Some of your other "pets" are known to wander around the Net, and to make strange interpolations and collections of words and phrases they encounter, as if attempting to interparse meaning into substance... and yet at root we're talking about a *mail server*. At what point did you first make that conceptual leap, in which a UNIX mailing list becomes the kernel of an intelligent agent?

**PXN:** I think the moment is defined when enough interest and attention focus on that mailing list or address as something of worth. Rover really is a pet, in one sense, because I "tend" it and sort of take it for walks—I like to have Rover walk in front and let people deal with it instead of me. Then I watch how they try to interact, and add functions to make Rover smarter. That fulfills a kind of feedback loop, in the sense of Norbert Weiner's definition of cybernetics. Some people have parsed our intentions; some even recognize how this Thing has become my "pet" in nearly every sense. But most haven't yet seen the real subtext of how it's only a half-step away from being interactive TV.

People try to reach us, and if Rover doesn't understand a message, it sends back a polite "I have no idea what you're saying, but here's what I understand" response. That fulfills another feedback loop. These loops provide the scientific basis for some kind of AI, turning the list+server into something more than a news pipe or TV broadcast. I could quote from Steven Levy's text on *ALife* [Artificial Life], but basically these lists/servers/agents are starting to fulfill the criteria.

**TF:** The implications are indeed staggering, especially when you bear in mind that the predominant trend in Net programming these days seems to be integration of services into integrated "environments." But at this point many people would ask, "Why go to all the trouble? Why hassle to create and continually modify a program that does things humans can do themselves?"

**PXN:** Because of the level of interest we've generated. There simply is no humanly possible way for us to have a real person answering all the questions. We couldn't afford that—probably ever. If you provide wires, people will

pull them. Telemarketing people with their phones and 800 numbers have *no* idea what's in store for them as the Net grows. We stick in a robot instead; it's a matter of numbers. I could personally respond to only, say,  $N$  people with any comprehension and still have time to put food on the table. So I and some associates have a list instead—that reaches  $N*500$ . But then even *more* people try to reach us whenever we get a magazine review, interview, etc., and rather than repeat ourselves silly, we add an intelligent agent. That covers, say,  $N*5000$ , provided we supply a help archive of files and hypertext for Rover to draw from.

I receive hundreds of calls/letters/e-mail each week, mostly from people asking for some kind of assistance. If any of those people really needed help they'd be dialing the local emergency number, or in counseling or somesuch, instead of being wired to a terminal. The volume makes all notions of communication with me—except for maybe a half-dozen of my closest friends—surreal.

-- from io.com/usr/ftp/pub/fwi/STAFF/pacoid.bio

**TF:** How do your customers and readers use the InfoBot?

**PXN:** Here's a summary of commands for Rover:

subscribe to join the list  
unsubscribe to leave the list  
list to list the available files  
get FILE to send you the file name "FILE"  
find KEY to send you a list of files containing "KEY"  
help to send you the intro file (same as "get readme")  
ping to check whether the list knows your address  
daily to receive daily msgs instead of weekly digest  
digest to receive weekly digest instead of daily msgs  
quit to quit scanning a msg, in case you have a .sig

For example, if you were to send the following message:

To: fringeware-request@io.com

get prices

Rover would send back to you a current price list for products listed in our online catalog (which can also be found in the back of **Fringe Ware Review**).

**TF:** What are some of your other "pets," and what do they do?

**PXN:** Well, there's Spewy and Chewy, who handle the basic mailing-list/news-service functions and digest service, respectively. I have one called Nosey, which uses various means to keep tabs on people I like to keep tabs on. I have another named Snoop, which runs the markov chain synthesis for us and updates our "memes" lexicon. It ends up building a lot of cool new words, which we like to think of as the "memetic interstices" of our discussions—a kind of quick-fix for our group narrative. There's a new one called Dopey, which has been learning how to parse error messages from all over the Net. If you think trying to correspond with  $N$  thousand people is weird, just wait until hundreds of those addresses start generating multiple bounces on a given day! I'm not aware of any other project like this, but it's a huge problem. As every new host plugs into the internetwork, more and more error reports and new formats spring up.

There's another pet, but it's been pulled for bad behavior. That was my personal agent, sniffer, spider, whatever the word is these days. It looks for any references to things I find interesting—great for research :)—but it chewed up too many system resources and locked up matters, and had to be suspended. Of course, we also have our Gopher and WWW and hope to put in a WAIS server (although we do that function via Rover now), which are all agents

in a sense. Source code for all our intelligent agents can be found in our archives.

**TF:** What sort of communications are not handled automatically?

**PXN:** Well, I have another personal agent called "Ron Lieberman" who gets politely nasty at times with people who bug me via e-mail....

=====

You can check out Fringe Ware Inc. using any of these methods:

FringeWare Email List:

mail fringeware-request@io.com

In text body:

subscribe

"Rover" (InfoBot):

mail fringeware-request@io.com

Include commands in text body.

Anonymous FTP:

...[ftp: io.com /pub/fwi](ftp://io.com/pub/fwi)

Gopher:

gopher io.com

Look under the "commercial" section.

The WELL:

go fringeware

Mosaic:

...<http://io.com/commercial/fringeware/home.html>

## Listserv

Listserv, written by Eric Thomas, is probably the most widely-used and established mailing-list software. This is true to the extent that many people use *listserv* as a synonym for any mailing list software, and even in reference to an individual mailing list (for example, "send a message to the Volkswagen owners' Listserv and may be you'll find out whether Beetle or Bug is the official name of that car"). Listserv is the most fully functioning of the packages discussed here. It includes mailing-list management, file services, and database services (not only can you store and distribute archive files, you and your Listserv users can search them using Listserv database capabilities).

Listserv supports interoperability between installations, allowing the production of a global list of mailing lists and subscriptions to any known mailing list via any Listserv installation. Listserv also has a "distribute" feature, which will send one copy of a message to an installation for distribution to those addresses maintained at (or close to) that installation.

For many years, Listserv ran only on IBM mainframe VM/CMS operating systems and was primarily associated with members of the Bitnet network. This association changed somewhat, when Dr. Thomas formed a company called L-Soft International, Inc., for the distribution of a new commercial version of Listserv and a VM/CMS mailer package called LMail. L-Soft has provided Listserv availability for those who are not part of Bitnet.

The first new development on the Listserv front was the release of version 1.8 for VM/CMS, which provided for operation of the Listserv software using TCP/IP protocols. The Bitnet network was originally (and still is, to a great degree) based upon the IBM Remote Spooling Communications Subsystem (RSCS) networking protocols. The Network Job Entry (NJE) portion of RSCS supports the exchange of mail and files on Bitnet for VM/CMS, MVS/SP, and VAX/VMS and other systems running RSCS-emulation software. (Today, most Bitnet traffic is carried via NJE under IP protocols, with transmission occurring mostly over the NSFNet Internet backbone.) The availability of Listserv 1.8 provides the ability to run Listserv independent of an RSCS network (Bitnet, for example).

The second development was the release of Listserv 1.8 for UNIX. The UNIX version will run on AIX, BSDi, Irix, OSF/1, Solaris, SunOS, and Ultrix systems. This version will interoperate with other versions of Listserv on TCP/IP. Existing lists based on VM/CMS systems can be ported to run under the UNIX version of the product. This release brings the Listserv feature set to platforms other than IBM VM/CMS. Versions for VAX/VMS and Windows NT are also announced. Any questions with regard to the availability of these products can be addressed to sales@lsoft.com.

The many functions of Listserv are reflected in its command set. Some of these commands may be familiar even if you haven't used Listserv before, since Listserv set a standard in this regard that has been copied by other programs. Because it has been the workhorse of the Bitnet network, Listserv also has the capability to provide information about the node structure of that network (via the SHOW command).

The following is a summary of Listserv user and owner commands through version 1.8. The parameters in braces ({} ) are optional. A vertical bar (|) stands for or. Where the meaning of a command is not necessarily evident, a brief description is provided in parentheses. Portions of the commands or options in uppercase represent the minimal abbreviations possible.

## User Commands

### Commands to join a list or modify your access:

SUBscribe *listname* *First\_name* *Last\_name*

SIGNOFF *listname*

SIGNOFF \*

SIGNOFF \* (NETWIDE)

SET *listname* (*options*

ACK | NOACK | MSGack

CONCEAL | NOCONCEAL

Files | NOFiles

Mail | NOMail

REPro | NOREPro

FULLhdr -or- FULLBsmtp

IETFhdr

SHORThdr -or- SHORTBsmtp

CONFIRM *listname* {*listname*} ...

Query *listname*

Query \*

REGister *First\_name Last\_name*

REGister OFF

Commands to find out information about a list, lists, or BITNET

nodes:

Help

INDEX *listname*

Info *topic*

Lists {*option*}

Detailed

Global

Global /*string*

SUMmary *node*

SUMmary ALL

SUMmary TOTAL

Query File *fn ft filelist options*

Query FLags

RELEASE

REView *listname* {(*options*)}

BY *sort\_field*

Country

Name

NODEid

Userid

BY (*sort\_field sort\_field*)

Countries

LOCAl

MSG

NOHeader

Short

SHOW

{*function*}

ALIAS *node* {*node*} ...

BITEARN

DISTRIBUTE

DPATHs *node* {*node*} ...

DPATHs \*

FIXes

LINKs *node* {*node*} ...

NADs *node node* ...

NETwork

NODEntry *node node* ...

NODEntry *node /string\*/string*

PATHs *node node* {*node*} ...

STATs

SCAN *listname string*

STats *listname {(options)}*

LOCAl

THANKs (check status of server)

**Listserv file-server commands:**

AFD ADD *fn ft filelist prolog* (Automatic File Distribution)

AFD DELete *fn ft filelist*

AFD List

AFD FOR *address* ADD|DEL|LIST (node administrators only)

FUI (File Update Information)

GET *fn ft filelist {(options)}*

PROLOG*text text*

GIVE *fn ft filelist {TO} address*

INDex {*filelist*}

PW *function*

ADD *pw*

CHange *newpw PW=oldpw*

DELete *oldpw*

SENDme *fn ft filelist {(options)}*

PROLOG*text text*

**Database and other commands:**

DATABase *function*

Search DD=*ddname* {ECHO=NO}

List

REFRESH *dbname*

DBase *function*

Search DD=*ddname* {ECHO=NO}

List

REFRESH *dbname*

DISTRIBUTE *type source dest options*

MAIL

FILE

RFC822

DD=*ddname*

{TO} *address* {*address*} ...

{TO} DD=*ddname*

ACK=NONE | MAIL | MSG

CANON=YES

DEBUG=YES

INFORM=MAIL

PRIOR=*nn*

TRACE=YES

FROM=*address*

FROM=DD=*ddname*

FOR *address command*

SERVE *address*

UDD (User Directory Database)

## Mailing List Owner Commands

### Commands used for file management:

AFD GET *fn ft* {*filelist*}

FUI GET *fn ft* {*filelist*}

GET *fn filelist* {*options*}

CTL

NOLOCK

PUT *fn ft {filelist parms}*

NODIST

CKDATE=NO

DATE=*yymmddhhmmss*PW=*pw*RECFM=F {LRECL=*nnn*}REPLY-TO=*address*

REPLY-TO=NONE

REPLY-VIA=MSG

TITLE=*file title*REFRESH *filelist {(options)}*

NOFLAG

UNLOCK *fn filelist***Commands used for list management:**{QUIET} ADD *listname address first\_name last\_name*{QUIET} ADD *listname DD=ddname*{QUIET} ADDHere *listname address first\_name last\_name*{QUIET} ADDHere *listname DD=ddname*{QUIET} DELeTe *listname address {(options)}*

GLobal

LOCAl

TEST

{QUIET} MOVE *listname address {TO} node*

```

{QUIET} MOVE      listname DD=ddname

{QUIET} SET       listname options {FOR address}

{QUIET} SET       *           options {FOR address}

                               NORENEW|RENEW

EXPLODE   listname {(options)}

                               BESTpeers n

                               Detailed

                               FOR node

                               PREFer node

                               SERvice

                               With(node {node} ...)

                               WITHOut(node {node} ...)

FREE      listname {(options)}

                               GLobal

GET       listname {(options)}

                               GLobal

                               HEADer

                               NOLock

                               OLD

HOLD     listname {(options)}

                               GLobal

PUT      listname LIST

Query   listname FOR address

Query   *           FOR address

STats   listname (RESET

```

UNLOCK *listname*

## Listserv Installation and Management

After it is acquired, Listserv will need to be installed by the system administrator. A separate individual can be set as the Listserv manager (the LISTSERV POSTMASTER), with the ability to create and modify mailing lists, and, if necessary, shut down and restart the Listserv software. List owners, once defined, also have a great deal of control over their list. A new list is created by the Listserv manager through creation of a file defining the list configuration parameters. The file name matches the mailing list name and the file type (CMS systems) is list. The LSVPUT command is used to create the new list within the Listserv system. Configuration options are the same as seen in the header of a REView command.

The following shows an example configuration for the Listserv mailing list NETMONTH, used to distribute an electronic periodical.

```
* NetMonth Magazine
*
* Newsgroups= bit.listserv.netmonth
* Review= Owner      Subscription= Open      Send= Owner
* Notify= No          Files= Yes
* Validate= Store Only
* Stats= Extended,Public  Ack=MAIL
* Errors-To= Owners
* Reply-To= "Dr. Philip Baczewski <NMONTHED@UNTVM1>",Respect
* Renewal= Yearly
*
* Mail-Via= DISTRIBUTE
*
* Local= MARIST*
*
* Notebook= Yes,E,Separate
*
* Owner= NMONTHED@UNTVM1 (Dr. Philip Baczewski)
```

- \* Owner= XXXXX@XXXXXX (A. Harry Williams)
- \* Owner = quiet: <XXXX@UNTVM1> (Dr. Philip Baczewski)
- \* Owner = <NMONTHED@vm.acs.unt.edu> (Dr. Philip Baczewski)

## A Closing Word on Listserv

Listserv is surely testament to the fact that a good idea, implemented well, will go a long way. With the Listserv mailing list count above 5000, Listserv's place within the arena of international networking is well-established. One factor in the success of Listserv has been its association with the Bitnet network. Bitnet, while in some ways technologically primitive, was easy and inexpensive to connect to and thus grew to include many U.S. colleges and universities. The free access to Listserv software by these sites led to the establishment of a service that has grown far beyond the Bitnet network. It remains to be seen what effect the commercialization of the Listserv software will have on the maintenance of its role as a leader in list-management services. The freeing of the software from the exclusive dominions of IBM's VM/CMS software environment, however, should have a positive effect on Listserv's continued proliferation.

## Some Aspects of Setting up Your Own List

There are several good discussions about setting up a mailing list available in print or electronic format (the file `LISTSERV TIP`, available from `listserv@bitnic.educom.edu` or the file or Diane Kovac's `LISTS START-UP` file available from `listserv@uottawa.bitnet`, for example). While these sources provide technical and organizational advice for starting a mailing list, there are several other factors which can be important. One of these is the type of forum you choose for your list.

### Choosing a Forum

The success of a forum may depend upon the intended audience and can range from the very controlled to the fully open. Your choice may depend upon the ratio of "signal" to "noise" that is tolerable by you and your subscribers. On a mailing list, "signal" is messages and information pertinent to the list topic, while "noise" is inappropriate or inadvertent posts (like signoff commands, messages like "Bill, got your message on the mailing list—How are the kids—," or comments like "Wow, how about that perfect game by Texas Ranger Kenny Rogers!"—except if that last one is sent to a sports-related mailing list).

The types of mailing lists, going from most signal to least, can be classified as follows:

Electronic journal	The most signal, because the content is generally controlled by an editor or editorial board; the least spontaneity and serendipity, since general posts are not allowed.
Mailing list digest	Incoming messages are screened, and the most pertinent messages are posted in groups to the mailing list.
Moderated list	Incoming messages are screened by a list moderator and only those pertinent to the list topic are distributed to list subscribers.
Open list	All incoming messages are redistributed to the entire list of subscribers (usually the most noise).

Not all open lists are noisy. If you have a group of dedicated and knowledgeable subscribers, the level of signal can be quite high. Some groups, however, will be unable to tolerate even the slightest level of noise. This seems to be true of college professors subscribed to scholarly mailing lists. Perhaps it's because a high signal level in publications is so intrinsic to the academic environment. Obviously, the intentions for which the list was created

will also dictate the tolerable level of noise. A general discussion forum provides more leeway than one on a very specific topic.

## Mailing Lists and Usenet

The interaction of mailing lists and Usenet news is continually on the rise. Often the largest possible audience will be available via both facilities. There are a number of ways that information can be cross-posted. Piping selected information from Usenet can be an important service for those without Usenet access. Providing mailing list access via Usenet allows people to read messages without having to subscribe and receive all messages.

### Usenet Digests

Perhaps one of the easiest ways to share information between Usenet and a mailing list is via a digest. Interesting Usenet posts can be gathered and redistributed to a mailing list. It takes the work of a digest editor to accomplish this task, but a news system's built-in features can make it a simple one (log or concatenate posts to a single file and redistribute that file to the mailing list, for example). Because of the extra work involved and the one-way flow of information, this technique is usually more appropriate for a special-interest area than a general discussion list.

### Usenet Echoes

Numerous mailing lists are echoed on Usenet or vice versa (chicken versus egg, you know). The largest single collection of these is located in the bit.listserv news hierarchy, which, as its name indicates, gateways numerous Bitnet Listserv mailing lists to Usenet. Mailing list and Usenet pairs also exist in the alt, comp (numerous), rec, soc, sci, and vmsnet hierarchies. David Lawrence (tale@uunet.uu.net) maintains a list of gatewayed newsgroups that is posted to bit.admin on a monthly basis. Jim McIntosh (jim@american.edu) at The American University runs a gateway for most of the bit.listserv hierarchy. That gateway is bidirectional, but other gateways might be one way only.

It is possible to set up your own gateway by having your news server subscribe to a mailing list the way a regular user would. There are some procedural items to follow. You will need to get permission from the mailing list manager (if it is not you) and also get appropriate approval from the Usenet hierarchy. It is also advisable to get agreement from the list members before expanding the distribution scope of their messages. More information in this regard can be found in a policy document that is regularly posted to bit.admin or by posting inquiries there or to the mailing list new-admin@american.edu.

## For More Information[e]

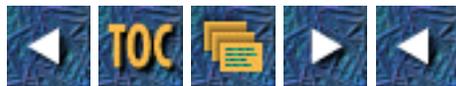
More information on mailing list software or its use can be found via the following Internet services:

Majordomo-Users on majordomo@greatcircle.com	
Majordomo-Announce on majordomo@greatcircle.com	
cren-list@yukon.cren.org	(Listproc 7 discussion)
unix-listproc on listproc@avs.com	
Istown-l@uga.cs.uga.edu	(Listserv mailing-list owners)
Istrev-l@uga.cs.uga.edu	(Listserv Review)
Istsrv-l@uga.cs.uga.edu	(Listserv discussion list)
Istsrv-e@searn.sunet.se	(Listserv evaluation package discussion list)
Istsrv-m@searn.sunet.se	(Listserv maintainer's list)

sales@lsoft.com
support@lsoft.com
UseNet: bit.admin
Usenet: news.lists

(Listserv lists posted to Usenet)





- [13](#)
  - [MBone: The Internet's Multimedia Backbone](#)
    - [--by Kevin Mullet](#)
    - [What's a Multimedia Backbone?](#)
    - [In the Beginning. . .](#)
    - [Doing It](#)
      - [SD: Session Directory](#)
      - [The Visual Audio Tool](#)
      - [NV: The Network Video Tool](#)
      - [WB: The LBL Whiteboard Tool](#)
      - [IMM: Image Multicaster Client](#)
    - [What's It Run On?](#)
    - ["Hey! Where'd Everybody Go?": Some MBone Troubleshooting Tips](#)
      - ["Are My Expectations Realistic?"](#)
      - [Is mouted Working?](#)
    - [Where Do You Go from Here?](#)
      - [MBONE Home Page](#)
      - [Frequently Asked Questions \(FAQ\) on the Multicast Backbone \(MBONE\)](#)
      - [The rem-conf Mailing List](#)
      - [MBone Contact Information](#)

# 13

## MBone: The Internet's Multimedia Backbone

[--by Kevin Mullet](#)

### What's a Multimedia Backbone?

Everybody's talking about it. Set foot in a trade show or a computer store, and someone will try and convince you that you simply must have it—after all, what's the use of having a computer that doesn't have it?

What is it? It's multimedia and one by one, it's turning all our computers into interactive televisions, stereos, telephones, shopping malls, arcade-quality video games, and talking, moving whatchamacallits that used to exist only in pulp science fiction. Okay—you could lead a perfectly normal life without it, but multimedia actually is pretty cool. Of the multimedia applications that exist in the here-and-now, real-time video conferencing is probably generating the most interest and industry.

Video conferencing is the ability to hold meetings of two or more people in geographically disparate locations without losing the immediacy, utility, and intimacy of a meeting in a conference room or a classroom. Just about anything you would expect to do in a conference or a class is fair game for video teleconferencing. With a good system, you should expect to see and hear all the other people in a meeting, draw on a whiteboard everyone can see, share documents, and brew a pot of coffee. Well, maybe not the coffee, but just about everything else is doable right now.

Armed with expectations about what a video conferencing system should do, you could easily drop five or six figures on a behemoth of a system that required specially-equipped conference rooms from which to use it (which might already lead you to say, "Why bother?"), dedicated leased lines, and possibly dedicated staff to cajole the whole thing into functionality. Such a system might be just the ticket for your needs. Some have found another way—*the MBone*.

The MBone is an international multimedia network within a network that runs over the Internet. Consisting of a backbone of special IP multicast routers that "tunnel" the multimedia information through the existing Internet infrastructure, the MBone is used every day by Internet users to exchange real-time or prerecorded video and audio—as well as with other, peripheral applications, such as a collaborative, online version of a classroom whiteboard, or a version of NCSA Mosaic customized for distance learning.

So what's the catch? Well, certainly a high-end system will get you flickerless video of around 20 frames per second, video-conferencing that looks as good as the promotional literature and videos, Hi-Fi sound, and no noticeable impact on your existing data network infrastructure. Great stuff, if you can afford it.

Compared to the Hollywood-epic quality of a high-end commercial system, the MBone is your aunt Molly's videos of her trip to Yellowstone. A typical MBone session is viewed in a window on the display of someone's desktop UNIX machine, and broadcast from a cheap fixed-focus camera and even cheaper condenser mike sitting on top of their monitor or on a spare corner of their desk. Equally likely is a broadcast of a professional conference or tutorial with views of the audience, the speaker, and the screen in front of the overhead projector. A good average video frame rate might be four or five frames a second, with occasional drops of service lasting from a few seconds to a few minutes, due largely to the fact that the MBone contends for the same physical resources as the rest of the Internet. Your participation in an Internet tutorial given from Tokyo may be interrupted because of a surge of Internet activity anywhere along the route between you and Japan. By the same token, your local telnet session to the host down the hall may take a performance hit because someone in the next office is watching a session in the current IETF meeting.

At this stage, the capacity of the MBone is fairly modest—one of its primary missions is still to serve as a testbed for Internet multicasts from the meetings of the Internet Engineering Task Force. On its main backbone, the MBone has bandwidth to support the equivalent of six audio or four video sessions. That's not six audio or four video sessions per person—that's the shared capacity of the 1200 or so networks that are connected to the MBone. Happily, the combination of existing social convention for restricting wide-area broadcasts to those of the greatest collective benefit—and the technical capability of MBone

tools to restrict the effective distance of any given broadcast—makes the network more usable than this threshold might make it sound.

Did I mention that apart from the cost of maintaining your Internet connection, and the load on your network, the attaching to the MBone and most of the software that uses it are free?

## In the Beginning. . .

The early seventies saw some of the first research in network-based audio on the ARPAnet. A standardized way of multicasting, the transmission of network data between defined groups of hosts, was needed at the network layer. While multicasting was available on such MAC-layer protocols as Ethernet and FDDI, the standard for multicasting with the network-layer Internet Protocol was defined by Steve Deering in 1989.

In early 1992 the Internet Engineering Task Force, the standards engineering body of the Internet, adopted Deering's Host Extensions for IP Multicasting for the Virtual Internet Backbone for Multicast IP, which got the name MBone later that year.

Additional protocols and applications were rapidly prototyped, refined, and put to use in such multicast events as live speeches by speakers ranging from Larry King to President Clinton. The locales of MBone multicasts have ranged from underwater off the coast of Baja, Mexico, to the Space Shuttle as it orbited the earth.

As 1994 began, 750 networks were attached to the MBone. By August, that number had mushroomed to 1200.

## Doing It

To participate in the MBone, your local-area network needs to be IP multicast-capable. For that to be true, a router on your local-area network must be an IP multicast router. In most cases, this means that you have a UNIX workstation of some kind running the mrouterd program, doing the actual tunneling necessary for MBone sites to interoperate on an Internet of largely non-IP, multicast-aware routers.

Actual use of the MBone means using one or more clients. Here's a survey of some of them.

### **SD: Session Directory**

Developed by Van Jacobson and Steve McCanne at the University of California's Lawrence Berkeley Laboratory, the Session Directory tool is a kind of graphical TV Guide to the MBone (see Figure 13.1). It doesn't necessarily display everything happening on the MBone, but SD does provide an easy-to-use interface for the main MBone applications.

### **[Figure 13.1. The Session Directory window, displaying information about an audio session.](#)**

The Session Directory window is split into two main areas and has a row of buttons on the bottom. The top area is a scrollable listing of the current events that are being advertised on the MBone. Events may be advertised any amount of time in advance. The bottom area displays information about selected events from the top area. The buttons permit opening (viewing/listening/participating), editing, and deleting of events.

Single-clicking on any particular event displays the description, multicast address, time-to-live, lifetime, media, creator, and creation date for that event. The description for an event is put in by the creator at the time of creation. The multicast address is a unique address that is shared by each participant in the event. The time-to-live value governs how far away that particular event will be visible to other persons on the MBone. The media is usually one or more of audio, video, or whiteboard.

Figure 13.2 shows the effect of selecting either the new or the edit buttons. In this event description window, you can define the parameters of an MBone event. Note the Scope area of the window. This permits you to set the TTL of the session so that it can only be seen at your site, or by the entire world.

**[Figure 13.2. The session editing window in SD, from which you create new sessions or edit existing ones that you've already created.](#)**

Here's a very important point: The viability of the MBone depends, in part, on the ability of its users to exercise discretion and restraint with regard to the TTLs of their sessions. The informal convention for determining if a session you want to broadcast has enough merit to multicast over a wide area is to send a posting to the rem-conf@es.net mailing list and get the opinions of your fellow MBone users.

The rem-conf mailing list, incidentally, is a good mailing list to be on if your thinking of getting on the MBone. A few weeks of lurking on the list should give you a good idea of whether the MBone is right for you. To join, send an e-mail message with your full name and preferred e-mail address to rem-conf-request@es.net.

## **The Visual Audio Tool**

The LBL Visual Audio Tool is used for audio sessions on the MBone. The window consists of a large area, listing the names of everyone else listening to, and optionally participating in, the audio session. On the right side are slider bars to control the volume and level for the speaker and microphone, respectively, as well as mute buttons for the mike, speaker, or both. VU meters dynamically show the output levels. See Figure 13.3.

**[Figure 13.3. VAT: Lawrence Berkeley Lab's Visual Audio Tool, is used for sending and receiving audio on the MBone.](#)**

Each of the names in the participant list has a checkbox next to it. Middle-clicking that checkbox will permit two listeners in a session to carry on a private conversation, out-of-band with regard to the rest of the session. LBL VAT also has the capability to DES encrypt the audio stream with a keyword—anyone attempting to hear what's going on in such a conversation should not be able to do so.

You can have an arbitrary number of VATs running at once, but because both you and your workstation probably have only one set of audio hardware, you can select which VAT can use your audio hardware at any given time by selecting the title bar of the window. The VAT with an inverse title bar is currently "active."

VAT also supports both a lecture and a conference mode. In conference mode, the VATs performance is tuned to be more conducive to interactivity. In lecture mode, VAT is tuned to make the most of the audio stream it's receiving at the expense of interactivity.

## NV: The Network Video Tool

Ron Frederick's Network Video tool is the principal means of generating and receiving video on the MBone. You will usually run this just as you would run VAT: by selecting a video session from the list of advertised sessions in the session directory tool.

### Figure 13.4. NV: Xerox PARC's Network Video tool for the MBone is used for viewing or broadcasting one or more video streams associated with an MBone session.

Once run, NV displays a video control panel with either two or three regions, depending on whether or not you are equipped with a video frame buffer. The top region has a thumbnail version of each of the video streams associated with a particular session. Frequently, there will be a video stream of the speaker, one of a projection screen, and maybe one of the audience. Beneath that is an area showing information about the current session as well as a place to put your name, each of which may be edited. If you are equipped with appropriate video frame grabbing hardware, a third section contains a slider bar that permits you to change your maximum bandwidth utilization while transmitting video, a toggle to switch between gray-scale and color transmission, and a place to pull up a list of people viewing your transmission.

Clicking in any of the video thumbnails in the control panel will produce a larger video window of the session, complete with a control panel of its own (which drops down if you click anywhere in the window). This control panel permits you to change the overall size of the video window, control brightness and contrast, toggle between grayscale and color, and capture the current image to another window for further processing.

NV uses a kind of "velocity compression" algorithm that gives priority to updating quickly changing sections of the video over others. The end result is that when you first bring up a video session, it may take a little while to get the entire picture filled in; but once it is, you'll get surprising good performance from relatively little bandwidth.

If there were seven deadly sins of video production, they probably get committed all the time on MBone video transmissions. A short list of things you can do to improve the professionalism of your video broadcast would include things like the following:

*Easy does it.* Probably the worst thing you can do in a video teleconference is hand-hold the camera, or constantly move it around. This is doubly so on the MBone, because every movement translates to that much more data being pushed around on the network, and that much less functionality for everyone watching a choppy picture on their workstation. Keep the camera on a support or tripod and pan, zoom, or otherwise move the camera only when absolutely necessary.

If you're multicasting from a conference and you've got one camera, it's probably much better to keep it on the speaker, and pan the microphone back and forth when questions are being asked from the audience instead of panning the camera. You'll miss out on getting the audience members in transmission, but your overall video quality will improve, and your impact on the network will drop significantly as well.

*Be enlightened.* Although there are situations where you may not be able to help it (such as when using an overhead projector in an otherwise darkened room), try to get the subject as evenly lit as possible. Lighting often seems to be the worst at someone's desk when they're broadcasting from a camera on top of their monitor, and their face is lit only by a reading lamp. You might want to experiment with turning on the

overhead lights or by pointing the lamp at a white wall to one side of your desk for better lighting. It might sound trivial, but paying attention to small details of production on "populist" media such as the MBone can vastly improve the end quality of your session.

*It's a slide, not a book, dang it!* This one is more an issue with slide- and foil-oriented presentations, but when it happens, it's the bane of any otherwise reasonable presentation. An overhead foil with a picture of the state of Texas and a list of every Internet site in the state in 8-point type doesn't do anyone any good. What might be useful information on the printed page is not necessarily useful information on a projected image. If that projected image is, in turn, broadcast on the MBone and viewed in a 320x240 or 384x288 video window on a computer display, the viewer may not even be able to deduce what state's on the slide—let alone that there's any other writing in the image. Reduce any slide to two or three simply stated ideas in large enough print to be readable in the end NV image. It may seem like over-simplification at the time, but the more direct and simply-stated you are, the more your audience is likely to remember the content of your presentation, rather than the fact that it just consisted of a bunch of slides no one could read.

An ideal answer to this, of course, is to develop your presentation online, then display it from a notebook computer splitting the video output between a display panel for the physical audience and an NTSC or PAL converter to go into a frame buffer. This enables you to display it directly to the Net using NV instead of pointing a camera at the projection screen.

## **WB: The LBL Whiteboard Tool**

Like the video and audio tools, WB, the multicast whiteboard, is also primarily written by Steve McCanne and Van Jacobson of the UC Lawrence Berkeley Lab. The whiteboard tool permits numerous persons to share a multipage/color/format virtual whiteboard over the MBone (see Figure 13.5).

### **Figure 13.5. LBL's Whiteboard Tool, WB, can be used for collaborative renderings.**

When you first run WB, it brings up two windows: a drawing window and a participant info window. The drawing window is where the actual whiteboard activity takes place. The smaller participant info window shows a list of everyone participating in the collaborative whiteboard session.

The participant info window has three main areas. At the top is a display of the most recently active participants. Below that is a list of everyone participating in this particular whiteboard session. Below that is displayed specific information about whichever participant you select in the middle, "participants," window.

Each of the participants listed in the middle window has a checkbox next to their name, and may occasionally have a pound-sign between their name and the checkbox indicating that their display hasn't been fully updated yet. If you select one of the names on this scrollable list, the bottom part of the window will display such information as the the originating IP address of the user, the amount of time they've been idle, the number of operations they've performed, and so on. If you check the checkbox next to their name, you will remove all the marks they've made on the current page from the display.

The drawing window consists of three main areas: the drawing surface, a page manipulation area, and the actual drawing tools. The drawing surface displays one of an arbitrary number of pages that you and your fellow session participants have created during the current session. The remote drawing surface doesn't

display your mouse pointer—only the drawings that you've created—so if you want to point to something, you have to indicate what you're pointing at by circling it or drawing an arrow pointing to it.

The page manipulation area includes buttons to create new pages, import PostScript or plain text, clone a page, or scroll through the pages you've produced so far. When you create a new page, the old one is effectively pushed on a stack, where you can scroll back to it. The postscript import ability permits you to incorporate more complex, higher quality graphics into your session than can be created with WB's drawing tools. Simple text import is convenient, especially for quick-and-dirty presentations where you've prepared text elsewhere. It's important to note that when any session participant scrolls to a given page, all whiteboards scroll to that page. You can suspend this action by clicking on the page number next to the scroll bar. To return to the default behavior, click once again on the page number.

The drawing tool area of the drawing window contains all the tools necessary for conventional whiteboard sketches. At the top of the area are the drawing tools for doing freestyle lines, straight lines, arrows, rectangles, and ellipses, as well as an eraser tool and tools for moving and copying objects on the whiteboard. Next are two drop-down menus: one to choose the text font, and one to select the brush size. After that is a palette of five brush colors and a set of three icons to permit you to change your overall page orientation.

Of all the main MBone tools, the whiteboard is the one that probably takes the most getting used to. Although it's analogous to a physical whiteboard, there isn't a common analog to the collaborative nature of the whiteboard—unless it would be a table full of people all drawing on the same pad of paper. It's probably exactly this uniqueness that makes WB one of the really great applications on the MBone. Once you get the hang of sharing the whiteboard with someone, you'll realize it's a real natural next to VAT and NV.

## **IMM: Image Multicaster Client**

One thing you're apt to notice as soon as you get LBL's Session Directory application up and running is that it's always populated with sessions that have names like GMS-4 Composite, GOES-7 IR 1, and IMM-Psychedelic Images. The first two are a continuous stream of real-time satellite imagery that is multicasted from Hawaii, and the third is a stream of images from the University of Oslo in Norway simply described as a collection of colorful images. See Figure 13.6.

### **[Figure 13.6. IMM: An image-multicasting client from Winston Dang at the University of Hawaii.](#)**

Imagery on the MBone isn't all real-time video for conferences and meetings—some types of data (and yes, some types of recreation) lend themselves well to a fairly low-bandwidth distribution of graphical still images that usually find their way to the root window or background of your display. IMM is the client that receives the multicasts put out by its corresponding server, IMMServe. These transmissions are composed of a series of images in either JPEG or GIF format.

Usually started from a session advertised in SD, the IMM client's display is quite straightforward. It tells you the name of the image you're viewing, displays the progress as you download a new image, and tells you approximately when you can expect the next image to be downloaded.

Although IMM can be told to download the images to a script, or to hang on to the last few files as it receives them, its default is to put the images on your root window.

There are lots of additional things to do on the MBone, including setting up a World Wide Web multimedia-on-demand server and using a multicast version of NCSA Mosaic for distance learning. Such applications are proof that for a testbed network, the MBone certainly does provide a great deal of functionality.

## **What's It Run On?**

Currently, most MBone-connected machines, both mrouers and clients, are some form of UNIX workstation. A large number of those are Sun SPARCstations. A brief list of systems that can interoperate on the MBone includes those running AIX 3.2.5, BSD/386 1.1, DEC OSF/1, HP/UX 9.01, IRIX, NetBSD 0.9, Solaris 2.3, SunOS 4.1.n, and some versions of Ultrix. All static lists like this, though, are out of date as soon as they're printed; you should certainly check the more dynamic lists on the Net before making any decisions about getting on the MBone.

## **"Hey! Where'd Everybody Go?": Some MBone Troubleshooting Tips**

The more amazing things get on the Net, the more amazingly convoluted troubleshooting becomes. The MBone is no exception to this rule. To listen to some real MBone jocks dig into these things, you might swear they were speaking in tongues. Multicasting mysticism notwithstanding, then, here's some things you might expect to do if you bring up your own IP multicasting subnet.

Probably the first thing you should do when you start troubleshooting your IP multicasting is to verify your baseline Internet connectivity. Check other services that you know should work and use much the same route. If they're down or performing poorly, you may have a general IP connectivity problem instead of a specific multicasting problem.

### **"Are My Expectations Realistic?"**

If you've got a T1 or better route to the Internet backbone, and the machines acting as your multicast routers aren't underpowered, and there's not a great deal of other traffic on the local-area networks around them, and you've gotten better performance in the past, then you've probably got good reason to be suspicious if you suddenly get poor performance. If, however, you just brought up your IP multicasting router for the first time and started watching your first network video session, but it doesn't look like broadcast-quality TV, don't be surprised.

One of the tricky things with technology like the MBone is to learn to separate your hopes from any reasonable expectations you may have. One way to do this is with the help of your fellow session participants. If your MBone setup is working at all, you can use the statistics present in most of your clients to compare notes with other users. Decorum dictates that when you do so, you ought to wait until a lull in the action, or try and get someone's attention out-of-band through the private-conversation feature of VAT.

Once you do, you can ask your MBone neighbor if he or she has been getting performance similar to yours, or if your numbers are within the range of reasonable performance. If your numbers are consistent with those of similar sites, that's bad—because that means that your performance has probably gotten about as good as it's going to get. If your performance statistics are uncharacteristically poor, congratulations! You get to go to the next level!

### **Is mroued Working?**

If nothing works at all—if you have a total failure of multicast routing at your site, then you've probably got a problem that's moderately easy to fix. It's the intermittent problems that resist repair.

A good first step would be to make sure that mrouterd is running on your network's multicast router. Mrouterd is the software that "tunnels" the IP multicasting traffic through intervening conventional routers on the Internet. It's necessary because very few router vendors provide the IP Multicast routing support necessary to do straightforward multicasting without using tunnels.

Check and see what the specific syntax is for your system, but here's one possible command to see if your host is currently running the mrouterd software.

```
%ps -uaxw|egrep '(^USER|^root.*mrouterd)'
```

```
USER          PID %CPU %MEM    SZ   RSS TT  STAT  START   TIME  COMMAND
root          21134  0.0  1.3   400   360 ?   S     Aug  5  99:44 mrouterd
```

```
%
```

If mrouterd isn't running, that's one (but not necessarily all ) of your problems. Restart it or, if you're not the sysadmin, find out if it was stopped on purpose before asking that it be restarted.

You can also generate a dump of the mrouterd's routing table by sending it the signal USR1 with the kill -USR1 command, and examining it to see if mrouterd has acquired a healthy number of routes. In the previous example, the process ID number of mrouterd is 21134, so the steps of generating a routing table dump and checking it for some real elemental information might go like this:

```
#kill -USR1 21134
```

```
#ps -uax21134
```

```
USER          PID %CPU %MEM    SZ   RSS TT  STAT  START   TIME  COMMAND
root          21134  0.0  1.3   412   376 ?   S     Aug  5  99:51 mrouterd
```

```
#ls -la /usr/tmp/mrouterd.dump
```

```
-rw-r--r--  1 root          103801 Aug 16 22:31 /usr/tmp/mrouterd.dump
```

```
#wc -l /usr/tmp/mrouterd.dump
```

```
1193 /usr/tmp/mrouterd.dump
```

```
#head /usr/tmp/mrouterd.dump
```

Virtual Interface Table

```
Vif  Local-Address                                Metric  Thresh  Flags
```

```

0    128.241.0.92    subnet: 128.241.0.80        1        1
                                peers  : 128.241.0.91
                                groups: 224.0.0.4
1    128.241.0.92    tunnel: 192.225.25.1        1        64
                                peers  : 192.225.25.1
2    128.241.0.92    tunnel: 129.162.150.145     1        16
3    128.241.0.92    tunnel: 129.7.1.1           1        64
#

```

In the preceding example, we accomplished the following five steps:

- Used the `kill` command to send the `USR1` signal to the `mROUTED` process to generate a multicast routing table dump
- Used the `ps` command to ensure that the same `mROUTED` process is still around and survived the `kill` command—which it would if it was done correctly, but it never hurts to make sure.
- Used the `ls` command to check the date on the `mROUTED.dump` file to ensure that it's the one we just generated—which it is.
- Used the `wc` command to count how many lines there are in the `mROUTED.dump` file, which will be roughly analogous to the number of routes known to `mROUTED`. There are 1193 lines in the file, which means there's probably a good healthy routing table.
- Used the `head` command to peek at the top of the file to see if the virtual interfaces defined are what we expect from our definitions in the `/etc/mROUTED.conf` configuration file.

There are a number of things we can learn from this test.

If `mROUTED` wasn't running, and multicast routing started working after it was restarted, it's likely the problem was that `mROUTED` died for some reason, bringing down your multicast routing. A simple test to see if IP multicast packets are making their way to your subnet is to see if any sessions show up in the session manager.

If `mROUTED` was running, but your routing table is empty, the kernel modifications to your `mROUTED` host may not have been done properly. Double-check your procedure, or check with the `sysadmin` to see what kernel you're really running on the host in question. It's also possible that you may not need to modify your kernel to do multicasting—some operating systems are becoming available with multicasting from the factory.

If you've verified that you're running the right kernel, work your way upstream and see if the `mROUTED`

upstream of you is functional. One easy way to accomplish this is to run the MRInfo gateway and use a WWW browser to walk through the MBone, traipsing from one mrouted to another, noting which tunnels are up and which are down.

In the following example, we use the UNIX curses WWW browser Lynx to retrieve the tunnel information from the multicast router tempeh.sesqui.net:

```
%lynx -dump "http://www.cl.cam.ac.uk/cgi-bin/mrinfo?tempeh.sesqui.net"

Mrinfo for 128.241.0.92 (tempeh.sesqui.net) [version 2.2] 2 mins 27 secs

128.241.0.92 -> 128.241.0.91    vinegar.sesqui.net          1/1
128.241.0.92 -> 192.225.25.1    ASBESTOS.MFSDATANET.COM   1/64/tunnel
128.241.0.92 -> 129.162.150.145 keiko.space.swri.edu      1/16/tunnel/down
128.241.0.92 -> 129.7.1.1        Masala.CC.UH.EDU          1/64/tunnel
128.241.0.92 -> 128.83.5.252      janus.gw.utexas.edu       1/64/tunnel
128.241.0.92 -> 131.194.132.1     earth.mms.trinity.edu     1/64/tunnel/down
128.241.0.92 -> 129.110.10.14    squirrel.utdallas.edu     1/64/tunnel
128.241.0.92 -> 192.231.8.130      192.231.8.130            1/64/tunnel
128.241.0.92 -> 129.120.1.39        hermes.unt.edu            1/64/tunnel
128.241.0.92 -> 128.249.27.73     PAVLOV.SSCTR.BCM.TMC.EDU 1/64/tunnel
128.241.0.92 -> 129.118.18.43     sparc5.cs.ttu.edu         1/64/tunnel/down
128.241.0.92 -> 198.65.128.14    tattoo.sccsi.com          1/64/tunnel
128.241.0.92 -> 198.65.128.11    bsdi.sccsi.com            1/64/tunnel/down
```

According to the above test, four of the configured tunnels are down for some reason. If you did this test on your mrouter and found that a tunnel you expected to be up was down, it would give you a good starting place to make some calls and see what's going on.

Also, as you can see from the URL above, this demo of the mrinfo gateway is in the UK. It's a good idea, if you expect to use it with any regularity, to download the gateway and run it locally.

Two other programs should be mentioned because of their usefulness in multicast troubleshooting: ipmcast and ipwatch. Ipcast dumps continuous output to your screen describing virtually every multicast packet on your network segment in real time—a great utility to see if you've got any multicasting going on at all and, if so, who's doing it. Ipwatch has a much more structured, full-screen display but gives less information—it tells you how much of the traffic on your Net is devoted to what broad category such as

multicast, Net news, X Window, Ftp, mail, and so forth. Both of these utilities are well worth keeping in your bag of tricks, to arm yourself against any potential Net problems you encounter in your Mbone travels.

## Where Do You Go from Here?

The first step is probably to find out more about the Mbone, its characteristics, and limitations and see if you still want to take the plunge. Here is a list of Internet online resources that will help you do just that.

### MBONE Home Page

URL: <http://www.eit.com/techinfo/mbone/mbone.html>

This is the top of a considerable tree of information about the Mbone. Topics range from a general "What is the Mbone?" to specific information on debugging the Mbone. If you have time to check out only one resource about the Mbone, this would be a good choice. Most of the stuff in this section is actually linked to this page.

### Frequently Asked Questions (FAQ) on the Multicast Backbone (MBONE)

URL: <ftp://isi.edu/mbone/faq.txt>

This is good, meaty starting place for essential information on the Mbone. A nice down-to-basics document, this FAQ list by Steve Casner doesn't repeat itself. Read it a couple of times. Once you've got the stuff in this document down pat, you've significantly reduced the chance for surprises down the road.

### The *rem-conf* Mailing List

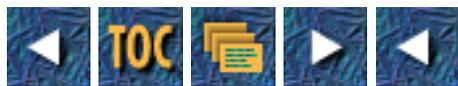
URL: not applicable.

Membership in this list is pretty much a must-have for folks who want to know what's up on the Mbone, what kind of events are likely to be scheduled, and what the various social conventions are in Mbone culture. Definately a good lurk list. Send a note to the human being at [rem-conf-request@es.net](mailto:rem-conf-request@es.net) and ask to be added.

### MBone Contact Information

You don't want to know anything else about the Mbone—you know what you want, and it's to get on the Mbone yesterday. This is way cool Infobahn stuff, and you want it right now. Okay—here's a list of organizations, e-mail addresses, and phone numbers. Happy multicasting!

URL: <http://www.eit.com/techinfo/mbone/contacts.html>





- [14](#)
  - [Community Computing](#)
    - [--by Kevin Mullet](#)
    - [Overview, Definition, and History](#)
      - [Freenets and the NPTN](#)
      - [Other Communities](#)
    - [Mission, Content, and Scope](#)
      - [A Truckstop on the Information Superhighway](#)
      - [Content](#)
      - [A Few Applications](#)
      - [Form](#)
    - [The Problems of Reaching Everyone](#)
      - [Networking in the Hood: Is Inner-City Networking Appropriate?](#)
      - [Accomodations for Differently-Abled Persons](#)
    - [The "F" Word: Funding](#)
    - [National Organizations](#)
    - [Three Examples](#)
      - [For More Information. . .](#)
        - [Gateway to Freenets and Community Computer Networks via World Wide Web](#)
        - [The Freenets Home Page](#)
        - [Community Network Guide](#)
        - [Back Issues of Networks and Community Newsletters](#)

# 14

## Community Computing

[--by Kevin Mullet](#)

## Overview, Definition, and History

New technology has always prompted the growth of communities. From the first gatherings of early man around campfires to later encampments around radios and televisions, technology has frequently influenced human community.

Nowadays, communities are sparked by a new kind of campfire—the difference being that such communities aren't bound by temporal or geographical restrictions. Most residents of such communities are rarely sure of the race, religion, age or even gender of many of their neighbors. As communities go, community computer networks are probably the most egalitarian.

### Freenets and the NPTN

In the mid-eighties, just about the time the Internet had just crossed the 1000-host threshold, Dr. Tom Grundner pressed an Apple ][+ into service as a BBS to link Case Western Reserve University's Family Medicine department to a number of remote clinical units. Somehow, the telephone number for this tiny departmental BBS got loose, and today's community computing movement was born. Like other movements, the community computing movement also had several birthplaces and sets of parents, but the events leading from this humble Apple ][+ are particularly significant because they would eventually lead to an international Freenet movement.

Once the dialup number for the Apple ][+ system was discovered by folks in the local community of Cleveland, Ohio, the BBS started receiving a variety of medical questions, in the hopes that they would be answered by one of the local physicians. Prompted by the intrepid sense of community of this situation, Grundner wrote a BBS system named Saint Silicon's Hospital and Information Dispensary, put it online, and wrote the experience up in *The New England Journal of Medicine*. The article and the success of the system attracted the attention and money of AT&T, Ohio Bell, and University Hospitals of Cleveland. These institutions supported an expansion of the system to an AT&T 3B2/400 with a ten-line dialup rotor and an expanded scope that included forums on government, arts, science, and education, as well as an e-mail platform for all users (see Figure 14.1). In the summer of 1986, the first Freenet was born.

#### [Figure 14.1. The Cleveland Freenet: The original Freenet is still running today.](#)

By 1989, Freenet II, as it was then called, consisted of multiple hosts sharing common storage and dialup facilities, which served well over 2000 logins per day, with many times that in registered users.

Well, you can't keep a good idea down—not that anyone tried. The idea of a freely available community computing system was too good to pass up for Youngstown, which opened their Freenet one year after the Cleveland system came online. 1990 saw three additional Freenets: Cincinnati's Tri-State Online, Peoria's Heartland Freenet, and the Medina County Freenet.

Soon, the National Public Telecomputing Network was formed as an umbrella organization for all Freenets, and by July of 1994, there were 37 Freenets online and 117 organizing committees in 41 U.S. states and 8 countries, either running existing systems or gearing up to do the same. No modest feat.

### Other Communities

Just as there are communities not bounded by a formal government, or categories like "city," "township," or "country," there are certainly online communities that are not Freenets. NPTN-affiliated Freenets get so much attention because they are perhaps the best organized of the lot, but a community computing system need not be affiliated with any other organization to make a meaningful contribution to its host community.

As long as there are computers and telephones, there will likely be dialup BBS systems. The fact that little Johnny or Judy can run one of a number of freely available programs on their new personal computer, pull the RJ/11 jack out of their phone and into their computer, and be information providers on the Infobahn is overwhelmingly attractive to kids of all ages. The medium really is the message.

The Cleveland Freenet got its start as a simple dialup BBS, much like the one in Figure 14.2. Before local-area networks were as prevalent as they are today, systems like this were most people's introduction to computer communications—which, for many people, soon led to stronger coffee, Jolt Cola, or other methods of trading an evening of sleep for one of downloading, chatting, or sending and reading e-mail.

**Figure 14.2. The humblest of community computing systems remains the single-line BBS.**

Many dialup BBS systems are joined in a network called FidoNet. While the Internet uses fairly upscale technologies, FidoNet nodes throughout the world are linked by telephone calls between systems and the (usual) cooperation of BBS sysops within any given local calling area. While FidoNet now has links to other wide-area networks, the mainstay of Fido traffic is still mail messages that are exchanged through modem calls from the sending to the receiving system—wherever in the world it happens to be, as well as an information superh. . .well, maybe an information farm road of Fido echoes that conceptually resemble Usenet news groups.

Another source of communal computing are the commercial providers such as America Online (see Figure 14.3) and CompuServe. One difference between these providers and a Freenet or raw Internet service is that these forums are highly structured affairs that may be quite reassuring to some users, and not to others. A downside of communing on such a system, though, is that because of the structure, the participants have somewhat less control over their environment than do the users of other systems. The flip side of that argument, though, is that the interface is usually reliable, and doesn't hold any surprises—letting you get right down to business once you log on.

**Figure 14.3. Commercial providers like America Online may provide a reassuring sense of structure.**

## **Mission, Content, and Scope**

Although such infotainment communities might seem to be marauding bands of Gopher, Web and Archie servers, depleting your free time and mind, most of them do actually have a rhyme and reason. The mission, content, and scope of a community computer system is what separates a profoundly meaningful communal experience from the trivialized subversive fad portrayed in the popular press.

## **A Truckstop on the Information Superhighway**

On the Information Superhighway, in a world of megamedia on demand and virtual theme parks, a community computer system is a kind of community diner. It's the kind of place where you can fill up on sweets just as easily as you can get legitimate nourishment—and you can catch up on all the local news and gossip and even learn something while you do so.

Strictly speaking, a community computer system is a computer system that is available for use by, and closely resembles in content, the community in which it is used. "Big deal," a curmudgeon might say. "The last thing I need to do is to play on a computer to find out about where I live." Actually, the idea's not as preposterous as it might sound.

Along with questionable air and water, vending-machine food that will still be fresh when the national debt is paid off, and ten-lane freeways that permit us the pleasure of driving home from work for two hours at ten miles an hour, another one of those nasty little side effects of the industrial revolution is that so many of us are packed into huge metropolitan areas where we spend more time contending for scarce resources than we spend getting to know each other. Community computing systems give us the ability to leverage opportunities to do so from online networks, as well as equally valuable opportunities to participate in democratic exchanges with our elected officials and exchange expertise with folks we might never otherwise meet or talk to.

## Content

There are essentially two approaches to organizing the content of a system like this. One is to organize by the source of the information; another is to organize by topic.

The advantage of organizing a system by source is that it scales very well. As long as you can find somewhere to pigeon-hole a new information provider, adding new information is very much a follow-the-recipe kind of task. With the exception of the occasional arguments about who gets space at the top level of the hierarchy, source-sorted systems are easy to manage. The downside is that like with most systems, the easier it is for the sysadmin, the harder it is for the user. Joe regular user will be able to find organizationally related information easily enough, but the real strength of such a system lies not just in the ease of use, but in the ease of browsing. The test is whether or not someone can come in off the street and quickly find some information they want—and do so in an intuitive manner.

Conversely, the advantage of organizing by topic is that users are much more likely to be able to find something when they want to. The difficulty in keeping up with such organization increases exponentially, though, as more information providers are added.

One possible compromise may be to organize by source, but index the contents of the system well enough that, armed with a few appropriate keywords, the user can go quickly to anything they want. An added feature of personal *bookmarks* or *hotlists* will further ease the user's journey by permitting them to keep a personalized list of their frequent haunts on the system.

## A Few Applications

The possible applications that would fit into a community computing system are much too numerous to list here. There are a few that stand out, though:

*The arts.* Arts events calendars, discussions of visual and performing art forms, virtual art galleries,

exchanges of musical compositions through MIDI files or collaborative composition online.

*Civic organizations.* Everyone from the Lion's Club to the Shriners, from the Red Cross to the United Way, has services and information they have a vested interest in getting out to the public.

*Consumer information.* Recalls, discussions of personal consumer experiences, gateways to state and national consumer watchdog groups, and information clearinghouses.

*Distance education.* Taking any variety of courses online; contributing to the communication channels of existing schools; students turning in homework or getting tutoring online; students in different states or countries collaborating on class projects; adult study material for civil service, LSAT, SAT, ACT, GRE, or other entrance exams; adult literacy assistance in the most anonymous and private of environments.

*Electronic mail and messaging.* Members can exchange electronic messages in both letter format and in real-time conversational format with other members of the community network, as well as with persons throughout the world on such platforms as SMTP mail or Internet Relay Chat.

*Employment.* Access to the plethora of job databases available on the Net, as well as the growing number of employers who are looking to the Net for their next employees.

*Fun and games.* Let us not forget the real reason we bought our first PC so many years ago. Games are actually one of the best ways to acclimate new initiates to computers; why not use them to encourage new community network users as well?

*Missing and abducted children.* This is a lot more effective than posters in grocery store windows. If you need to get the word out about a missing person, how about getting it out to the millions of people throughout the country who use the Internet every day?

*News from the source.* Many newswire services are already available on the Net. Many more original sources of news are available. During the Russian White House revolt, the full text of speeches made by Boris Yeltsin from the top of a tank were transcribed and posted to the Net in both Russian and English within minutes of their delivery.

*Political debate.* The core beliefs of today's mainstream were the beliefs of the radical fringe not all that long ago. Online political debate can bring the best of that process to your living room without the placards and bullhorns.

*Public health education.* Outside the Beltway, the president's plan was known to the Internet community first. The nitty-gritty of the human genetic-mapping project is taking place on the Net—it's not practical to do so in any other way. The ounce of prevention that's worth a pound of cure would fit nicely on a community networking system.

*Religious forums.* Usenet News: where the crusades are fought and refought every day. On the flip side, there are probably few better ways to assemble a group of persons who have beliefs exactly like yours. Meditation and fellowship, or proselytizing the unwashed masses. . .take your pick.

*Research on everything from schoolwork to geneology.* One of the original design goals of the Net. It's no surprise that research is still a primary preoccupation of the Net.

*Special interest groups.* One of the nice things about large community networks is that you're likely to

find other people who actually share your passion for *The Donna Reed Show*.

*Teledemocracy.* Every major constitutional document in full text; majority and minority supreme court opinions within minutes of their release to the public; every possible spin on every discussible issue; transcripts of every presidential and many cabinet-level press conferences; photo ops and the like; direct 1:1 interaction with each of your local representatives; minutes and agenda for precinct, school district, and city council meetings.

*Weather information.* Where is that hurricane and where is it headed? Exactly where did that quake in California hit, and how big was it? Should I take my umbrella today? Do I need to wrap my pipes?

## Form

An important distinguishing feature of civic networks is their choice of front-end software. The best choice depends on who the intended audience is and what services are offered. Three good candidates for front-end platforms for a community networking system are Freeport, from Case Western Reserve University, and Gopher or WWW platforms freely available from numerous sites on the Internet.

Figure 14.4 shows the Freeport software being used on the Buffalo Freenet. With its text-based menu-oriented approach, Freeport offers an easy, consistent interface and is well known, used, and supported throughout the Freenet community. An ongoing concern of community computing sysops is that they support a "lowest common denominator" interface, which usually equates to anything usable from a terminal or terminal emulator that supports DEC VT-100 without any of the niceties like double-width or double-height characters or alternate character sets—and Freeport is a champ at doing just that. A probable reason why so many Freenets use the Freeport software is that there's little (if any) risk associated with using it for community computing applications. With all the systems out there using it, Freeport can easily be regarded as a robust and battle-tested front-end solution.

### [Figure 14.4. The Buffalo, New York, Freenet through the CWRU Freeport software interface.](#)

Community computing is about more than accessing local resources, though—and access to regional and global resources is becoming increasingly important to local community system operators who want to offer their users citizenship in the larger global virtual village. Freeport does offer access to Gopher and Usenet news resources, as well as nearly anything else a creative sysadmin is able to leverage into the Freeport platform, but the real strength of Freeport is in managing local discussions, not presenting Internet-wide resources.

Gopher-based systems are an interesting alternative to the traditional Freeport approach. As of this writing, Gopher servers are available for MS Windows, Macintosh, and UNIX systems, although the most robust and scaleable servers are usually run using the University of Minnesota UNIX Gopher server and browsed using one of the many clients available. An advantage of Gopher is that while it is certainly doable to associate a UNIX text VT100-compatible client with a special-purpose userid, as done in many Freeport installations, users with IP-layer access to the Internet on UNIX Curses, X Window System, Macintosh, DOS, DOS w/MS Windows, NeXTstep, VM/CMS, OS/2, or MVS/XA platforms can all enjoy full access to any available Gopher in the world, using a client in their native operating environment.

Until recently, a valid complaint about Gopher was that it was read-only, which greatly restricted the number of applications Gopher could be used for in a community computing system. An extension to the Gopher protocol, Gopher+, does include more complex objects such as forms, and support for alternate character sets. Gopher, however, does have the characteristic that it presents everything as a menu. Some people like this; others prefer more flexibility. Perhaps a World Wide Web solution would be the ticket here (see Figure 14.5).

### [Figure 14.5. The Twin Cities Freenet viewed through the graphical NCSA Mosaic World Wide Web browser.](#)

Although the initial World Wide Web systems were brought online at about the same time as the initial Gopher systems, the growth of Gopherspace greatly outstripped the World Wide Web until the release of a critically-hailed WWW browser, from the National Center for Supercomputing Applications, called NCSA Mosaic (see Figure 14.5). The added usability and downright fun that Mosaic incorporated into the WWW infused new lifeblood into the Web, providing the stimulus for an explosion of new clients, servers, and standards. It is now difficult to say with certainty whether Gopher or the WWW are responsible for more Internet use and growth.

One of most distinguishing features of a WWW browser is that it is not constrained by the paradigm of a text menu. A Web page can be a scrollable document consisting of text or graphics which is "anchored" to other resources on the Net such as additional documents, Gophers, FTP sites, databases, audio, and video. Much of this requires a sizable bit of bandwidth, but there are alternate WWW browsers (such as Lynx, from the University of Kansas) that are perfectly suitable text-mode clients and can adequately meet the requirement for a lowest-common-denominator interface (see Figure 14.6). More complete information about what the World Wide Web and Gopher have to offer can be found in Chapters 10 and 18, respectively.

### [Figure 14.6. An alternative view of the Twin Cities Freenet through the University of Kansas' WWW client, Lynx.](#)

## **The Problems of Reaching Everyone**

It probably seems obvious that the larger a community gets, the more diverse it becomes. One of the responsibilities of administering a large system is seeing to it that appropriate interfaces exist for all your users. Once you understand the belief system implied in community networking, you'll do this not because it's politically correct, but because it's just plain correct.

### **Networking in the Hood: Is Inner-City Networking Appropriate?**

Most people would say that it's more than a little pretentious to suggest that inner-city neighborhoods are an appropriate place for computer networking. After all, the National Information Infrastructure is known in some circles as the country club network. Perhaps, though, such an embattled environment is one of the most important places to put access to a community networking system.

There is unlikely to be a larger group of systematically disenfranchised persons in the U.S. than those in

the hearts of its inner cities. The vacuum of hope that such situations present could potentially be affected by generating pockets of enthusiasm for activities that would keep kids in schools or neighborhood centers longer; activities that would permit single parents to receive training for new jobs without leaving their home, and enable them to have as much access to their representatives as their more advantaged counterparts; activities that would make the world beyond the neighborhood more visible, accessible and attainable. It's a hell of a row to hoe, but it's possible with an appropriately implemented community network—one that reflects and reinforces the good network that already exists in the neighborhood, and begins to displace the dysfunctional one.

## **Accomodations for Differently-Abled Persons**

Any community would be incomplete if all its members had 20/20 vision, comparable hearing, equal physical abilities, and the same learning abilities. To accurately reflect the host community or civic networking system, the system should provide access to all members of the community. In addition to the users already discussed, a few classes of users would benefit from specific accommodations.

Blind and visually disabled users could make better use of the system if an interface were available that did little or no full-screen updating so that it could be better used with a screen-driven speech device. It would be useful to think of a way to offer a class of terminal emulation such as "dumb tty" with whatever text-mode interface is offered, so that persons who use supplemental equipment with their displays can get the most out of their login sessions.

Consideration may also be given to an interactive voice-response interface to the system. Although such a system would probably offer read-only access, it would be beneficial to offer this means of access to persons who are unable to use a computer or terminal either because of economic reasons or because of difficulty with English literacy. As the price drops and ease of management rises, it won't be long before interactive voice response systems are as common as e-mail is today. It's not uncommon today for a multiline IVR system to fit into a single PC-class machine, and be no more difficult to run than a PC with about five or six application packages loaded on its hard drive.

Deaf and hearing-impaired users are not likely to encounter much difficulty using a community computing system because of the inherently visual nature of the medium. In fact, many deaf users find that the online world enables them to have interactions with hearing users, pleasantly unencumbered by common attitudes toward the deaf—frequently because it's not apparent that the person on the other side of the CRT *is* deaf.

Learning-disabled users would be likely to benefit either from an IVR system (as discussed earlier), or from a system in which any given part is available at more than one location in the hierarchy. For numerous reasons in addition to accommodating learning-disabled users, it's a sound idea to make the structure simple and feasible.

Users that have physical challenges necessitating a change in how they use a keyboard (or whether they do at all) would benefit from interfaces with a minimum of keystrokes. Reconciling this goal with the goal of minimum screen updates for visually-impaired users probably amounts to giving users an opportunity to select from two or more terminal types before the main interface is started.

The desirability of numerous terminal types for a single piece of software plays up to one of the great strengths of UNIX. UNIX uses a terminal-capability database with a standardized description language (a

set of macros, actually) to divorce any UNIX-based text application from the requirement to deal with the nitty-gritty escape-code escapades usually associated with full-screen text applications. The ease with which numerous terminal types can be accommodated is quite a strong argument in favor of using a UNIX-based solution for a community computing system.

## The "F" Word: Funding

Nothing can prohibit the existence of a community computing system faster than a lack of money. Everybody wants to know: How much does it cost? Bear in mind that any estimate is wrong unless it's based on your own research for your own requirements. Given that, some ballpark numbers are safe to throw out just as a conversation piece.

At the low end of the scale would be a Gopher or WWW server running in the background on someone's PC or Macintosh, somewhere where there was preexisting Internet access. Such a system would have negligible cost and equally negligible usability. It could probably support two or three users at the most and be able to offer consistency only when the owner of the PC or Mac wasn't using it for something else.

Further up on the ladder of cost is a medium-sized system. Such a system might be able to serve a population somewhat smaller than any large metro area. It would have at least one UNIX host to use as a server, around seventy to a hundred dialup telephone lines, and maybe 1.5 full-time equivalent employees in addition to the bevy of volunteers critical to any community system. Count on around \$120,000.00 per year.

For a larger system that would serve a community the size of a large metropolitan area, double the dialup lines, add another FTE, and start thinking in terms of around \$200,000.00.

Assuming you've done your own research and actually come up with realistic numbers, and that you've put together an advisory committee to come up with ideas on content and mission, and provided you haven't won Lotto recently, the next step is to think about funding. Three approaches to this are commercial support, grantsmanship, and self-sufficiency.

Commercial support usually comes in the form of loans or gifts of money, equipment, or services. Sometimes donations of office space or even just advice are possible. If nothing else, at least keep in periodic contact with a company that has refused all other means of support; they may one day decide to fund you after you've established yourself.

The most usable support you can probably get from a corporate sponsor is a cash gift. Corporate sponsors who give this type of support are usually doing so with the expectation that they will be able to do it in a fairly public manner, thus enhancing their public image. Equipment and service donations are also profitable to the donor in that they are frequently able to get more profit in terms of PR mileage out of last year's model of computer, modem, or terminal server than they could by selling it outright.

The fund-raising for a community computing system is never complete. One way to approach fund-raising would be to recruit enough qualified volunteers to ensure that someone is always researching, applying for, and getting grants to fund the system. It's definitely worth the while of anyone starting a community computer system to cultivate working relationships with the local grant-writing talent and convince them to sign on as volunteers.

Another option for funding is to try to make the system as self-sufficient as possible. This could involve splitting available userids on the system into several tiers. Free or low-cost userids would still be available to the general community, but commercial-use userids, from which businesses could put a shingle and do business on the main street of your virtual community, would also be available for a price that would go toward funding the system. If such users agreed to certain commonly-acknowledged ground rules about commercial practices on the Net—like refraining from unsolicited mailings—they could be a valuable part of the system, in more respects than the obvious financial one. Few communities, after all, exist without a business sector, and if the system is to truly reflect its host community, the business sector must be represented as well.

## National Organizations

All of this is definitely not new territory. Numerous national organizations exist to help with nearly every aspect of community computing, from technical advice to social consciousness. Here's a sampling of some of them.

*Americans Communicating Electronically (ACE)*. A federally-sponsored project to get all the important players in community computing (government at all layers, civic and arts organizations, job trainers, health care organizations, and so on) talking to each other. ACE seeks to enable access to such electronic resources for persons who do not own a computer or terminal and modem. For more information, send electronic mail to [info@ace.esusda.gov](mailto:info@ace.esusda.gov).

*CAUSE*. An international nonprofit organization dedicated to the management and use of technology in higher education. CAUSE provides leadership to information resource managers and strategic decision makers. The membership ranges from librarians to Information Service directors, all of whom have a vested interest in providing high-quality, well-managed network services to their users. For further information, send e-mail to [info@cause.colorado.edu](mailto:info@cause.colorado.edu).

*Center for Civic Networking (CCN)*. CCN focuses on the use of network in general, and the Internet in particular, for the broad public good. Ideas such as electronic town halls are right up CCN's alley. Their three-pronged mission is to provide public access to critical information regarding local and regional planning issues, to generally expand public access to the Internet, and to cultivate large public dialogs to assist the formation of public policy regarding such issues. For further information, point your Gopher client at the following server:

Name=The Center for Civic Networking  
Type=1  
Port=70  
Path=1/associations/civicnet  
Host=ftp.std.com

*Clearinghouse for Networked Information Discovery and Retrieval (CNIDR)*. If you've ever felt like you've lost track of all the information retrieval tools on the Internet, you'll appreciate CNIDR, which is a clearinghouse for information about such tools. Founded in 1992 with a grant from the National Science Foundation, CNIDR is a good place to start if you feel lost when confronted with the numerous information retrieval tools on the Net. For more information, try URL <gopher://is.internic.net/1/infoguide/about-internet/organizations/cnidr/>

*Commercial Internet Exchange (CIX)*. CIX speaks up for the commercial Internet user. Now that commercial users outnumber other Internet users, CIX has a hefty charge. Internet service providers wishing to join CIX agree to a set of principles regarding Internet commercial traffic and pay a fairly large annual membership fee. For more information about CIX, send e-mail to [info@cix.org](mailto:info@cix.org).

*Computer Emergency Response Team (CERT)*. Formed after the infamous Internet Worm incident, CERT is not the Internet cops, but they're the first people many sites call when they've got an Internet-related security problem. Their 24-hour emergency hotline (412-268-7090) is a kind of Internet poison control center that sysadmins can call and not only report an ongoing incident, but get expert advice on how they might address it. CERT electronically publishes CERT Advisories to notify the Net community of known security problems and fixes before (and sometimes after) they're exploited. For more information, send electronic mail to [cert@cert.org](mailto:cert@cert.org).

*Consortium for School Networking (CoSN)*. CoSN is a nonprofit organization that serves to help students and educators make better use of the Net. They're also an advocate on national networking policy issues such as school information resources, NREN access, K-12 networking, and development of new resources. For more information, send e-mail to [cosn%bitnic@cunyvm.cuny.edu](mailto:cosn%bitnic@cunyvm.cuny.edu).

*Electronic Frontier Foundation (EFF)*. If you've ever used a modem, surfed the Internet, gotten caught up in an endless cable-TV channel-check and wondered what it all meant, you're in good company—the EFF wonders too, and they've got some pretty good ideas about the impact of electronic communications on our constitutional rights. A nonprofit organization, the EFF champions the cause of new laws that protect the users of networks like the Internet, the establishment of a national network that carries an affordable variety of information, and the basic diversity of the online communities that make up the Internet and networks like it. The EFF is one of the most respectable sources of electronic civil rights information available. To get more information, send e-mail to [ask@eff.org](mailto:ask@eff.org).

*Institute for Global Communications (IGC)*. A division of the the nonprofit Tides foundation in San Francisco, IGC is a consortium of political activist networks including PeaceNet, EcoNet, and ConflictNet. By subscribing to a member network, you automatically get access to resources of the other member networks. For more information, send e-mail to [support@igc.org](mailto:support@igc.org).

*The Internet Society (ISOC)*. If it isn't already apparent to you that no one runs the Internet, consider yourself so informed now. (Actually, it's more accurate to say everybody runs the Internet.) The Internet Society doesn't run the Internet, but they have taken up the task of being Internet boosters and engaging in general promotion and education regarding the Internet. ISOC provides forums for discussion of topics ranging from the direction of the Internet and possible new applications to various ways to bring new local and regional networks "into the fold." For more information, check out the URL <gopher://ietf.cnri.reston.va.us:70/11/isoc.and.ietf>.

*The Internet Engineering Task Force (IETF)*. Someone's got to think up all this cool stuff. The IETF is the body that creates Internet standards and technologies. The IETF is made up of volunteers from all over the world, who periodically meet at various great tourist spots and try to come up with ways to outdo what they did at the last meeting—and usually succeeding. Many of the events at each IETF meeting are now broadcast in live audio and video over the Internet's Multimedia backbone—the *MBone*. For more information, check out the IETF Gopher at <gopher://ietf.cnri.reston.va.us:70/11/isoc.and.ietf/ietf>.

*The National Public Telecomputing Network (NPTN)*. All Freenets in the world are affiliated with the NPTN, the organization started by Case Western Reserve University's Dr. Tom Grundner in the mid-eighties; NPTN coordinated the growing number of Freenet community-based computing systems that were springing up, prompted by the success of Grundner's Freenet 1 system in Cleveland, Ohio. Today, the NPTN provides guidance, information "cyber-casting" feeds and structure to the 117 Freenet organizing committees in various parts of the world, 37 of which now have systems online. For more information, contact [info@nptn.org](mailto:info@nptn.org).

## Three Examples

Here are three brief examples of community computing systems.

The Rio-Grande Freenet is an example of a system that uses CWRU's Freeport software, presents its system using a standard VT100-compatible user interface, and makes itself available both through dialup access with modems up to 14.400Kbps or directly through telnet access on the Internet.

### [Figure 14.7. The Rio-Grande Freenet was the first operational Freenet in Texas.](#)

Located in El Paso, Texas, the RGFN is operated completely on a volunteer basis. It first came online in 1993, and offers registered userids to anyone in Texas, New Mexico, and Juarez, Mexico, who cares to go through the online registration process.

Those registered users enjoy relatively unfettered access to the Internet: electronic mail, numerous local discussion groups as well as Usenet, Listserv, Gopher, FTP, and Internet Relay Chat access.

The Delaware Valley's LibertyNet, located in Philadelphia, PA, offers a variety of very creatively packaged community information through graphical user interface World Wide Web browsers like NCSA Mosaic (see Figure 14.8), or through a text-mode WWW like the University of Kansas' Lynx (refer to Figure 14.6). Users who dial LibertyNet through modems or telnet to it through the Internet may also browse the system through Lynx, or use electronic mail programs such as Elm and Pine from a naked UNIX shell prompt.

### [Figure 14.8. NCSA Mosaic is just one of the ways that users may access The LibertyNet Electronic City.](#)

Users may browser through a database of national legislation as well as other topic areas including medicine, a community center, business and economic development, as well as gateways to Internet resources throughout the world.

In August of 1993, teachers from twelve schools in the Houston Independant School District set out to convince the folks in the state capital that they could teach Texas History better using a combination of textbooks, multimedia materials, and the Internet than through conventional means. A result of that effort is Armadillo (see Figure 14.9).

### [Figure 14.9. Armadillo, the Texas Studies System, provides materials for Texas middle schools.](#)

Armadillo, the Texas Studies System, is reachable through both Gopher and World Wide Web interfaces. Armadillo is a good example of resources being leveraged from an existing Internet presence to provide a community service. The computer and network necessary for Armadillo to exist actually belong to Rice University in Houston, Texas, which has collaborated with students and teachers in the Houston Independant School District. Their goal is to provide an individualized Internet presence containing information ranging from factoids about armadillos to a variety of "Super Projects" developed by K-12 students in Texas and put on the Internet for international access.

Systems that find it difficult to get startup funding may want to consider what HISD has done with Armadillo—using the resources that are already in place to leverage a community presence on the Net.

### **For More Information. . .**

There is a great deal of information available on the Net with regard to community computing in general and Freenets in particular. A Veronica search with the keyword sequence freenet -m or community is likely to be quite productive. Here are a few URLs to start you off, though.

### **Gateway to Freenets and Community Computer Networks via World Wide Web**

URL: <http://freenet.victoria.bc.ca/freenets.html>

An active list of community computing systems in Canada, the U.S., and Russia, with a short list of references at the end.

### **The Freenets Home Page**

URL: <http://herald.usask.ca/~scottp/free.html>

A well-structured list of Freenet resources by Peter Scott of The University of Saskatchewan Libraries.

### **Community Network Guide**

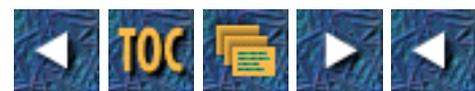
URL: <http://http2.sils.umich.edu/ILS/community.html>

Another well-structured list of community networking in general.

### **Back Issues of *Networks and Community* Newsletters**

URL: `gopher://gopher.nlc-bnc.ca/11gopher$root_pub%3a%5bnetcomm%5d`

An archive of *Networks and Community* issues back to issue #1, dated December 6, 1993.





- [Part IV](#)
  - [Finding Stuff: You Name It, a Guru Can Find It \(and So Can You\)](#)

## Part IV

# Finding Stuff: You Name It, a Guru Can Find It (and So Can You)





- [15](#)
  - [Life with Archie](#)
    - [--by Kevin M. Savetz](#)
    - [The Birth of Archie](#)
    - [Using Archie](#)
    - [Archie's Commands](#)
      - [Searching](#)
      - [Sorting](#)
      - [Limiting Your Search](#)
      - [What is whatis?](#)
    - [Telnet Tidbits](#)
    - [Using Archie via Electronic Mail](#)
    - [Using Archie with a Local Client](#)
    - [Setting Up Your Own Archie Server and Getting Your FTP Site Listed](#)
    - [Why Archie Isn't Perfect](#)
    - [What's Next?](#)
      - [For More Information](#)

# 15

## Life with Archie

### [--by Kevin M. Savetz](#)

There probably aren't many long-time Internet users who haven't heard of—or more likely used—Archie, a program for searching for files at anonymous FTP sites.

Archie is an information system that works as an electronic directory service for locating information on the Internet. It is most widely used for its unique ability to quickly scan for files available via anonymous FTP. If you know the name of a file, but you aren't sure where to get it on the Net, chances are good that Archie can tell you. For instance, if you've got a hankering to play the (low-tech but highly nifty) game Nethack, but haven't the faintest idea where to find it, Archie will point you to a few (well, actually more than a thousand) FTP sites harboring that program.

Although the program was first written in mid-1990, and didn't come into popular use until 1991, Archie quickly found a niche in the hearts of Internet users. Archie was originally intended simply to track the contents of anonymous FTP sites, but is being expanded to include a variety of other on-line directories and resource listings.

At last count, Archie's database consisted of more than 2,100,000 file names (totaling some 170 gigabytes of

information!) on well over 1,200 anonymous FTP sites.

## The Birth of Archie

Peter Deutsch is one of the people behind the scenes of the Archie program. He is president of Bunyip Information Systems Inc., a startup company specializing in Internet-based information tools. Bunyip, based in Montreal, Canada, was founded in early 1992 by the creators of Archie.

Archie began in 1986 when Deutsch was systems manager for the School of Computer Science at McGill University in Montreal. His predecessor had tried to convince the university to connect to the Internet, but because of the high cost—about \$35,000 annually for a slow link to Boston—it was hard to convince the right people that the investment was worth it.

As he took over, Deutsch found himself with several temporary Net links and a load of old equipment. With a little pressure on the administration, they set up electronic mail gateways, newsfeeds, and other Internet resources. Deutsch shook the right trees and managed to pay the bills.

He calls the birth of Archie a "classic example of serendipity." One of Deutsch's associates, the "resident pack rat," was in charge of tracking useful free software on the Net. In June of 1990, he wrote some simple scripts to go out and fetch listings of anonymous FTP site holdings for him, so he wouldn't have to check the contents to each site manually. Outsiders became interested in the service, so they added a front end to the software so others could do their own software searches.

The service was a hit immediately. "Within a couple of days we were seeing as many as 20 queries a day," Deutsch said. Deutsch promised his partners that he'd "buy them lunch when it hit 30 a day." Last I checked with him, he hadn't yet to deliver on that promise, although today 40 Archie servers (about 20 of them are open to the public) service around 200,000 queries every day. (Sounds like Deutsch owes his buddies lunch in the South of France!)

Deutsch and his partners formed Bunyip Information Systems, Inc. in January, 1992, "to support new versions of Archie and to and allow us to pursue our own particular vision of how the new crop of Internet services should all be put together." The first commercially-supported version of Archie—a complete rewrite of the software—was made available in October of 1992.



**Note:** What's a Bunyip, anyway? "A Bunyip is any one of several things," Deutsch said. "Before we came along it was, among others, a creature from Australian Aboriginal mythology, a character from a popular Australian children's story, a four-person skydiving maneuver and—I think—a kid's show in somewhere like Philadelphia. We chose the name because I am in fact half Australian and as a kid I read the book I mentioned. The name kind of stuck in my head."

---

## Using Archie

There are plenty of ways you can access Archie—all you need is an Archie client, the front-end to the database itself. If you have a shell account, your service provider may have installed Archie on your system. If so, that will likely give you quickest access to the system. (Type `archie` to find out if it's there.)

If it isn't, don't fret. You can also access Archie by telneting to a public Archie server. There are 20 or so public servers scattered around the globe. Or, if you prefer (or if you don't have access to telnet), you can send your Archie search requests via electronic mail. Both methods are quick and painless.

---



**Note:** There are other ways of accessing Archie as well. For instance, there's a wonderful shareware program called Anarchie—that's pronounced "anarchy"—for Macintosh users with IP connections. Anarchie integrates the functions of Archie and FTP. You can search for a file name, and it will quickly list matches on-screen. Just point-and-click on one of the files, and it is transferred right to your computer in one fell swoop. Or, if you use the X operating system, you can use the Xarchie program. But back to traditional Archie clients. . .

---

We'll look at the three main methods of accessing Archie separately. Although they are similar on the surface, each method has its quirks, strengths, and weaknesses.

Internally, Archie works the same way no matter what client you use. As you might expect, when you start an Archie search, the program doesn't try to log into 1,000 anonymous FTP sites at once to look for your file. Instead, Archie does its magic by searching from a pre-compiled list of files (a database that weighs in at about 250 megabytes). About once a month, the Archie server trolls the Internet's FTP sites, searching for new file listings. (If you must know, the program that actually does that is called the "Archie Data Gathering Component." Yikes.) So when you do your search, the hardest work has already been done. All Archie needs to do is poke around its list and show you what it finds.

## Archie's Commands

No matter which interface you use, Archie uses one basic set of commands. The telnet and e-mail interfaces each have a few extra commands, which we'll look at separately.

Find is the Archie command that does Archie's real work—searching the database for the name you're looking for. It works simply enough: Just type find pattern and you're in business; Archie will search its database and show you the files that match your query. (There's also a prog command, which works identically to find.)

Archie's true power to search for files is brought out with careful use of its flexible searching, sorting, and limiting functions.

### Searching

Archie includes a command for changing how it matches filenames when searching its database. The command, appropriately enough, is called set search.

Set search sub, the default searching mode, performs a case-insensitive substring search. A match occurs if any part of filename contains the item in your find command, without regard to case. For instance, searching for orange would hit orange-juice and MR.ORANGE.

The set search subcase command works like set search sub but is case-sensitive. Searching for the pattern TeX will match LaTeX but not latex or Texas.

Set search sub regex allows your search to take the forms of UNIX-style *regular expressions*. Regular expressions (similar to *wild cards* in the DOS world) give you more control over what names Archie will match. In this mode, you can do *fuzzy* matches or specify that your search word must be at the beginning or the end of the filename. For instance, searching for `^ail$` will hit on bail and tail, but not dogtail or tailwag. (I'm not going to give complete instructions on how to compose regular expressions here—that could be a chapter of its own.)

The set search exact mode will only hit on files that exactly match your search word, no questions asked. It's case-sensitive, so searching for Waffle.zip will hit on Waffle.zip but nothing else. Incidentally, the set search exact mode is the fastest search mode.

There are three more search modes: exact regex, exact sub, and exact subcase. These three search types cause Archie

to try an exact match first, then fall back to a regex, sub, or subcase search.

## Sorting

You can ask Archie to sort its hits before it shows them to you. If you choose a sorting method, Archie can (once it has searched its database) present its information in the manner that is most useful to you. Using the `set sortby` command, you can pick the method Archie uses to sort its output.

`Set sortby none` doesn't sort the hits—Archie will show them to you in whichever order it found them. Oddly, `set sortby none` is the default, although it is clearly the least useful way for Archie to sort. Not doing a sort is, however, faster than doing one.

`Set sortby filename` sorts files alphabetically by name. If you should want them sorted backwards alphabetically (but why?), you can `set sortby rfilename`.

`Set sortby hostname` sorts by the FTP site's host name, in alphabetical order. Again, to sort them backwards, do `set sortby rhostname`.

`Set sortby size` sorts the files by size—largest first. `Set sortby rsize` (you saw this coming, right?) sorts them from smallest to largest.

`Set sortby time` sorts the files by their modification date, with the most recently changed files first. On the other hand, you can see the oldest matching files at the top of the list with `set sortby rtime` command.

## Limiting Your Search

Normally, Archie searches all of those sites when you do a search. After being inundated with information overload by a few 100-hit searches, you'll probably find that you want to limit Archie's search scope a bit.

One way to do this is by limiting the number of hits you'll be shown. The `set maxhits` command forces Archie to only show you up to a specified number of matches. By default, Archie will show you the first 100 matches—if you want to see more, you can type a higher number, up to `set maxhits 1000`. On the other hand, you can speed the search by setting `maxhits` to as low as 1.

Another command, `set maxhitspm`, limits the number of files with the exact same name. If you type `set maxhitspm 10` followed by `find nethack`, Archie will only show you the first 10 hits of each filename containing the word `nethack`. This is a good way to shorten searches and weed out unwanted version numbers. A related command, `set maxmatch`, limits the number of distinct filenames returned. For example, imagine `maxmatch` is set to 2 and you search for `grape`—files in the database are `grapefruit`, `grape-nuts`, and `grapejuice`. Only `grapefruit` and `grape-nuts` are displayed.

The list produces a list of sites whose contents are contained in the Archie database. With no argument, all the sites are listed (watch out, this is a very long list). If you follow the `list` keyword with a regular expression pattern, Archie lists only sites that match your search. For instance, typing `list .fi$` lists all the Finnish sites in the database.

So, what if you're looking for a program (like `nethack`) that you're sure is on the Internet hundreds or thousands of times, but you just want to find it on a nearby FTP site; you don't care if it is available on a server half the world away. Enter the `set match domain` command, probably one of the most useful, but underutilized, Archie tools. It enables you to restrict the scope of your search based upon the domain names of the anonymous FTP sites being searched. For example, you can type `set match domain wustl.edu` to limit your searches to that single domain. Or, you can `set match domain ca` to search only sites in California. You can ask Archie to search multiple domain names by putting colons between them, for instance: `set match domain ca:apple.com:harvard.edu` will search all of California, plus Harvard and any anonymous FTP sites at Apple computer.

Archie includes the concept of "pseudo-domains" which are used as a shorthand for specifying domain names in

certain geographic areas. (ca in the preceding example isn't a true fully qualified domain name, but Archie treats it as one.) Here's a partial list of Archie's pseudo-domains. Type domains for a list of the ones in your Archie server.

africa	Africa
anzac	OZ & New Zealand
asia	Asia
ca	California
centralamerica	Central America
easteurope	Eastern Europe
mideast	Middle East
northamerica	North America
scandinavia	Scandinavia
southamerica	South America
usa	United States
westeurope	Western Europe
world	The World

What if you're looking for a popular program that works on many computer systems? If you want, for example, a uudecoder for your DOS machine, but don't want to know where to know about uudecoders for Macintosh, UNIX, Vax, and a dozen other machines, you can try the set match path command. It enables you to limit Archie's hits to files that live in a directory that contain the specified phrase. Why is this useful? Many anonymous FTP site administrators put software packages for DOS PCs in a directory path containing the name pc. Armed with that knowledge, you can set match path pc before you type find uudecode. Archie then shows you /pub/systems/pc/uudecode.c but not /info-mac/util/uudecode.sit.hqx or /pub/unix/uudecode.

You can tell Archie to allow more than one path matches through the gates by separating the parameters with colons. For instance, set match path pc:dos:windows will show you files in directories containing the words pc, dos, or windows.

## What is whatis?

Wouldn't it be great if Archie could search for programs by what they do, not just by their filenames? I'd like to be able to say, "Hey, Archie, make me a list of text editors for the Macintosh." Well, in a limited scope, Archie can do this.

In addition to offering access to anonymous FTP listings, Archie permits access to the whatis description database. The whatis database, in addition to filenames and directories, contains brief descriptions of files, which can be searched. Although it's a wonderful concept, the database contains descriptions of only about 3,500 software packages and documents, a far cry from the 2,100,000 files available via anonymous FTP.

Additional whatis databases are scheduled to be added in the coming months. Planned offerings include listings for

online library catalogue programs, electronic mailing lists, Frequently Asked Questions lists, and archive sites for the most popular Usenet newsgroups. (Then again, these have been in the "planning stage" for two years or so and seem no closer than they did back then.)

You can try it for yourself: type `whatis` followed by a word. Try `whatis RFC` or `whatis UUCP` to see how it works. Here's an example—the descriptions certainly make the purpose of the files much clearer.

```
archie> whatis rfc
```

```
RFC 1          Crocker, S.D.  Host software 1969 April 7; 7 p.
RFC 10         Crocker, S.D.  Documentation conventions. 1969 July
                29; 3 p. (Obsoletes RFC 3; Obsoleted by RFC 16)
RFC 100        Karp, P.M.  Categorization and guide to NWG/RFCs 1971
                February 26; 43 p.
RFC 1000       Reynolds, J.K.; Postel, J.B. Request For Comments
                reference guide. 1987 August; 149 p. (Obsoletes RFC
                999)
RFC 1001       Defense Advanced Research Projects Agency, Internet
                Activities Board, End-to-End Services Task Force,
                NetBIOS Working Group. Protocol standard for a
                NetBIOS service on a TCP/UDP transport: Concepts and
                methods. 1987 March; 68 p.
RFC 1002       Defense Advanced Research Projects Agency, Internet
                Activities Board, End-to-End Services Task Force,
                NetBIOS Working Group. Protocol standard for a
                NetBIOS service on a TCP/UDP transport: Detailed
                specifications. 1987 March; 85 p.
RFC 1003       Katz, A.R.  Issues in defining an equations
                representation standard. 1987 March; 7 p.
RFC 1004       Mills, D.L.  Distributed-protocol authentication
                scheme. 1987 April; 8 p.
```

## Telnet Tidbits

As noted earlier, you can telnet to a public Archie client to access the database. Select your nearest Archie client from the list (see the sidebar) and telnet to that site. Login as archie and, if you are asked for a password, type archie again. Once you're connected, you can do a basic search by typing find followed by a program name.



**Archie Servers You Can Telnet To:** Here's a list of public servers and where they're located. Please use the server closest to you. Give Archie the servers command for a current list of all public Archie servers worldwide.

archie.unl.edu 129.93.1.14 USA (NE)  
archie.internic.net 198.48.45.10 USA (NJ)  
archie.rutgers.edu 128.6.18.15 USA (NJ)  
archie.ans.net 147.225.1.10 USA (NY)  
archie.sura.net 128.167.254.179 USA (MD)  
archie.au 139.130.4.6 Australia  
archie.univie.ac.at 131.130.1.23 Austria  
archie.cs.mcgill.ca 132.206.51.250 Canada  
archie.uqam.ca 132.208.250.10 Canada  
archie.funet.fi 128.214.6.100 Finland  
archie.th-darmstadt.de 130.83.22.60 Germany  
archie.ac.il 132.65.6.15 Israel  
archie.unipi.it 131.114.21.10 Italy  
archie.wide.ad.jp 133.4.3.6 Japan  
archie.kr 128.134.1.1 Korea  
archie.sogang.ac.kr 163.239.1.11 Korea  
archie.rediris.es 130.206.1.2 Spain  
archie.luth.se 130.240.18.4 Sweden  
archie.switch.ch 130.59.1.40 Switzerland  
archie.ncu.edu.tw 140.115.19.24 Taiwan  
archie.doc.ic.ac.uk 146.169.11.3 United Kingdom

By default, Archie prints your search results on your screen as quickly as it can. On searches with many hits, this means it will "spam" your terminal with screen after screen of information. I don't know about you, but I can't read that quickly. Use the set pager command to tell Archie to give you search results one screenful at a time.

For the pager to work properly, Archie needs to know what kind of terminal you're using. Archie assumes you are using a dumb terminal unless you tell it otherwise. You can use the set term command:

```
set term terminal-type rows columns
```

If you are using a vt100 terminal, you can type set term vt100 and, henceforth, the pager will work properly. The rows and columns fields are optional. If you're working on an oversized screen, you might type set term vt100 50 132 to tell Archie your terminal's dimensions.

Even though you're using the interactive telnet interface, you may find that you would like a command's output of a command to be e-mailed to you. Using the mail command, you can do just that. If, for instance, you do a particularly useful search that you'd like to save for later, type mail followed by your e-mail address. Lo and behold, the output of your last successful command is sent to you.

## Using Archie via Electronic Mail

You can also send your query by electronic mail to an Archie database, which will dutifully mail the results back to you. Although this method can be used by just about everyone on the Net, electronic mail searches seem to take a lot longer than interactive ones.

To use Archie via electronic mail, send a message to `archie@archie.cs.mcgill.ca` with your search request in the message body. In fact, you should be able to send e-mail to any of the Archie servers listed in the sidebar. For instance, if you're just down the road from Rutgers University, there's no point in querying a server in Canada. Instead, send e-mail to `archie@archie.rutgers.edu`.

Thankfully, Archie's e-mail interface uses the same basic set of commands as the telnet interface (except e-mail users don't need to worry about the `set pager` and `set term` commands). There only a few notable additions.

path *address* tells Archie to where it should send its e-mail reply. Although it should be able to tell by looking at the return address on your e-mail, some funky mailers don't report a valid return address. Hence, you might enter `path dopey@dwarves.com` to force Archie's reply to go to that address. (By the way, the `path` command functions identically to the `set mailto` command.)

The `set max split` command tells Archie the maximum size, in bytes, of a file to be mailed to you. Any output larger than this will be split into pieces. You can set `max split` to any size from 1KB to 2 gigabytes. The default is about 50KB.

Another way to limit your incoming e-mail's size is to have Archie compress your search results before sending them. Use the command `set compress compress` followed by `set encode uuencode` to force Archie to compress (using the UNIX `compress` routine) and `uuencode mail` before sending it to you. By default, Archie sends e-mail uncompressed and not encoded.

The `quit` command tells Archie to ignore all further lines of your e-mail message. Although this isn't incredibly useful, you can use the `quit` command to prevent Archie from trying to interpret your 1000-line `.sig` file as a command.

## Using Archie with a Local Client

If you typed `archie` at your command-line prompt and were delighted to see a program rather than command not found, you can use the Archie client that's already installed on your system.

The most basic use of Archie from the command line is:

```
archie nethack
```

which will list about a billion occurrences of `nethack`. Archie's power comes when you use the command-line options. Type `man archie` for the full list of Archie commands available on your system. The command-line options let you search and sort in the same ways as using the telnet and electronic mail interfaces, although the command structure is different. To complicate matters, there are several different command-line archie clients. Here are the most important options on the Archie program I use:

```
-case           Case sensitive search
-nocase        Case insensitive search
-exact         Exact match
-reg           Regular expression match
```

-match #               Maximum number of hits

-sort date             Sort by date

-sort host             Sort by host

-reverse               Reverse sorting order

Typing `archie -exact nethack` is the same as doing `set sortby exact` followed by `find nethack` with the telnet or e-mail interfaces.

Also, if you're not doing a sort along with your search, make use of the `-along` option, which causes Archie to print hits as soon as they are available rather than waiting until it finishes searching.

## Setting Up Your Own Archie Server and Getting Your FTP Site Listed

The early Archie software was free for service providers to obtain and use. Today, Archie is commercial software: every site that offers the Archie service pays annual license and support fees. Service providers may then make their Archie server private or open to the public. For pricing information, send e-mail to [info@bunyip.com](mailto:info@bunyip.com).

If you run an anonymous FTP site and would like it to be included in the Archie database, send e-mail to [archie-admin@bunyip.com](mailto:archie-admin@bunyip.com).

## Why Archie Isn't Perfect

Like most things in life, Archie is not perfect. Although it does an admirable job of creating an enormous database of Archie information and searching that database, the program really doesn't have much to go on. For each file holed away in an FTP site, Archie knows the filename, modification date, directory path, and FTP site name.

What doesn't Archie know? It doesn't know about software version numbers. If you want Archie to only tell you where to find the latest, greatest, feature-laden version of MacPuke (or whatever), you're out of luck. Archie will happily list for you *all* occurrences of that fine program, though, if you don't mind sifting through them yourself. Alas, Archie doesn't know how to ignore old versions of software.

Sometimes, sifting through the filenames yourself to find the most recent version isn't so bad: some anonymous FTP sites give file names like `macpuke2-6.sit.hqx`, from which you can safely infer that they've got version 2.6.

Another way to tell how old a file is is by looking at the date. Here's the good news: Archie's database contains the modification time for each listed file. Here's the bad news: that date isn't when the file was actually created or compiled; it's when it was placed in that particular anonymous FTP directory. You can use the dates to tell how long a particular file has been rotting on someone's hard disk (making it pretty easy to avoid obsolete software). But just because it was placed there recently doesn't mean the file is new: the software could be as old as the hills, but recently placed in or moved to that directory.

To make matters worse, Archie doesn't have a function for excluding files older than a certain age. Wouldn't it be nice if you could look for `nethack` and limit Archie's search to files with creation dates less than 120 days ago? Well, you can't, although you can ask Archie to show the most recently modified files first.

Finally, although Archie offers a variety of powerful searching methods, it is still pretty unforgiving. If you search for a program called `peace-and-love` but the file is stored in all the archives as just `p&l`, you won't find it.

## What's Next?

So, is there life after Archie? Peter Deutsch thinks that this is just the beginning. "People are starving for information," he said. "Once we have more valuable information—not just better formatted filenames in Archie, I mean quality reference works and so on—you're going to find people will be willing to pay something for tools that help them tame Cyberspace. This in turn will bring in more information, which will feed the cycle. . .and we're off to the races. It's breathtaking to watch it happen right before our eyes. It's like the PC revolution all over again, only this time I find myself helping to make it happen. This is without a doubt the most fun I've ever had."

Bunyip's current project, currently being tested, consists of an information publishing and distribution service based on Archie. The initial pilot will concentrate on offering a basic set of system information offerings, including the existing Archie, an index of Gopherspace, a Yellow Pages service (for finding services throughout the Internet), an index of mailing lists, and "a whole bunch more," Deutsch said. The software, which insiders tell me is called Artemis, should be released by the end of 1994.

## For More Information

If you have any questions about Archie, you can send e-mail to the developers at [info@bunyip.com](mailto:info@bunyip.com). Send comments, suggestions, and bug reports to [archie-l@cs.mcgill.ca](mailto:archie-l@cs.mcgill.ca).

There's also a mailing list for discussion of the Archie software. To subscribe, send a request to [archie-people-request@bunyip.com](mailto:archie-people-request@bunyip.com).





- [16](#)
  - [Veronica](#)
    - [by Billy Barron](#)
    - [Fred Barrie and Steven Foster](#)
      - [--by Tod Foley](#)
    - [Access Tricks](#)
    - [Different or Invalid Results](#)
    - [The Stoplist](#)
    - [Relevance Ranking](#)
    - [Word Stemming](#)
    - [Multiple Word Searching](#)
    - [Boolean Searching](#)
    - [Word Delimiters](#)
    - [Using Synonyms](#)
    - [Seeing More Responses](#)
    - [Searching on Types](#)
    - [Link Files](#)
    - [Helping Veronica](#)
    - [Competing Systems](#)
    - [Future Directions](#)
    - [Summary](#)

# 16

## Veronica

by Billy Barron

## Fred Barrie and Steven Foster

--by Tod Foley

Fred Barrie and Steven Foster are the creators of Veronica, a powerful Gopher-based program that continually gathers and maintains a list of menu items on all Gopher servers worldwide—enabling you to search Gopherspace by *topic* rather than *location*. By finding your keywords and providing you with a simple menu-driven interface to navigate Gopherspace by, Veronica certainly lives up to her name: *Very Easy, Rodent-Oriented, Netwide Index of Computerized Archives*. She was the main topic of discussion recently on IRC channel *#veronica*, where Barrie and Foster met with me to talk about their work.

<tod> Tell me something about the origins of Veronica.

<barrie> Well, in the beginning we started by compiling the code for Gopher.

<foster> Steve and I were trying every little program and if it did not compile the first time, we gave up on it.

<barrie> Well, to Paul Linders credit, it compiled the first time and we were in the Gopher world.

<tod> How long have you worked together?

<foster> We started together in the fall of '91. We had both worked for NV computer services for a while before, but not together.

<foster> It got exciting together when we found we both thought Gopher was interesting. At that time we were the only 2 here who thought so.

<barrie> After the Gopher server was compiled at UNR, we started to think how to improve Gopher. Originally we were to index the Gopher world similar to Archie—that is, find out where an item is and then give the route to the user to find it.

<barrie> The idea was to give them the address of the Gopher server that matched their query and then the directory—they would have to go down to find that item.

<barrie> Steve came up with the great idea of actually connecting them direct, and then we hit the big time.

<tod> Ahaa! Like the idea that Archie and FTP should really be one tool....

<barrie> Yes, Archie and FTP should be one tool....

<foster> Actually, the dumb way of doing it would make some sense. At first, people were very interested in "Where is this resource?" and what institutional domain it belonged in; that is, they really felt they needed a lot of context.

<tod> The realspace was really that important to people?

<foster> Well, yes. I talked to Peterd [Peter Deutsch] shortly after our work began, and Peter did emphasize that...no, he emphasized that the "failure" of archie to directly connect is really a client issue, not a server issue. He said someone should do that client. But mainly, it was timing. The interactive

protocols weren't there until '91...late '91.

**<barrie>** Actually, Gopher is the FTP and Archie client that made that all work.

**<tod>** So what happened next for you and V?

**<barrie>** Well, we found some code that did a simple search of a Gopher site for administrative purposes, by Boone at Michigan State.

**<barrie>** Well, we took that code and modified, really took a machete to it :) and started to do an index of about 250 Gopher sites.

**<barrie>** The first index we used was on a NeXT computer, using the Digital Librarian with really really small data files to speed up the index process.

**<barrie>** After we brought the NeXT computer to its knees by the third day of the release...

**<tod>**!!!

**<barrie>**...we started to create new index schemes.

**<tod>** How many Gopher sites are there now?

**<foster>** June 1994: about 5800 Gopher sites.... It is a bit inexact because some gateways are halfway Gophers, maybe. From analysis of the last Veronica data collection, it looks like Gopher is being used more as "duct tape" than it was at first: many more gateways to FTP, Usenet, etc. At first we just had files, of course.

**<tod>** Okay—we're back in '92....

**<foster>** Okay, in 1992...

**<barrie>** Well, back in '92 we gave the guys at University of Minnesota a sneak peak at our little server.

**<barrie>** Farhad from U Minn came back and said we had to go to the Internet with this.

**<barrie>** So on November 17, we made an announcement that Veronica had started and asked people to comment on our great experiment in networking.

**<tod>** You originally saw V as just a personal tool?

**<foster>** NO NO—not just as a personal tool. But we needed the encouragement from the Minnesota team to announce it before we thought it was ready.

**<foster>** We wanted to work on it some more. They essentially said that the world needs this now, go for it guys, or some such Farhad-ism!

**<tod>** Well, Gopherspace sure the hell needed it....

**<foster>** That it did. Most interesting response.

**<tod>**"Most interesting?"

**<foster>** We posted to a few lists on the evening of Nov 17, and by the next morning our little NeXT

box had taken about 4700 search connections!!!!

<tod>!!!

<foster> We were pleased.

<barrie> Oh by the way—the code was written in Perl.

<tod> Thanx.

<barrie> Let us say that it was the first thing we had ever written in Perl.

<tod> You guys are quick learners....

<barrie> We were learning Perl and Veronica at the same time. We made a lot of improvements in the first three months.

<tod> What have some of the significant improvements been?

<barrie> At first, Veronica was a search that took quite a bit of time to run.

<barrie> On the order of an Archie search, somewhere about 20 minutes or longer per search.

<barrie> Most people were happy with this since they were used to Archie.

<tod> How did you cut that down?

<barrie> I read a book on how to index...after I tried to hack the WAIS code into Veronica.

<barrie> The WAIS code was faster but it still lacked something...about 10 minutes.

<barrie> I experimented a lot with different codes and finally got the search time down to a reasonable time—20 to 30 seconds. This was all in April 93 (right before GOPHERCON '93). We had to show off a little :)

*[Here Barrie gets offline for a short time; the interview continues.]*

<tod> Let's talk about your individual contributions—how's the work divided between you and Fred? Who does what?

<foster> It has varied at different parts of the project. Fred was definitely faster at learning Perl than I was. On the other hand, I saw some of how the architecture should work, first. The first cut was really very evenly split as to server code. I did the first writeups on the WAIS cut, Fred started it, and I continued with it. I was still trying to write the server in C, using pieces of WAIS code, when Fred redid it in Perl with his nice no-overhead algorithm. Last summer I went off and consulted on some other stuff for a while; Fred did the data harvests alone for a couple of months then.

Anyway, I took over managing the harvests last August. Have done them since. Fred wrote some new stuff into the server and harvesting code in the fall. Since Feb, I have been doing it without Fred. [He] has actually started a new project with a different firm, and I think he will talk about that when he gets back on.

I am working on some new things for Veronica; improved interfaces, mostly. Seems every month there is

a different pressing aspect to server bugs, or better control of no-index options, or something. In the next few months I want to work on interface issues.

<tod> There are plenty of people doing GUIs of one type or another.

<foster> YUP. I am definitely interested in the server end of it.

<tod> how do they fit with you, commercially/professionally? Has anyone approached you to package "Veronica in a Box?"

<foster> Interesting question. Yes, there have been some suggestions, but nothing real firm.

<tod> Is the code public domain?

<foster> No, the code is copyrighted. Shareware with no fee.

<tod> Heh!

<foster> Yes, that is the way we think it should be. I do think it is better to avoid proliferation of many indexers for the Net's sake. I am working with Peterd (for instance) to keep synchronized about options for controlling no-index flags, update frequency, and so forth. And of course we are working with the MN team to add things to Gopher servers which can simplify the indexing problem....

<foster> Tell me Tod, do you have any ideas about how to package "Veronica in a box?"

<tod> Heh. I'm thinking of a personalized Veronica. She could limit her harvests to categories determined by questions you answer.

<foster> What would that be like ( for you :^ ) ?

<tod> For me? Well, "interactive," "fiction," "games," "magick," and she could infer from that, and go do harvests for me.

<foster> Ah, yes. But not, of course, if each one actually does a harvest on-demand. Oddly, there is much more interest in many topics like that, but not so many resources on the Gophers. People do ask for lots of things that are not in the index yet. I hope I can do something with that information, to promote services to supply that need.

<tod> I guess it would really be sort of an AI-esque shell over a Veronica client....

<foster> Yes, the AI Veronica. I am actually interested in telescript and so forth.

*[At this point, Fred Barrie rejoins the conversation. Because of his quick return, he is now known as "Fast." ]*

\*\*\* Fast (barrie@shadow.scs.unr.edu) has joined channel #veronica

<fast> Hello again.

<tod> Hey, FastFred. <g>

<foster> These guys can't stand alone and do their own harvests; they need an infrastructure. At least with foreseeable channel capacity.... I certainly sound geeky here don't I ? <g>

<fast> Say, where are we?

<tod> We were just talking about AI Veronicas.

<foster> Right, we had just been thinking about how a personalized Veronica should work.

<fast> Whoa, went off the deep end didn't we?<g>

<tod><g>

<foster> I summarized past work division, and told Tod that you had been doing something else since Feb, but other than that have not talked about the future.

<tod> Even though the AI Veronica is a ways off, I was wondering what's the next step now for Veronica?

<fast> Well, in my opinion we need to bring V fully up to Gopher+, with both an index for Gopher+ fields and a neat ask block interface.

<foster> Yes, we will make V handle URLs soon, for URL-savvy clients....

<fast> I think the URLs will be very important. We've talked about a V that returns html for those clients....

<tod> co0lco0l. URLs are starting to be listed in mags & directories in print....

<foster> Yes, but more important, V needs to provide Gopher addresses in URL form when asked, because the client software can do stuff with those URLs.

<fast> I have envisioned a BIG V that could handle WWW, FTP, and Gopher.

<tod> Yaas—The other side of the tool...the supertool....

<foster> YES. That side. Big server side. I want to do it. Actually, I am thinking a lot about that.

<fast> Well, the supertool is not that far away. Veronica can do both FTP and Gopher now but we turned off the FTP.

<foster> Yes, exactly. It is not so far away. And it will need the AI to do it correctly, or something a little smarter on the client side. Need to sort and cross-link and so forth.

<fast> I think if Mosaic could do Gopher+ we would be there.

<tod> How far away, realistically, is the required AI?

<foster> I think the required AI is available. But I don't know how to really use it until we have URNs.

<fast> Yes, URNs would make the job easier.

<tod> URNs—Names?

<foster> Yup. Uniform Resource Names...

<fast> ...so we have one resource for any service.

<tod> Fff... who does \*that\* indexing job?

<foster> He hee.

<fast> Say anyone...we all could do it....

<foster> That is a good question. We are asking whether the AI will do it or librarians will do it. But I think Fred is getting at a potential for us EACH to register our domain name space of resources, as we are each providers...so we could have URNs within a year? 2?

<fast> Yeah 2...the Internet community is sometimes slow in committee.

<foster> Lots of far-out ideas here. One very interesting thing to me is a sort of filter that could impose various editorial organizations on the realm of resources—a plug-in editor module. If I like Tod's view of the Magick on the Net, I can use his view. Otherwise, use McKenna's :^}

<tod> Uh-huh. Different filters for different industries/whatever....

<foster> Yes.

<fast> I think that all of the next-generation stuff can be done if we only had more time to experiment. The difficulty is two years ago we could waste bandwidth, but now the Internet is growing so big that it is unwise to do that. So we need to get our ducks in a row before we announce the next tool.

<tod> How long do you plan on being "in charge" of Veronica? Do you see "passing her along" some day, or continually refining and developing her?

<fast> Actually, I have already passed the Veronica stuff over to Steve.

<foster> I want to keep developing her. Continual refinement does not look boring to me. Lots of interesting stuff keeps coming along.

<fast> I got a \*real\* job and am no longer employed at UNR. My efforts are being put into NorthStar for now.

<tod> What are you doing there?

<fast> NorthStar is an attempt at indexing the Web. So far so good. (Plus it will fall into Veronica in the future.)

<fast> What I did was take the existing code for Veronica and modify (note I am not a programmer but a major hacker) it so that it could understand URLs, then I went out and indexed the WWW with this piece of code. Same idea, gather the information, retain what I need for the index, index that information, and present it in proper WWW format for clients.

<tod> It seems that conservation of time is always a key issue with you—is this a life thread ;-)?

<fast> Well yeah...so many things to do and so little time.

<foster> Interesting...time conservation is a major thread for us both. We wonder if we are normal.

<fast> <g>

<foster> We are Net guys. Our view exceeds our sleep requirement ...

<fast> The major problem, however, to the great tools of the future besides time is the computing resources. It is hard to experiment on a working Veronica server.

<tod> You need a (what?) a pseudoserver for testing?

<fast> Yes, a fast workstation for testing purposes that is not being bogged down.

<tod> Um-hmm.

<fast> Something we can reboot often :)

<fast> Not that I have ever brought a computer to its knees, mind you :)

<tod> Heh. Are there any things you'd like to be sure to say to the future gurus who read this interview?

<fast> My suggestion to them is: On the Internet, go ahead and just do it. No one else will do it until you do, so if you get an idea....go for it.

<tod> Like the god Nike sez!

<fast> Oh yes, no committee work either....

<foster> I think I also would make a plug for public interest. We are all shaping the way the Net will be, and what its potentials will be...so keep thinking about how it "should" be.

On the surface, Veronica may seem very simple. You enter a search word, and it returns a menu of items. Then you browse through the items looking for what you really want to do. Veronica is surprisingly complex and has power that the Internet guru can only exploit.

The Internet guru can use Veronica more efficiently than a novice and therefore will have more time for other tasks. In this chapter, it is assumed that you can use Gopher and know the basics of using Veronica.

## Access Tricks

One of the most difficult things about Veronica often is just gaining access to it. Each Veronica server only allows a limited number of simultaneous searches (usually two to six). If these slots are busy, you will get a "connection refused" message.

An Internet novice just keeps trying repeatedly to get through manually. Statistics I generated based on the University of Nevada Reno and the University of Manchester, Veronica servers showed that this method is not good. When one server is busy, they usually are all busy. It is much better to wait and try at a different time.

From my research, I determined that the worst time of day to access a particular Veronica server is between 10 a.m. to 6 p.m. server time (see Table 16.1).

On the other hand, the best times appear to be between 9 p.m. and 12 a.m., and then from 1 a.m. to 9 a.m.. For some odd reason, between 12 a.m. and 1 a.m. is a busy time.

### Table 16.1. Veronica usage based on time of day.

<i>Hour</i>	<i>Percentage of Queries</i>
12 a.m.	4.19
1 a.m.	2.26
2 a.m.	1.96
3 a.m.	1.97
4 a.m.	2.02
5 a.m.	2.25
6 a.m.	2.53
7 a.m.	2.97
8 a.m.	3.63
9 a.m.	4.26
10 a.m.	5.15
11 a.m.	5.74
12 p.m.	6.29
1 p.m.	6.52
2 p.m.	6.33
3 p.m.	6.32
4 p.m.	6.06
5 p.m.	5.58
6 p.m.	4.98
7 p.m.	4.76
8 p.m.	4.34
9 p.m.	3.77
10 p.m.	3.29
11 p.m.	2.81

The intermediate Internet user would probably come up with the idea of a war-dialer program that would keep trying until it got through. However, the real Internet guru would realize that all this does is slow down Veronica service for all other people. Therefore, the Internet guru would not use this method.

A real Internet guru with quite a bit of spare disk space (1.3GB for the April 1994 dataset) can run his/her own Veronica server to guarantee access. If the security tool TCP Wrapper is installed also, it is easy to control who can access the Veronica server. More advanced Veronica server administrators can even set up two Veronicas: One for public Internet access with a limited number of simultaneous connections, and one for local access with a high number of simultaneous connections.

Related to this discussion is a program called Maltshop. Some Veronica servers have a bad habit of

crashing frequently. Maltshop is installed by a Gopher server administrator, and it automatically tests the servers to see if they are up. Unfortunately, some server administrators have set up their Maltshops to check too frequently, which has put some strain on the Gopher servers out there. In fact, the author of Maltshop has called it a virus.

Hopefully, one of the improvements mentioned at the end of the chapter will largely eliminate the need for Maltshop. If you set up Maltshop in the meantime, please set it to check only about once or twice a day.

## Different or Invalid Results

Though, in theory, all Veronica servers should give you the same results, they do not. Some Veronicas are running an old version of the database. Others may remove some parts of the data that are of little value to conserve disk space. Some Veronica servers may even be running an old version of the Veronica code that may have some bugs.

Even when Veronica works normally, you probably will notice that many of the items returned do not work. Several events cause this to occur.

First, sometimes Veronica ends up indexing a collection of files that expire quickly. If you run such a collection, check the section "Helping Veronica," later in this chapter. Second, Gopher administrators love to move items around their Gopher servers to make their Gopher servers look better. This is okay because Veronica will find the items the next time it indexes. Other Gopher administrators will restrict items from public view, making them inaccessible. Finally, sometimes when you are looking for an item, the Gopher server that it's on is just down.

## The Stoplist

Most indexing systems, including Veronica as well as WAIS, contain a list of words that are not indexed. This list is known as a *stoplist*. The words in the stoplist are commonly occurring words that are not useful in almost all searches. For example, words like *the*, *a*, and *an* can be found in the stoplist. The advantages of the stoplist are that it makes the index much smaller and saves CPU time when a query is being processed because the word can be safely ignored if it is part of the search string.

An Internet guru should probably learn what words occur in the stoplist. The Veronica stoplist can be found in the Veronica server package, which can be found via anonymous FTP on futique.scs.unr.edu.

## Relevance Ranking

As many Internet gurus are also WAIS users, they are used to a WAIS feature known as *relevance ranking*. Relevance ranking assigns a relevance number to each document. The more times the search string appears in the document, the higher its ranking. Veronica has no similar feature, so the order Veronica returns items has no meaning.

## Word Stemming

Occasionally, you may want to search on both the single and plural forms of a word. You may also want to search on both the noun, adjective, verb, and adverb forms of the word. *Word stemming*, also known as *wild carding*, is the solution to this need. The way word stemming works is that it finds all words that start with the specified characters.

While doing a Veronica search, end the word with a \* (asterisk) to specify stemming. For example, you may want to look for book and books. In this case, you enter book\* as your search string. This will return all words starting with book followed by any letters. In addition to what you are looking for, you will find that the search returns bookkeeper too, so you need to be careful when analyzing the results.

Some systems allow you to control the number of characters that can be added to the stem. For example, book? returns book and books, but not bookkeeper. Veronica unfortunately does not support this feature.

It should also be noted that a stem by itself (for example, \*) is not a valid search string. Also, it does not make much sense to use common stems such as a\* or under\*. Thousands of items will be returned, and the result will be so big that it's meaningless.

## Multiple Word Searching

Multiple words can be used in a Veronica search. For example, if you are looking for the book Tom Sawyer, you can just specify tom sawyer as your search string (remember that Veronica is case-insensitive). First, Veronica will find all items with the word tom. Then, Veronica will find all items with the word sawyer. Finally, it will compare each list and return items that are in both lists.

It should be noted that any multiple word search is not order dependent. If a search is done on tom sawyer, you will also be returned items of the form sawyer tom if they occur in the Veronica database. Currently, there is no way to specify that you want hits only of a specific order to be returned to you.

In multiple-word searching, there is no performance advantage in reordering the terms as there is with some database systems. Searching for tom sawyer and sawyer tom will take the exact same amount of time and give you the same results.

## Boolean Searching

Frequently, Veronica returns hundreds if not thousands of entries when a query is entered. Every once in a while, that is useful. However, most of the time, it is not. Boolean searching allows you to more closely specify your query on your target. Another benefit of Boolean searching is the ability to include more words into your search and make your results larger, but contain exactly what you need.

Boolean searching uses three keywords: AND, OR, and NOT. In addition, parentheses allow words to be grouped together and processed first. The use of AND is unnecessary as any two adjacent words excluding OR and NOT will be ANDed together automatically, as we saw previously.

OR works by evaluating each of the words and then returning all of the items that match either word. For example, we are interested in finding all Gopher items about Bach or Mozart. We can do the search bach or mozart. Veronica will first find all items with the word bach in the title and then it will find all the

items with the word mozart. Then it will present all of these items to the user.

NOT finds items where the first word is in the item and the second word is not. You cannot start a query with the word not. For example, saying not baseball is an invalid query. It is not really a useful query anyway, as it would return hundreds of thousands of items.

Parentheses are useful for more complex searches to make sure that what you want is what you actually are going to get back. The easiest way to see this is to run through an example. You want information on either left or right field. To begin with, you enter left or right field. Veronica returns you over 2000 items. When you start looking through them, you realize that Veronica has returned a lot of items that have the word left but not the word field in them. The reason is that Veronica interpreted the query as left or (right field). After figuring this out, you enter (left or right) field, which returns you just six items that all meet your requirements.

You may be wondering why Veronica grouped the original query as left or (right field). This is because Veronica parses the query from right to left instead of left to right as most systems. Instead of remembering this and wasting your valuable time figuring out how to build your query so it parses correctly right to left, it is much easier and highly recommended to just use parentheses.

## Word Delimiters

Veronica has an unusual concept of what a word is that it applies when it is indexing a title. This is very important to know when you are querying Veronica. Veronica treats any special characters as a delimiter. Therefore, a newsgroup name such as alt.bbs.internet would be indexed under alt, bbs, and internet separately. When you are performing searches, it does not make any sense to use special characters either. If you want to search and find items about the newsgroup alt.bbs.internet, you can just search for alt bbs internet or alt and bbs and internet.

## Using Synonyms

At the current time, Veronica does not support a thesaurus feature. If it did, then when you searched on a word, you would also find items with other words that have the same meaning. At times, it is important to do searches of this nature to find all the items you are looking for.

The solution for now is to pull out your own thesaurus and build your own list of related words, and do a query on all of them. Let's say we are interested in the 1960s and hippies in particular. I looked up the word *hippie* and decided to enter the query hippie or hipster or beatnik or bohemian.

If you do not have a thesaurus handy, several thesauri exist on the Internet that you can use. If you do not know where they are, just query Veronica to find out. One caveat, however, is that most of the thesauri accessible by the general Internet community is the 1905 version of *Roget's Thesaurus*. Needless to say, it is not current—especially on topics like computing.

The danger of any thesaurus feature is that in English, at least each word has a single different meaning. Therefore, some of the results may not be exactly what you are looking for, but you can always ignore the items you don't want. Some of the Gopher clients even support a key to remove items from your display so as you go through your Veronica results, you can make the items you are not interested in disappear. For example, on the University of Minnesota UNIX Gopher client, you can invoke this

function by hitting D when your arrow is pointing at the item you no longer want to see.

## Seeing More Responses

Very frequently, you will enter a Veronica query, 200 items will be returned to you, and at the end you will see an item that says **\*\* There are 1958 more items matching the query 'golf'**, or some similar message. This is often a good time to do a Boolean search, as was previously discussed. However, at other times, you'll want to see all of these items.

To see more items, you can add the option `-m` to the end of your queries (for example, `golf -m`). If you specify it as `-m`, it will return all items no matter how many there are. Alternatively, you can specify a number after the `-m` to specify exactly how many items you want to see. For example, `golf -m500` returns you the first 500 items matching `golf`.

It is important to use `-m` only when you need it. Some clients have trouble with long menus being returned. Also, using `-m` adds an additional load to the Veronica servers. Most of the time when a large number of items are returned, the user will not use them, so the extra values are unneeded. The server has to do some extra processing to return all of the items to the user, so not using `-m` saves a little bit of time.

## Searching on Types

Every item in Gopherspace has an associated type (see Table 16.2). By default, Veronica returns you all items no matter what their type is. By using the `-t` option, you can specify what types you want to get back. The `-t` is followed by a one character type that you want to get. For example, if you only want to see directories (menus), you can specify `-t1`.

**Table 16.2. Popular Gopher types.**

<i>Type</i>	<i>Description</i>
0	File
1	Directory/Menu
2	CSO PhoneBook
3	Error
4	Binhex file
5	DOS binary archive file
6	Uuencoded file
7	Index, such as Veronica, WAIS, and so on
8	Telnet session
9	Binary file
T	TN3270 session
I	Image

h	HTML file
---	-----------

## Link Files

Internet gurus can use Veronica to add items to either their own Gopher server or their bookmarks file. If an `-l` option is added to the search, the first item returned is a link file that contains links to all the items that were returned in the search. This item can be saved into a file and then included into a UNIX Gopher server to build a new menu, or in the `.gopherrc` file on UNIX to add bookmarks. This is also useful when you need more than one session to go through the results of a Veronica search.

## Helping Veronica

The Veronica database is ever growing. In April 1993, Veronica only consumed 300MB of disk space. By April 1994, Veronica had grown to 1.3GB. If the current growth rate continues, Veronica will be up to 5.6GB by April 1995.

The Veronica team has been finding ways to keep the database as small as possible without losing valuable data. Unfortunately, these methods alone are not sufficient to deal with the growth of Gopherspace by themselves. If the growth continues, fewer sites will be able to run Veronica databases. The fewer sites will have to absorb more of a load, meaning they will be busier than ever.

If you are a Gopher server administrator, as are many Internet gurus, then you also can help Veronica work better. Most importantly, it is critical that items of little or no value to the Internet community at large not be included in the Veronica database. An item that falls in one of the following classes generally should not be included in Veronica:

1. Experimental items that are not ready for production use.
2. Items that expire quickly. This is especially the case for Usenet News articles.
3. Restricted directories that should be accessible only by a limited number of users and not the whole Internet.
4. Items that are not of general interest.
5. Items that are useful to the Internet community at large, but the server administrator does not want to include in the Veronica database due to the fear of your server not handling the load of the entire Internet.
6. Items that are useful to the Internet community that exist in several locations on the network, but your copy of the items is older and more out-of-date than the ones existing elsewhere. A classic example of this is the hundreds of sites that have partial collections of RFCs.

The instructions in this section are written around the University of Minnesota UNIX Gopher server, which is the most popular Gopher server in existence. It is expected that if you are reading this book, you

should be able to modify these into what is necessary for your particular server—or know how to ask on the comp.infosystems.gopher newsgroup for further assistance.

If you want to remove your whole server from the Veronica database and you are running version 2.x or higher, it is easy to do. Just add the line `veronica index: no` to your `gopherd.conf` file. For version 1 or excluding parts of your Gopher Tree, you need to ask for the current standard on comp.infosystems.gopher or by e-mailing Steve Foster at `foster@scs.cnr.edu`.

Also, it is possible to generate a file with all of the items that you want indexed by Veronica. Then Veronica will use this file instead of traversing your Gopher tree and looking at each individual item. As this feature is undergoing some changes even while I am writing this, I will not specify the details for now.

## Competing Systems

Though Veronica was the first system of its type to index Gopherspace, other systems are starting to do it also. Internet gurus should keep track of these developments because each will probably be useful in its own way. The other systems mentioned next should support the options in the previous section to allow people to make sure their Gopher server is not indexed. If you notice they do not support them, you should inform the system's author that they do not.

Most likely before this book hits the streets, Archie will also support Gopher (it is in beta test as I am writing). The major difference between Archie and Veronica is that while Veronica indexes everything unless the Gopher administrator specifically requests it not be indexed, Archie will index only those sites that request to be indexed. In April 1994, Veronica indexed around 6900 Gopher servers, but the Archie people had requests that only 600 servers be indexed.

Rhett "Jonzy" Jones, author of Jughead, proposed a scheme by which Jughead will now allow searches of large parts of Gopherspace. The plan is to have Jughead servers have knowledge of each other in a DNS-like scheme. Other sites that run Jughead will be the only ones indexed by Jughead, so the number of sites listed in Jughead will probably be smaller than either Veronica or Archie. Some people doubt that this scheme will work. However, the unofficial motto of the Internet community is that almost anything is worth a try, and we can learn as much from our mistakes as our successes.

## Future Directions

In the middle of writing this chapter, I attended the GopherCon 94 conference. During the conference, many discussions were held about the future of Veronica. I will present the features that are of interest to the Internet guru and are planned to be added to the code. However, I make no guarantees that these additions will be made. Of course, if a particular modification is important to you, you are always welcome to take the source code, modify it, and send the modification back to the Veronica Team.

As I write, an alpha-test version of Veronica is being worked on. In this version, Veronica will group the items it returns by hostname. For example, you search on the word `dog`; it will give you back a list of menus. The first menu is a list of all items with `dog` in the title. Selecting this menu gives you the standard Veronica results. The rest of the menus list host names. If you select the host name, you see all the items with the word `dog` on that hostname. The benefit of this feature is that you can get an idea of

where sites are that contain materials of interest to you.

A couple of ideas about improving Veronica performance has been talked about recently. Veronica is written in Perl and runs under INETD. What this means is that the Veronica source code is compiled every time someone does a query. Heinz Stoewe is working on a way to keep Veronica running and not have it recompile every time a query is done—this should make a minor improvement in performance. My estimate is that it will save a quarter of a second per query. In the longer term, it is hoped that Veronica will be rewritten in C. Once that happens, it is hoped that Veronica's bottleneck will be purely disk I/O.

Mark McCahill of the Gopher Team came up with an excellent idea. The idea was to do a search with a special flag and end up in a random spot in Gopherspace. This would be a good method for you to find new, unexpected resources in Gopher.

Veronica's database will be undergoing some changes. First of all, the stoplist will be increased. Gopher gateways to FTP, X.500, and other services will no longer be included in the database. Some discussions were held about removing all filenames from the database.

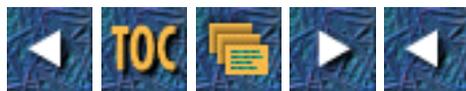
Several of us came up with a method to minimize the Veronica access problems I discussed earlier. A Gopher-to-Veronica gateway is planned. The gateway will accept a query. The gateway will contact a preferred Veronica server if the system administration has defined one. If the preferred server does not resolve the query due to load, the gateway will randomly try another server. It will keep trying servers until the query is successfully completed or too many retries occur.

In the longer term, Veronica may support synonyms. When this feature is implemented, a user who searches for dishes would also see the occurrences of plates in the Veronica database. This is a common feature in other searching systems.

Another interesting possible future addition is the addition of *facets*. Facets give access to an idea by some of its other features instead of just its title. For example, searches could be done by Geography, Discipline, or time period. These additions will allow queries to be narrowed down more easily.

## Summary

Veronica is useful to all Internet users, including the Internet guru. The rapid growth of Gopherspace is starting to strain Veronica. It is up to us—the Internet gurus—to find ways to improve Veronica and keep it the useful service it is over the next few years.





- [17](#)
  - [Discussion Forums](#)
    - [Finding Mailing Lists](#)
      - [An Overview of Mailing-List Software](#)
      - [Listserv](#)
      - [Listproc and Majordomo](#)
    - [Sources for Mailing-List Information](#)
      - [Finding Scholarly Mailing Lists](#)
      - [Finding Electronic Journals](#)
    - [new-list: A Mailing List for Mailing Lists](#)
    - [--by Tod Foley](#)
  - [Finding Electronic Mailing Lists Via a Listserv Database Search](#)
    - [Searching the Databases](#)
    - [An Example Search](#)
    - [Analyzing the Search Output](#)
    - [Getting More Information by Retrieving Archives](#)
    - [Printing Database Items](#)
    - [Some Example Database Hits](#)
  - [Some Additional Benefits of the Listserv Database Capability](#)
    - [Finding Other Databases](#)
    - [Searching a Mailing List Archive](#)
    - [Finding Listserv Installations](#)
  - [Finding the Usenet Newsgroup you Want](#)
    - [An Example nn News Reader Search](#)
    - [Finding Newsgroup Messages via Archie](#)
  - [Finding Mailing Lists and Newsgroups in Gopher](#)
    - [Finding Lists in Gopher Menus](#)
    - [Using Veronica to Search Gopherspace for Newsgroups](#)
    - [An Example Veronica Search](#)
    - [Electronic Journals via Gopher](#)

# 17

## Discussion Forums

[--by Philip Baczewski](#)

When many people think Internet, they think e-mail and discussion forums. These are the arenas that bring people together to

communicate about items of common interest. If there is power in technology, perhaps it is that power to bring people together, independent of place, age, economic status, or academic standing. The Internet guru is comfortable finding and using new and interesting forums. Because of the sheer volume of forums, this task is not as easy as it used to be. What follows here are some techniques to make the process of discovery more efficient and effective.

## Finding Mailing Lists

If you've hung around the Internet for any length of time, it is likely that you are familiar with mailing lists. Along with their cousins, the Usenet newsgroups, mailing lists are one of the longer-standing members of the community of Internet resources. Mailing lists and newsgroups now both number in the thousands. The challenge these days is not in finding mailing lists, but in finding the mailing lists that interest you.

### An Overview of Mailing-List Software

If you've hung around the Internet for any length of time, it is likely that you are familiar with mailing lists. Along with their cousins, the Usenet newsgroups, mailing lists are one of the longer-standing members of the community of Internet resources. Mailing lists and newsgroups now both number in the thousands. The challenge these days is not in finding mailing lists, but in finding the mailing lists that interest you.

### Listserv

To be effective in finding mailing lists, it is helpful to understand a little about how and where mailing lists are maintained. There are currently three major mailing list software packages being used on the Internet. Listserv, written by Eric Thomas, is a package long associated with the Bitnet research and educational network. Listserv also includes a database-search and file-distribution capability, and it supports a high degree of interoperation between installations. The number of mailing lists maintained by Listserv installations is now around 5000.

Because so many Listserv mailing lists exist, when you search for one on a particular topic, Listserv installations will be an obvious place to look. Until recently, Listserv had been used primarily on the Bitnet network, so most Listserv installations are maintained at colleges and universities. While many lists are devoted to the scholarly discussion of various topics, most lists are open to more general discussion and are not restricted to the academic community. (It's important, however, to be sensitive to the purpose of a particular mailing list and to not violate the list's or Bitnet's usage guidelines.)

Listserv was originally written for VM/CMS IBM Mainframe systems, and it used the NJE networking protocol employed by Bitnet for its intercommunication. Listserv services have been available to Internet sites via e-mail gateways. Newer versions of the software, available from LSoft International, Incorporated, support the TCP/IP mail protocol and will expand its operation to Digital Equipment's VMS operating system as well as to UNIX systems.

### Listproc and Majordomo

Two new players in the mailing list arena—Listproc, by Anastasios Kotsikonas, and Majordomo, by Brent Chapman—both run only on UNIX operating systems. Listproc is designed to have some similarity to Listserv in its user commands, but it is not capable of interoperating with Listserv installations. Majordomo is gaining popularity as a mailing list management option because it runs on many kinds of UNIX systems and is easily installed and configured. (For more detailed information about using these various mailing list packages, see Chapter 12, "Listservs and Mailing Lists.")

Being cognizant of which software package is being used to maintain a list will help in the process of locating and subscribing to a mailing list. Listserv is the software known to most people and has features that are very helpful in locating a particular list and in providing subscription information (the LIST command, and so on). Finding and subscribing to non-Listserv mailing lists might require a bit more work on your part, but such information is also not hard to find if you know where to look.

## Sources for Mailing-List Information

A number of sources have taken on the task of compiling and organizing mailing list information. These sources can be the first place to look when searching for a topical e-mail forum.

### Finding Scholarly Mailing Lists

As online information resources become more entrenched as a part of scholarly study and human intercommunication, the classification and organization of such information has become an important new endeavor. For example, Diane Kovacs, of the Kent State University Libraries, has compiled a directory of scholarly electronic mailing lists and newsgroups. The directory is actually a set of files that can be acquired from [listserv@kentvm.bitnet](mailto:listserv@kentvm.bitnet) and via anonymous FTP from [ksuvxa.kent.edu](ftp://ksuvxa.kent.edu) (in the Library directory).

The files that make up the directory are as follows:

ACADLIST README	(an explanatory note for the Directory with an index)
ACADLIST FILE1	(Anthropology - Education)
ACADLIST FILE2	(Futurology - Latin American Studies)
ACADLIST FILE3	(Library and Information Science - Music)
ACADLIST FILE4	(Political Science - Writing)
ACADLIST FILE5	(Biological sciences)
ACADLIST FILE6	(Physical sciences)
ACADLIST FILE7	(Business and general academia)
ACADLIST CHANGES	(all the major additions, deletions and alterations)

The organization of lists and newsgroups by subject enables easy access to those on one particular field of study. This is obviously a valuable service to the Internet community of scholars, but it can also point the way for others seeking information on a particular topic.

## Finding Electronic Journals

Another similar, very useful directory is one of electronic journals, compiled by Michael Strangelove at the University of Ottawa and available from [listserv@acadvm1.uottawa.ca](mailto:listserv@acadvm1.uottawa.ca). It consists of two files, EJOURNAL1 DIRECTRY and EJOURNAL2 DIRECTRY, with entries organized by category of publication (journal, newsletter, digest, and so on). Each entry has the journal title, ISSN number (if any), a description, subscription information, submission information, related electronic mailing lists (if any), back issue information, and a contact name and address.

To retrieve any of the aforementioned files from a Listserv site, you can send a mail message with the command `GET filename`, where you substitute the name of the file you want for *filename*. The file will be sent to you as a mail message.

These two examples are not the only indexed collections of mailing lists and discussion groups, but they are among the most comprehensive in their respective areas. As additional indexed collections are created, they are often announced on the mailing lists [PACS-L@uhupvm1.uh.edu](mailto:PACS-L@uhupvm1.uh.edu) and [net-happenings@is.internic.net](mailto:net-happenings@is.internic.net). These are good lists to monitor for such developments, along with many other new Internet services.

## ***new-list*: A Mailing List for Mailing Lists**

If you wish to keep abreast of all new mailing lists that are created, a good mailing list to join is [new-list@vm1.nodak.edu](mailto:new-list@vm1.nodak.edu). This Listserv mailing list is the *de facto* forum where people announce newly created e-mail groups. It seems that one or two new mailing lists are announced per day, making this a fairly active forum. In addition to announcements of new lists and changes in the status or location of existing ones, it is also possible to post an inquiry if you are looking for a list on a particular topic.

The type of posting is identified in the e-mail subject field. `NEW:`, followed by a mailing list name, means that the message body is announcing a newly created list. `CHANGE:`, followed by a mailing list name, indicates a change in status or location, and `SEARCH:`, followed by a topic, indicates a search for a mailing list on a particular topic. List members can directly reply to the person requesting the information. New-list is moderated by Marty Hoag at North Dakota State University. You can subscribe to new-list in the usual way: send a mail message to [listserv@vm1.nodak.edu](mailto:listserv@vm1.nodak.edu) with the command `SUBSCRIBE NEW-LIST yourname`. To post an inquiry, however, send a message with the subject `SEARCH:...` and a message body describing what type of list you are looking for, to the address [new-list@vm1.nodak.edu](mailto:new-list@vm1.nodak.edu).

## Gail Williams

*Photo by Matisse Enzer*

**--by Tod Foley**

"I think hosts are the essential link in where and how the Well evolves, and I have a lot of hope.

How we use the Well, and how we 'use' one another as employees, customers, allies, products, neighbors...all those words we use...is very important. Not just to the community, to the survival of the business that provides 'the platform' and the customer and technical support, but also to the way we treat each other on the planet.

The microcosm is very important.

If I didn't believe that I wouldn't work here."

—from "On Online Community," by Gail Williams

As Conferencing Manager for The WELL, Gail Ann Williams is known by thousands of users as <gail>. Reading and posting in dozens of topic categories daily, her job is to help The WELL's Conference Hosts keep their users' conversations active and "on thread." The fact that she manages to keep on top of all these chains of thought is amazing in its own right; but Gail also manages to instill a beneficent and caring undercurrent in the sometimes-rowdy WELL environment. In a recent e-mail interview, she talked to me about her job, about her views on conference hosting, and about the relationship between online forums and theatrical presentations.

**TF:** The WELL has been around longer than most Internet conferencing sites, and it has had a chance to develop some fairly clear guidelines which others have only begun to glimpse. As The WELL's Conference Manager, you come in contact with thousands of posts on a daily basis and participate in hundreds of threads—some of which must occasionally get out of hand. When it comes to "managing" these conferences on daily run-throughs, what basic guidelines do you like to keep in mind?

**GW:** The most exciting thing about the WELL conferences is that the conference hosts do most of the work that allows juicy conversation to continue. My work turns out to be much more like *host support* than *host management*—though I do select and evaluate the hosts, and accept new public conference proposals. The general autonomy of the hosts is essential to the creative spirit of the WELL.

I like to sample conferences, but since my e-mail and my policy-oriented conference rounds take a big chunk out of each day, I don't get around as much as you might imagine. The WELL has far too much posted on it each day for one person to read, so the community has to do most of the "managing." I'm often contacted by relative newcomers who've ruffled oldtimers' feathers and are getting the "white corpuscle" treatment—which may, in fact, help the health of the group, but brings up important questions about peer-pressure and dissent. So there's a balancing act in that arena, too.

**TF:** Can you give me an example of such a "balancing act?"

**GW:** Well, at times I've been able to say "Hey, puffball, hydrant just seems to want to end this without losing too much face. You're more mature, can you figure out a way to let him have a truce there?" and then mail hydrant with "Dear hydrant—your contributions here are refreshing in many ways. One thing that you may find helpful in getting into the dialog and becoming a beloved and fierce member of the local gang is to remember that puffball is not going away. You may have a shot at persuading him in the long run, but telling him to log off and die seems to be provoking him. I understand that you see his volleys as harassment; I have a hunch he feels the same about yours. Do you have any ideas of how to agree to disagree, and to settle into an articulate argument again?"

Now and then that works. I can speak directly and with respect to each party and help break a log jam. But even better are the times when conference hosts do this and I don't even hear about it. We're deep with experience around here, though no approach works for everyone all the time.

**TF:** One thing I've learned in my attempts at moderating is that it requires a new and unique social skill set. In your .plan you describe it this way: "Hosting is an evolving skill, at best an emerging art form, ranging from 'take out the dirty ascii ashtrays after the party' to much, much more." What traits and attitudes do you consider most important in a good host or moderator?

**GW:** I've thought about this a lot, and I feel that the most important characteristics have to do with passion and restraint (or

detachment—if you'd rather be Buddhist about it). You need to have passion for the subject matter, the medium, and the people who come to participate. This love can move you to keep people from trampling all over your conference or list; it can have a tough aspect, if it's part of a love for the continuing conversation in all its intrinsic messiness. And you need to be able to avoid dominating the conversation, and expecting a particular outcome. Whether by logging off and going for a walk, or just not letting it matter in that particular way.

WELL hosts take on a wide array of styles, one of which is like the stage manager for an improvisational troupe. You can say, "Psst, you're on now!" in e-mail now and then, but if you move to a directoral or critic's role, you'll be seen as the vile antagonist by your ascii stars, and get sucked into the drama of it all. It's important to remember that every player is the protagonist. This is mundane in a way; it's been said forever, and it maps to old wisdom like "do unto others.." or even "the customer is always right." Each contributor is the hero of the argument, the pleasant chatterer, the savvyest information swapper. Each user is the center of the Net. And yet, in a dispute, the infallible users may not all be right! What an adventure!

Basic conversational skills are perhaps the most important, and the regular offline party-host talents, too. How do you make people welcome and draw them in? How do you keep a fist-fight from breaking up the potluck dinner?

Ideal Host Qualities:

- Having been online, and expressing oneself for a while... a year or more is best.
- A long fuse in arguments... but not avoidance of conflict situations.
- A willingness to shift from 'center stage' to a supportive, active listening role, and back.
- A body of knowledge and broad enthusiastic interest in the subject matter.
- Willingness to learn and teach the use of conferencing software, and the local online customs, in an easy manner online as we go along.

-- <gail>

**TF:** Like the cast and crew at the theater, the people "behind-the-scenes" of the conferences do different things, and yield different rewards for their participation, than the "audience members" who come through their areas. We've talked about "hostly" qualities for keeping *users* involved, but what keeps the *hosts* involved?

**GW:** Hosts can get a lot out of running a great conference; "networking" in the professional contact sense, or in the sense of making friends who care about a mutual area of interest, perhaps professional recognition, a soap-box of their own, a chance to learn from other experts, a way to do community service. And for some, perhaps the satisfaction that can be felt by a great stage manager or director or teacher: hosts can watch others take part in a fabulous conversation that has a life of its own and still muster some pride about the space they've defined.

I believe that great hosting is an evolving art form for some, so it's exciting to bring WELL hosts together to share their definitions and advice with one another. There are aspects of magic in hosting well, and we are beginning to figure out how to teach each other the best of what we do to keep the magic alive online.

**TF:** How do you think the conferences would differ if based upon a more traditional corporate structure—in which the "hosts" are paid specialists, perhaps receiving bonuses for high participation?

**GW:** Man, that's a tough one. I am sure things would feel different if hosts were paid employees, but I am not sure what the overall effects would be. Some people perform better, or can put more in, if paid. Others are more passionate *because* they are here for love. Love in all senses of the word, from a public service stance, to the best sense of the artistic drive of the word "amateur," to falling in love with individuals and having this become the center of their social lives.

I wonder about this, and I know the hosts do, too. In many ways it would be fair and wise to pay hosts, but in other ways it could undermine the incredible outpouring of goodhearted exchange that hosts and all users bring to the table here at the WELL.

**TF:** Although your work at the WELL is a full-time occupation, you originally signed up to do community relations and marketing for the theater company you were involved with. Members of The WELL's Theater Conference have read about your experiences with various forms of experimental theater, including interactive theater. How does your experience in the world of theater affect your perception of the WELL? How is theatrical production similar to conference management?

**GW:** It's remarkable how computer-mediated looks to a theater person. I'd spent years doing theater in moderately formal settings, and in quirky freeform situations, like in-character street improv, and on the radio with call-ins. So when I logged on a few times and noticed what people were doing on the WELL, it was like a blast of electricity to the brain... It's both talking and writing... Script-improv! Improvisation at the author level, interaction before interpretation, the writer as actor and heckler!

This is exciting territory. I spent a year online asking myself questions like "Who is the audience?" "Who is the protagonist, our hero?" and "Why do we treat each other like scenery?" Most people don't like being mere props or extras, but we're all part of the painted backdrop in somebody's story. Online, that dynamic can be illuminated in a strange way... We lose the visual clues for who wants to be in the spotlight, and who wants to blend in to the crowd, muttering a few asides. So we tend to have to reinvent ourselves with some heat and friction as we get used to a new kind of reflection from our online collaborators.

**TF:** What theatrical works or experiences helped you develop this mindset?

**GW:** The writings about interface as theater, such as the work of Brenda Laurel, which I discovered after being online for some time, helped me see why this process of examining the WELL with a theater metaphor had so many layers. You could reach deep down to the ritual origins of theater. Add in my varied models for theater interaction, from my work with the political satire of the Plutonium Players, the sex-role spoofs of Ladies Against Women, and the joy of studying therapeutic psychodrama and teaching theater games... It came together in a most impressive way.

**TF:** Obviously there's a lot of extremely ephemeral stuff to learn in this area; The WELL is still blazing a trail into uncharted territory, but what words of advice would you have for people who are considering setting up their own conferencing systems?

**GW:** Well, reading, posting, and even volunteering on other systems is a start. If I were to set up a system from scratch, I'd start with artists and consultants who are not yet well-known, but who were talented. There is a strata of people who crave getting out there with great content, and who don't need to be paid to be wonderful. Finding them would be the first challenge. And I'd honor dissent. Even snarky clueless carping.

Just as a government that commits to free expression has to sort out the seemingly useless venting from the valuable input, the system founder who wants commitment and good feedback needs to figure out a way to avoid being riled by those who seem to be renting a surrogate authority figure to rail against for a couple of bucks an hour....

Complaints are a time suck, but incredibly valuable. Good, ongoing, online policy-thrash can teach you a lot. It's both a pain and an asset.

While each of us makes a learning journey, we have to remain aware that the "reality" is evolving. That to believe you "get it" is not the most helpful stance in a lot of situations.

The fluid nature of virtual reality—even in a text-based version—is amazing.

-- <gail>

## Finding Electronic Mailing Lists Via a Listserv Database Search

If you don't need to monitor new mailing lists on a daily basis, there is still a way for you to take advantage of the new-list mailing list. The Listserv at North Dakota State University maintains a service that allows you to search for a particular keyword in text bases of Bitnet and Internet mailing list information.

### Searching the Databases

There are three databases upon which you can do a keyword search. Lists is a globally-known database of the Listserv mailing lists; intgroup is a searchable copy of a list of Bitnet and Internet interest groups; new-list is an indexed version of the new-list mailing list archives. While there is some overlap of the three different databases, it is generally a good idea to search all of them. By searching the new-list database, you get the benefit of all the previous messages, without having to monitor the list daily or look through all of the archive files.

The following is an example of a database search. Lines like those shown can be sent as a mail message to the address listserv@vm1.nodak.edu. A batch search takes advantage of Listserv Command Job Language (LCJL). However, it is not necessary to be fully conversant in the language. You can copy this example, use it as the basis for your queries, and easily

accomplish your searches.

```
//DBlook JOB Echo=No
```

```
Database Search DD=Rules (f=mail
```

```
//Rules DD *
```

```
Select music in lists
```

```
Index
```

```
Select music in intgroup
```

```
Index
```

```
Select music in new-list
```

```
Index
```

## An Example Search

In the example, all items that contain the string music anywhere in the text of an item would be listed in the output. After the Listserv processes your request, it sends you a mail message containing the results of your search. In the example search setup, the SELECT command is used to specify that music is to be located in items of each of the three databases. The INDEX command causes the display of a line of information for each match found in the database. The information returned is usually enough to enable you to subscribe to a particular mailing list; however, more detailed information can be obtained via an additional database job.

The following example shows some sample output from the previous search that used music as the keyword. The output has been edited for the sake of space, but you can still get an idea of how diverse a set of topics you can find with one search.

```
> Select music in lists
```

```
--> Database LISTS, 95 hits.
```

```
> Index
```

```
Ref# Listname Nodename #Sub List title
```

```
-----
```

```
0009 MPB-L BRUFPB 25 Lista para Musica Popular Brasileira
```

```
0049 ROCK TRITU 272 Rock&Roll Music Discussion List
```

```
0088 ALLMUSIC UFRJ 4 ALLMUSIC PEERED LIST
```

```
0117 GUM BRUFGM 79 Grupo de Usuarios MUSIC do Brasil (GUM)
```

```
0255 EARLYM-L AEARN 356 Early Music List
```

```
0310 KLARINET VCCSCENT 139 Klarinet - Clarinettist's Network
```

```
...
```

```
2569 MILES HEARN 240 Discussion of Jazz trumpeter Miles Davis & his vast+
```

2664 EMUSIC-D AUVM 948 Electronic Music Digest  
 2665 EMUSIC-L AUVM 430 Electronic Music Discussion List  
 2666 SYNTH-L AUVM 344 Electronic music "gearhead" list  
 2672 SERAVES AUVM 185 South East Rave List  
 ...  
 3284 MUSPRF-L CMSUVMB 153 Music Performance and Pedagogy  
 3347 PNWMLA-L UWAVM 23 PNWMLA-L  
 3387 IRTRAD-L IRLEARN 284 Irish Traditional Music List  
 3633 TRASHCAN UICVM 70 The Music of the Trash Can Sinatras  
 3645 ORTHODOX IUBVM 375 Orthodox Christianity  
 3650 FILMUS-L IUBVM 190 Film Music Discussion List  
 ...  
 4226 4AD-L JHUVM 811 4ad recording artists list  
 4388 FOLKDJ-L PSUVM 125 Folk and Bluegrass DJs  
 4542 BGRASS-L UKCC 676 Bluegrass music discussion.  
 4633 ETHMUS-L UMDD 418 EthnoFORUM, a global ethnomusicology forum.  
 ...

> Select music in intgroup

--> Database INTGROUP, 58 hits.

> Index

Item #	Date	Time	Recs	Subject
-----	----	----	----	-----
000013	94/02/28	00:00	28	78-L@cornellc.cit.cornell.edu
000027	94/02/28	00:00	23	adolph-a-carrot@ANDREW.CMU.EDU
000052	94/02/28	00:00	26	ALICEFAN on LISTSERV@WKUVX1.BITNET
000055	94/02/28	00:00	26	ALLMUSIC%AUVM.BITNET@VM1.NODAK.EDU

...

```

000210 94/02/28 00:00 19 Cloud-Zero@cs.mcgill.ca
000236 94/02/28 00:00 11 concrete-blonde@ferkel.ucsb.edu
...
000284 94/02/28 00:00 22 DEAD-FLAMES@VIRGINIA.EDU
000285 94/02/28 00:00 20 DEAD-HEADS@VIRGINIA.EDU
000298 94/02/28 00:00 17 dewy-fields@ifi.uio.no
000316 94/02/28 00:00 20 DJ-L%NDSUVM1.BITNET@VM1.NODAK.EDU
000333 94/02/28 00:00 32 EARLYM-L@AEARN.BITNET or
000359 94/02/28 00:00 31 EMUSIC-D%AUVM.BITNET@CUNYVM.CUNY.EDU
000397 94/02/28 00:00 36 FILMUS-L on LISTSERV@IUBVM.BITNET
000407 94/02/28 00:00 43 Folk_Music@nysernet.org
000408 94/02/28 00:00 29 FOLKTALK on LISTSERV@WMVM1.BITNET
...
000733 94/02/28 00:00 25 MILES on LISTSERV@HEARN.BITNET
000743 94/02/28 00:00 32 MLA-L%IUBVM.BITNET@VM1.NODAK.EDU
000747 94/02/28 00:00 42 Mmos2-L@knex.via.mind.ORG
000764 94/02/28 00:00 45 music-research%prg.oxford.ac.uk@NSS.CS.UCL.AC.UK
000765 94/02/28 00:00 14 mutex@stolaf.edu
000791 94/02/28 00:00 29 NEW-ORLEANS on MAIL-SERVER@mintir.new-orleans.la.us
...

```

> Select music in new-list

--> Database NEW-LIST, 99 hits.

> Index

Item #	Date	Time	Recs	Subject
000037	89/03/08	12:00	138	NEW LIST: HP-28@NDSUVM1 HP-28 Calculator Discussio+
000052	89/04/19	17:37	47	NEW LIST: EMUSIC-L or EMUSIC-D - Electronic Music
000059	89/05/10	12:54	28	NEW LIST: ALLMUSIC Music Discussions

...

000308	90/08/21	13:53	43	NEW LIST: OPERA-L Concerning All Opera and Lyrical+
000362	90/11/24	15:49	28	NEW LIST: VW5EARN@AWIWUW11.Bitnet Early Music List
000455	91/03/15	12:37	15	List Search: Twin Peaks
...				
000603	91/09/20	14:18	19	NEW LIST: Folk_Music list
000612	91/10/09	16:44	105	NEW-LIST List Search Digest - 9 Oct 1991
...				
000708	92/01/31	08:58	57	NEW LIST: Sun Ra
000713	92/02/03	20:13	30	NEW LIST: UPNEWS Update-Electronic-Music-News
000737	92/02/29	15:08	92	NEW LIST: Camelot - Arthurian Legend and Grail Lore
...				
000781	92/04/15	13:33	48	Address Change: Sun Ra and his Arkestra (avant-gar+
000785	92/04/23	19:53	28	NEW LIST: 78-L 78 RPM Records and Music Collectors
...				
001030	93/01/15	08:47	28	NEW LIST: PIPORG-L Pipe Organs and Related Topics
...				
001170	93/05/16	21:19	30	NEW: NEW-ORLEANS Mailing List
001209	93/06/14	20:10	32	NEW: kosmos - Paul Weller list
001223	93/06/17	17:07	25	SEARCH: Popular Culture
001227	93/06/22	12:31	16	NEW: grind - grindcore/death metal/heavy thrash mu+
001249	93/07/07	09:32	36	NEW: WNN - ABC's "World News Now" Discussion List
001276	93/07/21	16:15	30	NEW: ENO-L - Brian Eno
...				
001669	94/02/14	14:45	34	NEW: ACTMUS-L - Asian Contemporary Music
001670	94/02/14	15:07	26	NEW: PIANOMAN - Billy Joel
001699	94/03/03	10:31	74	NEW: SIXTIES-L

```

001715 94/03/11 11:49 27 NEW - SIDEDRUM - International list for Scottish D+
001733 94/03/21 17:33 41 NEW: flu-list - Band info list run by the band.
001740 94/03/29 08:22 21 NEW: Shot-of-Rhythm - the music of John Hiatt
001785 94/04/20 09:00 55 CHANGE: Lists at Ferris have moved
001788 94/04/20 14:29 24 SEARCH: Crash Test Dummies

```

## Analyzing the Search Output

Note that there are three sections to the previous example, each one corresponding to a search request you made. Because the results are sent via e-mail, the lines are limited to 80 characters, so some of the descriptions are truncated. The truncation is indicated by a plus sign (+). If you actually perform this search, you will notice that many of the matches again refer to the McGill University MUSIC (Multi-User System for Interactive Computing) operating system (long a source of some confusion; it has nothing to do with the aural kind of music). You can subscribe to the Listserv mailing lists by sending the usual SUBSCRIBE command to the nearest or the host Listserv installation. Other kinds of mailing lists may have differing instructions on how you can join.

## Getting More Information by Retrieving Archives

In the case of the preceding search example, and with a little more work on your part, additional information can be retrieved from listserv@vm1.nodak.edu. The items actually refer to postings from the NEW-LIST mailing list, which is maintained there. You can send for a log file containing the particular message based upon the information returned by the database search. To do so, send the following command as the first line of a mail message to the Listserv: SEND NEW-LIST LOGyymm, where instead of the letters yymm you substitute the corresponding month and year found in the Date column of the database search output. Note that because they are generated from the NEW-LIST mailing list, these log files are available only for searches on the NEW-LIST database and not the other databases.

## Printing Database Items

Rather than having to look through an entire log file, however, it would be more convenient to just select the complete text of interesting items. Fortunately, the PRINT command of a database search enables you to do so. When doing a batch database search, a PRINT command is usually used in a second job because an INDEX command is needed to find item numbers for particular elements of the database. In a subsequent search following the theme used so far, suppose you were particularly interested in jazz music. You could receive the entire text of the item by referencing its index number (under the heading Ref# in the lists database and under Item # in the other two databases). Because a PRINT command often returns one or more pages of text, you should probably limit the number of items you request in order to save e-mail bandwidth as well as your own mailbox. The following example shows the job setup for just such a search.

```
//DBlook JOB Echo=No
```

```
Database Search DD=Rules (f=mail
```

```
//Rules DD *
```

```
Select music in lists
```

```
Print 2569 3633
```

```
Select music in intgroup
```

```
Print 791
```

```
Select music in new-list
```

Print 708 781

## Some Example Database Hits

In pursuing this example, the Miles mailing list, the Trashcan mailing list, the New Orleans list (after all, there's no place for jazz like New Orleans), and the Sun Ra mailing list may all have piqued your interest. The resulting output of the preceding search job setup is shown here.

```
> Select music in lists
```

```
--> Database LISTS, 95 hits.
```

```
> Print 2569 3633
```

```
>>> Item number 2569, dated 94/04/18 -- ALL
```

```
*
```

```
* Discussion of Jazz trumpeter Miles Davis & his vast musical career
```

```
*
```

```
* Review= Public
```

```
* Subscription= Open,Confirm
```

```
* Send= Public
```

```
* Notify= No
```

```
* Files= No
```

```
* Reply-to= list,ignore
```

```
* Validate= No
```

```
* X-tags= Comment
```

```
* Notebook= Yes,L/301,Monthly,Public
```

```
* Stats= Normal
```

```
* Errors-To= Owner
```

```
* Mail-Via= Dist2
```

```
*
```

```
* Owner= TBUCK@KNOX (Tom Buck)
```

```
*
```

```
>>> Item number 3633, dated 94/03/08 -- ALL * The Music of the Trash Can Sinatras
```

```
* SEND= Public
```

\* REVIEW= Public  
\* SUBSCRIPTION= Open  
\* NOTIFY= Yes  
\* REPLY-TO= List,Respect  
\* FILES= No  
\* VALIDATE= STORE ONLY  
\* ACK= NO  
\* CONFIDENTIAL= No  
\* NOTEBOOK= YES,L,monthly,public  
\* OWNER= U52549@UICVM  
\* EDITOR= U52549@UICVM  
\* ERRORS-TO= U52549@UICVM  
\* Renewal= 1-Yearly  
\* Mail-via= DIST2

> Select music in intgroup

--> Database INTGROUP, 58 hits.

> Print 791

>>> Item number 791, dated 94/02/28 00:00:00 -- ALL

Date: 28 Feb 1994 00:00 GMT

From: <interest-groups-request@nisc.sri.com>

To: INTGROUP@VM1.NoDak.EDU

Subject: NEW-ORLEANS on MAIL-SERVER@mintir.new-orleans.la.us

NEW-ORLEANS on MAIL-SERVER@mintir.new-orleans.la.us [Last Update 5/93]

NEW-ORLEANS is a mailing list for discussing the city of New Orleans, it's history, culture, people, food, etc. Restaurant reviews, music notes, political comments, and general potpourri dealing with the city are welcome.

A companion list, NEWORL-DIG, is also now available. This will be a monthly (or so) digest of the NEW-ORLEANS mailing list. It will always include at least one or two restaurant reviews, local entertainment news, and other tidbits.

To subscribe to NEW-ORLEANS, send the following command to MAIL-SERVER@mintir.new-orleans.la.us:

```
SUBSCRIBE NEW-ORLEANS
```

To subscribe to NEW-ORLEANS, send the following command to MAIL-SERVER@mintir.new-orleans.la.us:

```
SUBSCRIBE NEWORL-DIG
```

Owner: Edward J. Branley <elendil@mintir.new-orleans.la.us>

```
> Select music in new-list
```

```
--> Database NEW-LIST, 99 hits.
```

```
> Print 708 781
```

```
>>> Item number 708, dated 92/01/31 08:58:48 -- ALL
```

Date: Fri, 31 Jan 1992 08:58:48 CST

Reply-To: SATURN@KNOX

Sender: NEW-LIST - New List Announcements <NEW-LIST@NDSUVM1.BITNET>

From: SATURN@KNOX.BITNET

Subject: NEW LIST: Sun Ra

Sun Ra via Saturn@KNOX.BITNET

Well, as you can tell by the address from which this message originates,

I am finally starting a Sun Ra mailing list! Topics for discussion will

include:

- \* Sun Ra's music, albums, videos, poetry, and philosophy. All things Ra!
- \* Magazine and newspaper articles or books that talk about Sun Ra and/or contain interviews are also game.
- \* Members of the Arkestra and album projects with which they have been involved.
- \* Other artists who are similar to Sun Ra, or who claim Sun Ra as an influence.
- \* Free Jazz in general, and other types of Jazz as they relate to Sun Ra's music.

For those of you who may not know: Sun Ra is a jazz musician who leads a big band and claims to be from the planet Saturn. He has been doing such for the past 35-40 years and is nearing 80 years of age. He is also a poet and even, in a sense, a spiritual leader. He has well over 100 albums to his credit (with a little over a dozen on CD) and I hope to maintain a discography of Sun Ra recordings. Perhaps a list of magazine and newspaper articles and books dealing with Ra can also be developed.

This is not a Listserv-type list, so I will be running it by hand. All administrative-type e-mail should be sent to Saturn@Knox.Bitnet as well as submissions for the list. I hope to get a Sun Ra digest out every other day or every third day as needed. If you know of any other Sun Ra fans who have electronic mail capabilities, please feel free to forward this message on to them.

To subscribe to the Sun Ra mailing list, just send a note to me at this account: Saturn@Knox.Bitnet. For those of you who are trying to send from networks other than Bitnet: If you have trouble getting to that address, try sending to Saturn%Knox.Bitnet@uicvm.uic.edu. I won't be able to send out the first digest for another week or so, but go ahead and send notes asking to be added to the mailing list, or submissions for the first digest.

Owner:

Tom Buck <Saturn@Knox.Bitnet>

editor of the Sun Ra mailing list. "Sound Can Wash Clothes"

- Sun Ra -

By comparison, here is the same item from the lists database.

>>> Item number 781, dated 92/04/15 13:33:55 -- ALL

Date: Wed, 15 Apr 1992 13:33:55 CDT

Reply-To: TBUCK@KNOX

Sender: NEW-LIST - New List Announcements  
<NEW-LIST@NDSUVM1.BITNET>  
From: TBUCK@KNOX.BITNET  
Subject: Address Change: Sun Ra and his Arkestra  
(avant-garde jazz)

The Sun Ra mailing list, formerly at <Saturn@Knox.Bitnet>, has moved to <Saturn@Hearn.Bitnet> and is now a real-time list utilizing the ListServ software at Hearn. To subscribe, send a message to <Listserv@Hearn.Bitnet> with the following as the body of the message....

Subscribe Saturn Your Name

(for example)

Subscribe Saturn Tom Buck

Topics for discussion include:

- \* Sun Ra's music, albums, videos, poetry, and philosophy. All things Ra!
- \* Magazine and newspaper articles or books which talk about Sun Ra and/or contain interviews are also game.
- \* Members of the Arkestra and album projects with which they have been involved.
- \* Other artists who are similar to Sun Ra, or who claim Sun Ra as an influence.
- \* Free Jazz in general, and other types of Jazz as they relate to Sun Ra's music.

For those of you who may not know: Sun Ra is a jazz musician who leads a big band and claims to be from the planet Saturn. He has been doing such for the past 35-40 years and is nearing 80 years of age. He is also a poet, and even a spiritual leader in some sense. He has well over 100 albums to his credit (with almost a dozen on CD) and I hope to maintain a discography of Sun Ra recordings. Perhaps a list of magazine and

newspaper articles and books dealing with Ra can also be developed.

Happy Trails and Later Days....

Tom Buck (TBuck@Knox.Bitnet), owner of the Sun Ra mailing list.

"Sound Can Wash Clothes"

- Sun Ra -

You'll immediately notice that the lists database offers far less descriptive information than the other two. At this point you might give up on finding more about the Trash Can Sinatras (I wonder if Frank knows about them?), but more information on the Miles list can be found in the intgroup database. With the additional information from a search-and-print job, you are ready to subscribe to the lists that interest you (see the following note).



**Know Your Mailing List References:** After you have a mailing list description, the next step is to subscribe. When examining a mailing list reference, make note of the following information:

*Subscription address.* in the case of Listserv facilities, subscription commands can be sent to the particular Listserv installation that maintains the list or to the closest Listserv installation.

*List format.* some mailing lists broadcast all messages sent to them, while on others a list moderator places the messages into digest form before they are sent to you. Digest mailing lists usually generate much less mail traffic than their "undigested" counterparts.

*List distribution.* One piece of information available is the various Listservs at which a particular list is maintained. In general, the more sites at which a list is maintained, the more popular that list is, and therefore, the more mail traffic it will generate.

*List moderator.* Most list descriptions include the name of the moderator and the moderator's network mailing address. Whether you subscribe to a list or not, the moderator can sometimes be helpful in providing additional information about that list, or suggesting alternate sources of information.

After you absorb this information, accessing network mailing lists is quite simple.

---

## Some Additional Benefits of the Listserv Database Capability

In the preceding example, we searched the new-list archives for mailing lists of interest. At many Listserv sites, archives of other mailing lists are maintained, and searchable databases of those archives are also available. These archives, combined with the Listserv database capability, can be quite informative when you are looking for the answer to a particular question. Instead of posting your question to a topical mailing list, you can search that list's archives to see if that question or a similar one has been posted before.

### Finding Other Databases

There is currently no global listing of all Listserv databases, but mailing-list archives and databases are generally maintained on the Listserv installation that supports that list. You can find out what databases are available at a particular Listserv site by sending the command DATABASE LIST in a mail message. With that information, you can design a search job to look for your question. The following is a typical list of Listserv databases.

<i>Database</i>	<i>Description</i>
ACA-L	Archives of "ACA-L"

ADA-LAW	Archives of "ADA Law"
ADVISE-L	Archives of "User Services List"
AG-EXP-L	Archives of "AG-EXP-L Ag Expert Systems"
AXIOM	Archives of "AXIOM Computer Algebra System"
BIOMED-L	Archives of "BIOMED-L Biomedical Ethics"
BLINDNWS	Archives of "Blind News Digest"
DOWN-SYN	Archives of "Down Syndrome"
FUSION	Archives of "Fusion - Redistribution of sci.physics.fusion"
GEDCOM-L	Archives of "Genealogical Data Communications Specs"
GRAPHNET	Archives of "GRAPHNET - Graph Theory"
GUISPEAK	Archives of "Access to GUI via Speech"
HOMESAT	Archives of "HOMESAT - Home Satellite Technology"
JES2-L	Archives of "JES2 discussion group"
LINES-L	Archives of "LifeLines Genealogical System"
LISTS	Information on all the network-wide Listserv lists
LORE	Archives of "LORE - Folklore List"
PEERS	Information on all the Listserv servers in the network
POWER-L	Archives of "POWER-L IBM RS/6000 POWER Family"

## Searching a Mailing List Archive

Statements for searching an archive might look like the following (remember that a batch search needs additional LCJL statements preceding these lines):

```
Select home remedies in lore
```

Index

In the preceding example, index lines for items matched on both words (independent of position within the item) would be returned. The full text could then be retrieved using the PRINT command as previously discussed.

## Finding Listserv Installations

Most Listserv sites maintain a *peers* database, which contains information on all known Listserv installations. To find all the installations in a particular U.S. state, you could use the following statements:

```
Select Texas in peers
```

Index

To list all known installations, use the statements:

```
Select * in peers
```

Index

In this case, the asterisk (\*) acts as a wildcard and will return a line for each entry in the database. Wildcard searches should be done judiciously; they can take quite a bit of processor time on the Listserv machine and generate quite a bit of output to your electronic mailbox.

The database capabilities of Listserv are too extensive to allow complete documentation in this discussion. For more information on Listserv database searches, see the file LISTDB MEMO, by Eric Thomas, available from the closest Listserv installation by sending the command GET LISTDB MEMO.

## Finding the Usenet Newsgroup you Want

Finding Usenet newsgroups is not usually a problem if you have access to a news reading program and a corresponding news feed. Groups will be there available to you to read—several thousand, in fact. Finding a group or the messages that interest you can be a daunting challenge when you are faced with the volumes of messages typically available via a news reader.

Fortunately, many news readers support a search function that allows you to find specified strings in the subject or From: field of a posted message. With such a capability, you can look for specific topics or for postings from specific people, without reading the entirety of one or more newsgroups.

### An Example nn News Reader Search

Figure 12.6 shows a Usenet search using the nn news reading software as an example. The command that generated this multiscreen output was `nn -mxX -sjazz`, which tells nn to search for all occurrences of the word *jazz* in the subject field of messages and merge the found messages into a metagroup (that is, collect them for reading all at once without having to enter each newsgroup). The `x` and `X` options work with the `-m` (meta) option to instruct nn to perform its search for both read articles (`x`) as well as in unsubscribed groups (`X`). In this case, the intended target is information about jazz music—however, the basketball team by that name shows up in the search as well.

```
a GibsonL5          17  SOLO JAZZ GUITAR
b FITZGERALD        5   Jazz Discography
c linda brown       36  -
d Jeff Vineburg     10  Jazz style pick-ups
e Jack A. Zucker    32  >
f Ammar Aganovic   117  >Suggestions needed for Mexican Jazz mods
g Zoltan Janosy     15  >Wanted: Midi files of jazz standards
h Tim Llewellyn     52  >>African Videos (was Re: T<>ouring late
summer/early aut
i Roger Brown       10  >>>
j Tim Llewellyn     27  >>>>
k Gary Allen P000   36  >North Carolina Jazz l Steve A. Chall      21  >
m JOHN BARR         5   >>
n Vincent Pelote    13  IJS Jazz Register and Indexes update
o K Laufer          35  >>Latin Jazz recs
p paul drobny       47  >>
q Alan Cibils       26  >
r sheldon White     12  >
s Anthony Corman    15  >>>
```

-- 12:34 -- SELECT -- help:? -----Top 15%-----Read 247644 articles in 618 seconds (32 kbyte/s)

MERGED NEWS GROUPS: 119 ARTICLES

a,R. Lynn Rardin 11 >>Latin Jazz recs  
b dld1980@pan.com 6 >  
c Steve Larsen 22 >>Dirty Play; Thankfully the Jazz Won  
d Marc Sabatella 49 >>>>chordless jazz  
e Stephen Guattery 33 >>>>  
f Ed Price 59 >>>>Suggestion for New York jazz club?  
g m j rosenstein 15 >>>  
h Lynn Goods 9 >>>>  
i Doug Tygar 9 >>>>  
j Ken McCarthy 13 >>>>>  
k Ed Price 54 >>>>  
l m j rosenstein 36 >>>  
m dld1980@pan.com 6 >  
n Gary Allen P000 23 >jazz  
o Chris Sipe 25 >>  
p sl8gn@cc.usu.edu 19 >  
q BROWN M SAXON 21 >  
r Coz 7 >>  
s crutcher@woods 40 -

-- 12:40 -- SELECT -- help:? -----31%-----

MERGED NEWS GROUPS: 119 ARTICLES

a John M. Black 31 Minn. jazz scene..  
b Sten W Johnson 14 >  
c Leo Chan 7 Utah Jazz Superteam for SALE - \$15 OBO  
d Travis X. Emmitt 18 CHAMELEONS / JAZZ BUTCHER

e Ed Price 12 >>>>What is Jazz ? (was: What is fusion?)  
f AFC PeterS 12 >  
g Richard Metcalf 18 IMPROVIJAZZATION NATION  
h Richard Metcalf 18 -  
i Kevin Jundt 23 [\*] Acid Jazz 1.2v2; a phone dialer  
j Matt Smith 15 >>Jazz newsgroup  
k C Koerner 111 >JAZZ in the German Culture  
l Ashot W Bakunts 28 >Manhattan Jazz Quintet  
m william.j.heryy 22 >>  
n Andrew Anda 29 >>Starting Jazz violin  
o sldqv@cc.usu.edu 6 JAZZ wins !!!  
p Jowett Garth S. 14 ManHattan Jazz Quintet/Septette  
q Jowett Garth S. 23 ManHattan Jazz Septette  
r MODESTOG@DELPHI 125 SPURS REPORT: <SPURS CHOKE(ED) BY JAZZ> s Rod Davies  
7 Jazz Gifs/Image files

-- 12:40 -- SELECT -- help:? -----47%-----

Andrew Anda: >>Starting Jazz violin  
article <KEITH.94May2161712>

5 May 1994 17:13 In

(George Keith) writes:

|>

|> I was wondering if anybody out there has any suggestions that might

|> help a classical violinist/Irish fiddler get familiar with the

|> concepts of Jazz playing? Books? Good people to listen to (I know of

|> Grapelli)? Attitudes? Personal experience?

|>

A couple of books to check out are:

Jazz Violin Studies, by Usher Abell, pub. by Mel Bay;

Jazz Violin, by Glaser & Grappelli, pub. by Oak;

related style books:

Blues Fiddle, by Lieberman, pub. by Oak;

Vassar Clements Fiddle, pub. by Oak;

Contest Fiddling, by Phillips, pub. by Mel Bay.

All of the above have LOTS of notes with varying degrees of theory, history, discography, and biography thrown in.

-- 12:41 --rec.music.makers-- 1 MORE --help:?--Top 71%--

All of the above have LOTS of notes with varying degrees of theory, history, discography, and biography thrown in.

You may want to focus on a particular jazz style and non-violin instrument (including possibly voice) to endeavor to emulate, e.g., Ponty, who was heavily influenced by the horn style of Miles Davis, has quite a different sound from Smith, Venuti, and Grappelli, the "old-guard" swing masters.

G'DAE,

-- 12:41 --rec.music.makers-- 1 MORE --help:?--Bot--

After looking through three screens of the 119 messages matched on this topic, you might be inclined to read the message about Starting Jazz Violin (especially if you happen to be a violinist). The same technique already described can be used to search newsgroups for a variety of topics. One advantage of newsgroups versus mailing lists is that they provide large amounts of information all at once. There is so much information, however, that finding what you want can sometimes be a problem, even if you know in which groups you should look. Using a news reader with a search capability can greatly enhance the process of finding messages appropriate to your interest.

### Finding Newsgroup Messages via Archie

If you don't have access to Usenet News or you don't have access to a complete news feed, you might want to use Archie to find newsgroups. Many sites maintain newsgroup archives, so you may not even need a news reader to access some information, however, you will need telnet or an Archie client (to use Archie) and FTP access (to retrieve the information). Because it is logical to expect that there might be newsgroups about music with a name like something.music.something, you might want to construct an Archie search using a regular expression like the following:

```
archie -m 500 -r '.*\[Mm]usic.*'
```

An edited version of the resulting output is shown here.

Host ftp.diku.dk

Location: /pub/doc

```
FILE -rw-r--r--      70849  May 21 1992  FAQ-comp.music
```

Location: /pub/music/guitar/r

```
    DIRECTORY drwxr-xr-x      1024  Jan  9 03:15  Romantic.Music
```

```
    DIRECTORY drwxr-xr-x      1024  Oct 11 20:49  Roxy.Music
```

Host wilma.cs.brown.edu

Location: /u/mlm/atarraya/SIC/abril

FILE -rw-r--r-- 7221 May 16 1993 festival.musica

Host ftp.cco.caltech.edu

Location: /pub/humor/canonical.lists

FILE -rw-r--r-- 66167 Feb 9 19:35 canonical.music Host  
ftp.cis.ksu.edu

Location: /pub/NetBSD/NetBSD-current/src/usr.bin/calendar/calendars FILE -r--r--r--  
5450 Dec 17 07:05 calendar.music.Z

Host ftp.uu.net

Location: /doc/music/guitar/r

DIRECTORY drwxrwxr-x 512 Jun 14 1993 Roxy.Music

DIRECTORY drwxrwxr-x 512 Apr 9 1993 romantic.music

Location: /systems/apple2/gs/gsos/nda/old

FILE -rw-r--r-- 16665 Jul 10 1992 background.music.bsq.Z Location:  
/systems/unix/bsd-sources/usr.bin/calendar/calendars

FILE -r--r--r-- 5450 Oct 27 1989 calendar.music.Z

Location: /usenet

DIRECTORY drwxr-xr-x 1024 Jan 25 1993 alt.music.alternative

DIRECTORY drwxr-xr-x 512 Jan 25 1993 alt.music.progressive

DIRECTORY drwxr-xr-x 512 Jan 25 1993 alt.music.rush

DIRECTORY drwxr-xr-x 512 Jan 25 1993 alt.music.tmbg

DIRECTORY drwxr-xr-x 512 Jan 25 1993 aus.music

DIRECTORY drwxr-xr-x 512 Jan 25 1993 comp.music

Location: /usenet/control/alt

FILE -rw-rw-r-- 1776 Jun 19 1993 alt.binaries.sounds.music.Z

FILE -rw-rw-r-- 453 Apr 6 1993 alt.journalism.music.Z

FILE -rw-rw-r-- 1333 Apr 3 1993 alt.music.a-cappella.Z

FILE -rw-rw-r-- 2405 Nov 27 1992 alt.music.alternative.Z

FILE -rw-rw-r-- 770 Aug 16 1993 alt.music.bela-fleck.Z

```
FILE -rw-rw-r--      1143  Apr  6 1993  alt.music.canada.Z
FILE -rw-rw-r--      6016  Sep 14 1992  alt.music.progressive.Z
FILE -rw-rw-r--           731  Jul  7 1992  alt.music.queen.Z
FILE -rw-rw-r--      1601  Nov 27 1992  alt.music.rush.Z
FILE -rw-rw-r--           568  Dec 12 1992  alt.music.ska.Z
FILE -rw-rw-r--      3740  Apr 27 1992  alt.music.the.police.Z
FILE -rw-rw-r--      2129  Apr 26 1992  alt.music.the.police.ct1.Z
FILE -rw-rw-r--      1041  Nov 11 1992  alt.music.tmbg.Z
FILE -rw-rw-r--      1002  Jun  4 1993  alt.music.world.Z
```

Location: /usenet/control/ba

```
FILE -rw-rw-r--           379  Apr  1 1992  ba.music.Z
FILE -rw-rw-r--      1480  May 19 1993  ba.music.drumming.Z
```

Location: /usenet/control/cle

```
FILE -rw-rw-r--           494  Mar  1 1993  cle.music.Z
```

Location: /usenet/control/comp

```
FILE -rw-rw-r--           829  Apr 26 1992  comp.music.Z
```

Location: /usenet/control/k12

```
FILE -rw-rw-r--           363  Aug 11 1991  k12.ed.music.Z
```

Location: /usenet/control/phl

```
FILE -rw-rw-rw-           391  Jan 20 1993  phl.music.Z
```

Location: /usenet/control/rec

```
FILE -rw-rw-r--      1183  Jul 27 1993  rec.music.a-cappella.Z
FILE -rw-rw-r--      1868  Oct 21 1992  rec.music.afro-latin.Z
FILE -rw-rw-r--           669  Apr 26 1992  rec.music.beatles.Z
FILE -rw-rw-r--           678  Apr 26 1992  rec.music.bluenote.Z
FILE -rw-rw-r--           653  Apr 26 1992  rec.music.cd.Z
FILE -rw-rw-r--      2095  Apr 26 1992  rec.music.christian.Z
FILE -rw-rw-r--           659  Apr 26 1992  rec.music.classical.Z
```

```

FILE -rw-rw-r--      1604  Oct 27 1992  rec.music.classical.guitar.Z
FILE -rw-rw-r--      1113  Jul 26 1993  rec.music.classical.performing.Z
FILE -rw-rw-r--          353  Oct 27 1992  rec.music.compose.Z
FILE -rw-rw-r--          688  Apr 26 1992  rec.music.country.western.Z
FILE -rw-rw-r--          659  Apr 26 1992  rec.music.dementia.Z
FILE -rw-rw-r--          658  Apr 26 1992  rec.music.dylan.Z
FILE -rw-rw-r--      2251  Oct 21 1992  rec.music.early.Z
FILE -rw-rw-r--          660  Apr 26 1992  rec.music.folk.Z
FILE -rw-rw-r--      1905  Apr 26 1992  rec.music.funky.Z
FILE -rw-rw-r--      2258  Apr 27 1992  rec.music.gaffa.Z
FILE -rw-rw-r--          653  Apr 26 1992  rec.music.gdead.Z
FILE -rw-rw-r--          655  Apr 26 1992  rec.music.makers.Z
FILE -rw-rw-r--          625  Nov  9 1992  rec.music.makers.bass.Z
FILE -rw-rw-r--          630  Nov  9 1992  rec.music.makers.guitar.Z
FILE -rw-rw-r--          648  Nov  9 1992
rec.music.makers.guitar.tablature.Z
FILE -rw-rw-r--          955  Jun 18 1993  rec.music.makers.marketplace.Z
FILE -rw-rw-r--          654  Nov  9 1992  rec.music.makers.percussion.Z
FILE -rw-rw-r--      1072  Feb 16 1993  rec.music.makers.synth.Z
FILE -rw-rw-r--      1578  Apr 26 1992  rec.music.marketplace.Z
FILE -rw-rw-r--          640  Apr 26 1992  rec.music.misc.Z
FILE -rw-rw-r--          650  Apr 26 1992  rec.music.newage.Z
FILE -rw-rw-r--      1664  Apr 26 1992  rec.music.phish.Z
FILE -rw-rw-r--          731  Dec 21 1992  rec.music.reggae.Z
FILE -rw-rw-r--      4774  Apr 27 1992  rec.music.reviews.Z
FILE -rw-rw-r--      1047  Feb 16 1993  rec.music.synth.Z
FILE -rw-rw-r--      1919  Apr 26 1992  rec.music.video.Z

```

As you can see from the example, many alt.music groups were found, as were many rec.music newsgroups. The example

output has been edited to save space; in reality, many more items were matched in this search. The files ending with the .Z suffix are probably compressed archive collections. With creative use of Archie and FTP, you may be able to find many Usenet postings without ever cracking a news reader.

## Finding Mailing Lists and Newsgroups in Gopher

The volume of Gopher servers (called Gopherspace) currently occupying the Internet is becoming a good place to look for information generated originally in mailing lists and newsgroups. Some Gopher menu items actually echo UseNet groups, providing each posted message as a separate file. For example, you might find a jobs (like misc.jobs) group included in a Gopher's Employment submenu. This is particularly appropriate for moderated groups and digests, because most Gopher servers do not enable you to post information. So far, Gopher is mostly used as a one-way information channel, with the user/client on the receiving end.

### Finding Lists in Gopher Menus

You can browse a Gopher server's menu structure to find list and group echoes, but on most Gophers you can do a Jughead search to find local menu items containing a particular word or several words. Sometimes you can't avoid tripping over an appropriate menu, like the Discussion Groups menu item on the University of Minnesota Gopher (literally, the "mother of all Gophers"). Such a menu might have information only about mailing lists and newsgroups, but often actual news items will be included as well.

### Using Veronica to Search Gopherspace for Newsgroups

A Veronica search often yields positive results as well. The following screen dump shows the results of such an effort (edited to conserve space). The example search phrase is Discussion Groups, and the output shows that this is a popular item in Gopherspace. As the example indicates, after a display of 200 items, there are still 297 more matches to be found.

```
Search gopherspace: discussion groups
```

1. "Slow-Reading" discussion groups
2. Rec.games.frp.\*: Welcome to the roleplaying discussion groups!
3. TV Discussion Groups, Etc. (monthly posting)
4. Compendium of Internet Discussion Groups
5. C-International Womens Discussion Groups
6. Discussion Groups/
7. Other Discussion Groups/
8. Mycology Discussion Groups via Internet
9. About LISTSERV Discussion Groups
10. About Electronic Discussion Groups
11. Electronic Discussion Groups/
12. Ch 3. Electronic Discussion Groups/Forums
13. News/Discussion Groups
14. Discussion and News Groups/

15. Gaggle of Discussion Groups for Diverse People
16. Gaggle of Discussion Groups for Diverse People
17. Law-Related Discussion Groups, Publications/
18. Law-Related Discussion Groups, Publications/
19. News and Discussion Groups/
20. News and Discussion Groups/
21. Electronic Discussion Groups
22. Electronic News and Discussion Groups/
23. Discussion groups/
24. LISTSERV Discussion Groups ... (Harris)

...

195. Electronic discussion groups in Women's Studies/
196. Electronic discussion groups in Women's Studies/
197. Academic Discussion Lists and Interest Groups/
198. Academic Discussion Lists and Interest Groups/
199. Discussion Groups (Listservs)/
200. Discussion Groups (Listservs)/
201. \*\* There are 297 more items matching the query "discussion groups" ..

## An Example Veronica Search

Another interesting exercise is to perform a Veronica search on a particular interest topic, such as we did for mailing list databases above. To return to the topic of music, the following example shows what's available in Gopherspace in that regard. Once again, to save on space the output has been pared down from the original.

Search gopherspace: music

1. music
2. music
3. Music Cataloguing for Nonspecialist
4. Music Materials in Libraries
5. Computer-Assisted Instruction for Music Uniform Titles.
6. Medieval & Renaissance Music

7. Multimedia Music Project

...

19. music\_bluenote\_faq

20. music\_bluenote\_sources

21. music\_bluenote\_welcome

22. music\_classical-faq

23. music\_composition-FAQ

...

69. Alt.music.alternative FAQ -----\\----- FAQ:

70. Alt.music.alternative FAQ part one of two -----\\----- FAQ:

71. Alt.music.alternative FAQ part two of two -----\\----- FAQ:

72. [alt.music.enya] FAQ (Monthly Posting) -----\\----- FAQ

73. Alt.music.james-taylor -----\\----- FAQ: ...

104. Rec.music.info: List of Music-Oriented Fan Clubs

105. Rec.music.info: List of Music-Oriented Fan Clubs

106. Rec.music.info: List of Internet Musical FTP Sites ...

116. Music Notation Programs - a list to answer a FAQ

117. Rec.music.classical.performing - FAQ

118. Rec.music.classical.performing! -----\\----- Welcome to

119. Rec.music.info: Submission Guidelines for rec.music.info

...

186. Music - All Music Guide (Ferris State Univ)/

187. University of Utah Music Gopher/

188. Master Gopher Server @ Univ. of Minnesota (Music)/

189. Music - The American Music Resource - Via Sunsite Gopher Server/

190. Music - Bagpipe Archives - Dartmouth College Dept. of Computer Sci.../ ...

200. Preacher praises music that glorifies God

201. \*\* There are 9644 more items matching the query "music" available \*\*..

After the first 200 items in this search, there are still 9644 more to find! Music seems to be a popular item in Gopherspace. Most of what you find in the search results are documents and FAQ files, but some of these files can lead you to a list or group on a particular topic. Other menu items lead to submenus, which may contain echoes of news posts. In any event, when looking for a forum on a particular topic, don't overlook the tremendous volume of information in Gopherspace.

## Electronic Journals via Gopher

One of the most comprehensive collections of electronic journals is maintained and accessible via Gopher. CICNet has an ongoing project to collect, maintain, and catalog electronic serial publications. At last count, they have over 700 titles, most accessible via a Gopher connection. To find this collection, point your Gopher client at [gopher.cic.net](http://gopher.cic.net) and look in the e-serials menu selection. The following screen dump shows the menu as it appears in Gopher:

Electronic Serials

- ```
--> 1.  Read Me First!  
    2.  A List of What's Here  
    3.  Alphabetic List/  
    4.  General Subject Headings/  
    5.  About Electronic Publishing and E-Journals/  
    6.  About the CICNet Electronic Journal Project/  
    7.  Hypertext E-Journal Sampler/  
    8.  Other Journal Archives/  
    9.  Thank You!  CICNet's Archive Volunteers
```

You will find helpful information in the Read Me First! item and in the Thank You! item. The General Subject Headings menu item may be the place you want to start looking, in order to find a serial on a particular topic.





- [18](#)
  - [Gopher](#)
    - [The Gopher0 Protocol](#)
    - [Gopher+ Protocol](#)
    - [Client Tricks](#)
      - [Finding out Where You Are](#)
      - [Bookmarks](#)
      - [AFTP](#)
      - [Configuring Your Client](#)
    - [Server Tricks](#)
      - [Features of Different Servers](#)
      - [Mirroring Servers](#)
      - [Forms](#)
      - [Shepherd](#)
      - [Subject Menus](#)
    - [Gateways](#)
      - [Archie](#)
      - [NetFind](#)
      - [Whois](#)
      - [Webster](#)
      - [FTP \(Hack\)](#)
      - [News](#)
      - [X.500](#)
      - [Databases](#)
      - [Z39.50](#)
      - [Other go4gw Gateways](#)
      - [Gmail](#)
      - [Forms Gateway](#)
    - [Security Issues](#)
      - [Insecure Versions and Practices](#)
      - [Controlling Access to Your Server](#)
      - [Public Access Clients](#)
      - [AdmitOne Authentication](#)

- [Future Developments](#)
  - [3-D Gopher](#)
  - [Archie](#)

# 18

## Gopher

*by Billy Barron*

Gopher originally made a name for itself by being a simple, fast, reliable, and powerful protocol. However, the original version of Gopher (designed in 1990) ran out of steam for new features after only two years of existence. The protocol has been upgraded, with numerous gateways to work around the problems and limitations of Gopher. Also, many tools have been written to better manage Gopher.

The result is that Gopher is now simple for the user, but it can be complex if you look deep enough. Much of the complexity can be utilized only by a guru. In this chapter, I hope to give you the background needed to become a Gopher guru and to show off some of the tricks that gurus use with Gopher.

### The Gopher0 Protocol

The original version of the Gopher protocol is now known as the Gopher0 protocol. It is important for the guru to understand the basic protocol especially when writing or debugging clients, gateways and servers, but it is also useful in understanding how Gopher works and how to create links between Gophers.

Before looking at the protocol syntax, it is important to remember that all Gopher items have a Host, IP Port Number, Path, Type, and Name. The Host and IP Port define where the Gopher server is that has the item on the Internet. The Path tells where that item is on that Gopher server. The Type tells what type the item is. This is similar to a file type and is documented in Table 18.1. The Name is the name of the item as it should appear on the screen.

**Table 18.1. Gopher item types.**

| <i>Type</i> | <i>Meaning</i>          |
|-------------|-------------------------|
| 0           | ASCII file              |
| 1           | Directory               |
| 2           | PH phone-book server    |
| 3           | Error                   |
| 4           | Binhexed Macintosh file |
| 5           | DOS binary archive      |
| 6           | UUencoded file          |

|   |                                      |
|---|--------------------------------------|
| 7 | Index-Search item (WAIS, and so on)  |
| 8 | Telnet                               |
| 9 | Binary file                          |
| + | Redundant server (never implemented) |
| g | GIF                                  |
| I | Image file                           |
| T | TN3270                               |

The actual protocol is very simple. The client connects to the server and then requests a path name, followed by a <CR><LF>. The server then returns the item. When the server finishes sending the item, it closes the connection.

If the item being returned is a menu, then the following syntax is used with one menu entry per line:

```
Type<TAB>Name<TAB>Path<TAB>Hostname<TAB>Port<CR><LF>
```

The important variation on this syntax is that <CR><LF> may optionally be replaced by a <TAB> followed by some more data. This will be important when we talk about Gopher+. Gopher0 clients, however, should just ignore the additional data beyond that <TAB>. The client then uses the information to display a menu to the user.

You should have noticed that Gopher will open and close a connection for every item the client wants to see, instead of maintaining a static connection like some protocols (such as FTP). One of the effects of this is that the server keeps no memory of what the client has done previously. This is known as being a *stateless server*.

The effects of this design choice are very important to comprehend, if you are trying to become a guru. On the positive side, the server is very low overhead and is a much simpler program than if it kept state. The negative aspects are that the opening and closing of connections can be slow at times, and the lack of state makes implementing some features very difficult. Even so, Gopher tends to be very responsive, and the Gopher server uses little system resources.

## Gopher+ Protocol

The Gopher0 protocol has served its purpose well and continues to do so. By mid-1992, though, it became apparent that Gopher had some limitations and could not handle some of the demands being placed on it. The Gopher Team at University of Minnesota (UMN) designed a new version of the Gopher protocol called Gopher+ to meet these demands. Some of the features that were added in were forms, identification of server administrators, password protection on selected items, and alternative views of a document.

One of the important design criteria of Gopher+ was that it be totally compatible with Gopher0—so compatible that Gopher+ clients could talk to Gopher0 servers and Gopher0 clients could talk to Gopher+ servers, with no changes to the Gopher0 software. Unfortunately, while the Gopher+ protocol meets this design goal, it is not as clean-cut and simple as the basic Gopher protocol.

Gopher servers return information in the following format *if* Gopher+ information is associated with the item instead of the Gopher0 format given previously:

```
Type<TAB>Name<TAB>Path<TAB>Hostname<TAB>Port<TAB><+>Gopher+ Information
```

Gopher+ also has some error-handling features in the protocol, but this section ignores them; usually, only

client-and-server authors need to know this part of the protocol. This section focuses instead on the most important feature of Gopher+, which is attributes.

Every Gopher+ item has at least two attributes associated with it, although in theory it can have an infinite number. Each attribute has an attribute block that describes it. Normally, the client does not see the attributes unless it specifically asks for them. The client requests them by adding a TAB and a ! to the end of the request to get all of them, as in the following:

```
0Welcome to Texas Metronet <TAB>!
```

At times, however, the client may be interested in only some attributes. In that case, include the TAB and the !, then add a + and the name of the attribute. Some sample attribute blocks might look like the following:

```
+INFO: 0Welcome to Texas Metronet<TAB>howdy<TAB>metronet.com<TAB>70<TAB>+
```

```
+ADMIN:
```

```
Admin: Mart Hacker <hacker@metronet.com>
```

```
Mod-Date: 1 July 1994 <19940701133333>
```

```
+VIEWS:
```

```
Text: <5K>
```

```
Postscript: <10K>
```

```
Image/GIF: <7K>
```

```
+ABSTRACT:
```

```
This is the Texas Metronet Welcome document.
```

In this attribute block, the INFO block gives the Gopher descriptor string. INFO is a required attribute for all Gopher+ items and must come first.

Next is the ADMIN block, which contains administrative information. It is also required for every Gopher+ item and must contain at least the ADMIN and the Mod-Date, for Modification Date, fields. The ADMIN field contains someone who you can complain to if there is a problem with the item. The Mod-Date lets you know when the item was last updated. The Mod-Date first contains a human-readable form, and then the date in a machine-readable format of *YYYYMMDDhhmmss* where *YYYY* is the year, *MM* is the month, *DD* is the day, *hh* is the hour, *mm* is the minute, and *ss* is the second. Other fields that may show up here include Score (used by WAIS searches), Score-range, Site, Organization, Location, Geographic, Time Zone, Provider, and Author (though these tend to be used infrequently).

The VIEWS block is very important. It allows a document to be available in multiple format with the client picking the format it wants. This block tells the client which formats or views are available and the file size of each. In the previous example, the document is available as a text file, Postscript, or a GIF image file. The views are described as MIME content types.

The ABSTRACT attribute block allows an item to contain a short abstract. This helps the Gopher user see what the document is about before downloading it. Unfortunately, this item is used rarely.

## Client Tricks

This section discusses tricks commonly used by Gopher gurus in using Gopher clients. It attempts to be as client-independent as possible. However, because many gurus are UNIX users and the UMN UNIX Gopher client is probably the most common Gopher client in use, this section covers it while showing tricks that can be performed by Gopher clients. If you are using another client, you will need to try to figure out how to do the equivalent with your client. However, some clients may not have some of the features discussed. Your only choices at that point are to live without the feature or to install another client package. In addition, if you are using an old version of the UMN UNIX Gopher client, some features may not be available. If so, you may want to upgrade to the latest version.

### Finding out Where You Are

With Gopher, it is easy to start off locally but then, without realizing it, end up looking at something halfway around the world. Many clients give you the ability to find out where you are at any given moment (see Figure 18.1). In the UMN UNIX Gopher client, this feature is invoked by typing the equal sign (=).

#### [Figure 18.1. The UMN UNIX Gopher display of where you are.](#)

In this display, the lines starting with # are comments and unimportant to Gopher itself. The Name= line is the name of the item on the current menu. The Path= is the path used by the Gopher protocol. The Host= line is the hostname that the server is running on. Port= is the port number the server is running on. Finally, the URL line is Uniform Resource Locator (URL), which describes how to find this item on the Internet in a compact form. URLs are discussed in detail in Chapter 10, "World Wide Web." Feel free to jump ahead and read it; later sections of this chapter use some URLs.

### Bookmarks

Frequently when using Gopher, you are somewhere that you want to come back to later. Of course, you could just write down the information about where you are and then tell Gopher to go there again. However, it is much simpler to do this using a feature called a *bookmark*. A bookmark is exactly the same as a paper bookmark.

Using the UMN UNIX Gopher client, you highlight the item you want to make a bookmark for and then type a. You will be requested to enter a name for the bookmark. Then if you later type v, a menu of bookmarks will appear, and you can select which bookmark you want and go right back to that item.

If you are a heavy user of the bookmark feature, from time to time you may want to clean up some old bookmarks that you no longer need. In the UMN UNIX Gopher client, this is easily accomplished by typing d while looking at the bookmark you want to delete.

### AFTP

In some of the recent releases of Gopher 2.X, a feature known as AFTP is present. AFTP stands for Anonymous FTP. It basically enables you to use your Gopher client as an FTP client instead of using FTP directly. In many cases, using Gopher is much easier and faster than using FTP directly.

To use AFTP, just type f in the UMN UNIX Gopher client. You will be prompted for an anonymous FTP node

name, and then (optionally) a selector string. This selector is the path you want to see on the anonymous FTP server. If you do not specify one, you will see the top-level directory of the FTP server.

The major disadvantage to this is that it is resource-intensive and does add some load to your Gopher server. This is because AFTP uses your home Gopher server to handle the FTP requests. Also, Gopher is unable to handle some FTP sites that run unusual versions of the FTP server.

## Configuring Your Client

Your Gopher client most likely has quite a few useful configuration options. One of the most important is the ability to define external viewers to use. The Gopher philosophy is to keep the client software as simple as possible. To accomplish this goal, Gopher needs to call other applications when performing operations such as display graphics, looking at a WWW document, and connecting to remote sites for login.

Also, bookmarks are actually part of your client configuration too. On UNIX, all of these configurations (including the bookmarks) can be found in a file called `.gopherrc`. An example `.gopherrc` can be seen in Listing 18.1.

### Listing 18.1. A sample `.gopherrc` file.

```
RCversion: 1.1

map: Text,more %s,lpr %s

map: Text/plain,more %s,lpr %s

map: Audio/basic,play -v 40 -,

map: Image,xv %s &,lpr %s

map: Terminal/telnet,telnet %s,

map: Terminal/tn3270,tn3270 %s,

map: text/html,lynx %s,

map: Image/GIF,xv %s,

map: Image/PCX,xv %s,

SearchBolding: no

bookmarks:

#

Type=7

Name=Search for Files
```

```
Path=7  
Host=rodent.cis.umn.edu  
Port=11111  
#  
Type=1  
Name=Electronic Journals (via CICNet)  
Path=1/e-serials  
Host=gopher.cic.net  
Port=70
```

The RCVersion line is used to specify what format this RC file is in. The map lines specify MIME context types which we discussed while talking about attribute blocks and how to view and print that item type. The context types are generated on the Gopher server based on the file extension. The %s in the viewer or printer command section includes the filename of the item automatically into the command. The SearchBolding option controls whether or not the text found by a search is bolded. Finally, the last few lines are bookmarks to items in Gopherspace.

In the UMN UNIX Gopher client, all of these configuration parameters can be set by using the Bookmarks feature and the O command. Other Gopher clients have similar features.

An important part of correctly viewing different types of items is to install a good set of external viewers. Look in Chapter 10 for the location of many useful viewers. Others are available, but they will require you to do some research to decide what you want and which product to select.

## Server Tricks

Although Gopher servers are easy to install, some of them also contain quite a bit of powerful functionality that can be exploited only by a guru. In addition, running a server often means using some external utilities to accomplish some tasks that the server can not perform on its own.

### Features of Different Servers

The most common Gopher server in existence is the UMN UNIX Gopher server, which runs on UNIX or VMS. It is fully featured and tracks the leading edge of the Gopher world, including Gopher+. It is highly configurable but easy to install, and it has excellent performance. Some new versions, however, tend to have occasional bugs. The main negative of this package is that version 2.X is not free to commercial companies. Version 1.13 is freely available, but it does not support Gopher+ and has at least one serious bug (and several minor ones). The current version can be found as the following:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/gopher2.016.tar.Z>.

The VMN Gopher server's main competitor is GN, by John Franks. GN was written in response to the

commercialization of Gopher 2.X and therefore is freely available to everyone. GN does not, however, support Gopher+ or many other advanced features. On the other hand, it can also act as a WWW server—a useful feature if you want to run only one server to handle both your Gopher and WWW needs. It can be acquired from the following directory:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/gn>

UMN has also written a Gopher server for the Macintosh. It is known as GopherSurfer. GopherSurfer is good for low-end Gopher server needs. It tracks the latest Gopher features fairly closely. However, it is not freely available for commercial companies. GopherSurfer is available from the following:

[ftp://boombox.micro.umn.edu/pub/gopher/Mac\\_server](ftp://boombox.micro.umn.edu/pub/gopher/Mac_server)

There are two major PC Gopher servers. One is GO4HAM for MS-Windows. It requires a TCP/IP package that supports WINSOCK.DLL. GO4HAM is available from:

<ftp://ftp.informatik.uni-hamburg.de/pub/net/gopher/pc/go4ham>

The other Gopher is based on the KA9Q software, which is in the public domain and runs under MS-DOS. It is available from:

[ftp://boombox.micro.umn.edu/pub/gopher/PC\\_server/ka9q](ftp://boombox.micro.umn.edu/pub/gopher/PC_server/ka9q)

Many servers for other platforms exist too. Several of them can be found with a little bit of searching in

<ftp://boombox.micro.umn.edu/pub/gopher>

## Mirroring Servers

It is often useful to make a copy of a menu from a remote Gopher server. The major benefits include increased speed, increased reliability, and less load on the remote Gopher server. At least two different scripts for doing this exist: GopherClone and MetaClone. With either package, you change to the appropriate directory on your server. Then you enter the name of the script, followed by the URL. It will then make a Gopher menu on the remote server starting at the URL and recursively copying submenus. Both GopherClone and MetaClone are available in the following directory:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools>

## Forms

Forms supported by the Gopher server require using the ASK block facility. On the UMN UNIX server, you need to create a file with a .ask extension. This file will contain the form itself defined in the Gopher ASK block language. For example, a basic form may look like:

Note: Please enter your last name

Ask: Last Name:

This example form requests your last name. After getting the last name, the server looks for a script file that will have the same filename but without the .ask extension. This script file will then get the values entered by the user via standard input. The script needs to carefully check to make sure that the input will not cause a security problem and then perform the needed processing. The needed processing can be just about anything, from sending an e-mail message to entering data into a database to giving the user requested information.

The client sees an ASK block by requesting the ASK attribute block by the Gopher+ protocol. To use Gopher

forms, therefore, a Gopher+ server and client must be used. A later section, "Forms Gateway," discusses a forms gateway that will get around this requirement.

## Shepherd

Some Gopher administrators have problems with their Gopher server dying every once in a while. It can be very embarrassing; the users tend to see that it is down before the administrator does. Fortunately, a piece of software named Shepherd exists and can be acquired from the following:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools/shepherd1.0.tar>

Shepherd detects that the Gopher server is down and automatically restarts it. In fact, Shepherd can do the same for HTTP (HyperText Transfer Protocol) and other such servers.

## Subject Menus

At one time or another, many Gopher gurus have decided to organize Gopherspace into subject-oriented menus. As has been repeatedly proven, Gopherspace is too large and grows too fast for any person working part-time to even begin to stand a chance to do it. Even someone working full-time on it for pay would have a hard time doing it.

However, the guru still has a couple of options. One is to search for subject-oriented lists created by other people and provide links to as many of these as possible. The downside to this approach is that the users have to search multiple subject directories to find information on one topic.

Another alternative is a program called LinkMerge. LinkMerge downloads the menus from various places that have menus on the same topic. Then LinkMerge merges the menus together to form one menu. This gets over the downside of the previous approach. To use LinkMerge effectively, the Gopher administrator needs to develop a list of other Gophers strong in particular topics (although a decent list is included with the LinkMerge package). Then on a periodic basis, the administrator needs to run LinkMerge, preferably late at night, to update the menus because, as you know, Gopherspace changes frequently. LinkMerge is available as:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools/linkmerge0.1.shar>

## Gateways

In addition to accessing the data types that are natively supported in Gopher, Gopher provides some mechanisms to access other data types such as other Internet services, databases, and even Usenet News. These mechanisms are typically called *gateways*. Gopher is flexible and offers many different ways for these gateways to be implemented and interface with Gopher.

One of the older approaches is to implement a server that talks the Gopher protocol but provides the needed service. More recently, the go4gw server was designed. The go4gw server sits on an IP port and talks the Gopher protocol (recent versions even talk the Gopher+ protocol). The go4gw server has an API (Application Programming Interface) that can be used by programs. Several people have written interface modules that use the API to tie various services into Gopher.

If you are interested in writing your own interface module, the best way to learn is to look at some of the sample Perl programs included with the go4gw software. Speaking of which, the software is available from the following:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/go4gw2.02.tar.Z>

## Archie

Included with the go4gw package is a Gopher gateway to Archie. In many ways, the Archie-Gopher gateway is superior to using an Archie client. The largest benefit is that this gateway first checks the load average on the various Archie servers. Then it issues the query to the Archie server with the lightest load, which usually will give you the quickest response.

Also, this gateway returns the items in a nice Gopher menu. You can then retrieve an item by just selecting the item from the menu instead of having to look at the Archie results and then issue the appropriate FTP commands.

It should be noted, however, that this does not make Brendan Kehoe's popular Archie C client obsolete. The C client has a few extra features that a guru may use (priorities, for example). However, in all honesty, the key reason it is not obsolete is that the Archie-Gopher gateway depends on it being installed on the server for the gateway to work. You can acquire the Archie-Gopher gateway from the following:

<ftp://ftp.cs.widener.edu/pub/archie.tar.Z>

## NetFind

Also included with the go4gw gateway package is a gateway to NetFind. NetFind enables you to find people's e-mail addresses on the Internet. If you have used the UNIX NetFind client, the only advantage you will find with using the NetFind-Gopher gateway is that it is a little prettier. To use this gateway, you will need to acquire the NetFind package from the following:

<ftp://ftp.cs.colorado.edu/pub/cs/distrib/netfind/netfind4.6.tar.Z>

Also, you will need to regularly download the updates of the NetFind data from this address:

<ftp://ftp.cs.colorado.edu/pub/cs/distrib/netfind/seeddb.tar.Z>

## Whois

Whois is another gateway that is included with the go4gw software. This gateway provides an easy-to-use mechanism to query the various Whois databases, such as the InterNIC's.

## Webster

NeXT machines came with Webster Dictionary servers automatically. It was an obvious step for Gopher to include support for them. This gateway is also part of the go4gw package. Due to licensing reasons, however, you can legally use Webster Dictionary servers only at your own site. It is generally a violation of copyright law to use a server belonging to another site.

## FTP (Hack)

Previous sections discuss AFTP as a method for using FTP within Gopher. AFTP is fairly recent and available only in the UMN UNIX software. Previous to that, the g2ftphack gateway was used for a similar function. The g2ftphack gateway is included with the go4gw package. It works when the user types in the FTP server he or she wishes to connect to. The gateway then FTPs to there and presents the user with the files on the FTP as a menu. It is still useful for users of clients that do not have the AFTP feature built in.

## News

Several different Gopher to Usenet News gateways exist. Included with the go4gw package is g2nntp. This particular package allows whole newsgroups or just individual articles to be listed. Another one of the packages is gonntp (from Louisiana State University), which is available at the following address:

<ftp://boombox.micro.umn.edu/pub/gopher/incoming/gonntp-exec-LaTech.tar.Z>

gonntp has been modified to use INN's (InterNetNews) XHDR command. This version is called gonnrp and is quite a bit faster than the original gonntp. It can be found at the following:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/gopher-gateways/gonnrp>

All of the gateways treat each Usenet News article as a Gopher item. Many users want to use the Gopher purpose newsreader, but it is not good at that. The real purpose of these gateways is to include some information from Usenet News, along with other information on the same topic.

## **X.500**

X.500 is a complex directory service system that fits the whole world into one massive heirarchical framework. The Gopher-X.500 gateway software is known as go500gw. It makes X.500 look like Gopher menus, items, and searches. Unfortunately, this gateway is, at times, a little slow. The software is available from the following address:

<ftp://mojo.ots.utexas.edu/pub/src/go500gw.c>

## **Databases**

SQL (Structured Query Language) databases such as Sybase and Oracle can be interfaced with Gopher through a gateway. The SQL gateway can turn Gopher commands into SQL commands. Some of the SQL functionality is lost in the process, but much of it still exists. The gateway can be acquired from the following:

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/gopher-gateways/gophersql>

## **Z39.50**

Z39.50 is a protocol used by the library community to exchange bibliographic records. It will be an increasingly important facility on the Internet over the next few years. Right now, almost all library users have to telnet to various libraries, login, and then use that site's user interface. It requires a lot of effort on the part of the user to learn all these different user interfaces. Instead of that, the user can use a Z39.50 client program, which will talk to the various libraries. The user needs to know only one user interface.

Gopher has one of the first usable gateways to Z39.50. It is available from

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/gopher-gateways/go4zgate>

The installation takes some time and is a little difficult. Also, you must be running Gopher+ on both the server and the client to use the Z39.50 gateway, because it uses fill-out forms.

Also, it should be noted that the Z39.50 protocol is still in the process of enhancement, so you should periodically check for new versions of the gateway that support some of the Z39.50 features coming along. One of the most important features that should be coming down the pike is holding information so you can see if the library actually has the book you are looking for on the shelves.

## **Other go4gw Gateways**

go4gw has a few more gateways included that we have not discussed, all of which seem to be used infrequently. A gateway to an areacode server is available. The geography gateway enables you to get information on cities, latitudes, longitudes, populations, evaluations, and other such information. In addition, there is a gateway to SNMP (Simple Network Management Protocol). The SNMP gateway is, in general, useful only to network administrators, because it is a system for monitoring and managing networks.

Finally, go4gw is extendable. Therefore, if you need a gateway that does not exist and you know Perl, it is easy to add one to go4gw.

## **Gmail**

Gmail is very different than any other gateway discussed so far. All these other gateways are for the user viewing information external to Gopher. In contrast, Gmail is a gateway that allows people to post documents into a UNIX Gopher server via an e-mail message. This is useful for people who are on a different platform from your Gopher server and do not know how to use UNIX. Also, it is often the easiest way to post an item if you are on a different UNIX machine, because it will place and set up the item correctly automatically for you.

The Gmail administrator installs Gmail and then sets up a security file. In this file, the administrator can control who can post to which Gopher menu. The security checking is weak and based on e-mail return addresses, but it does have a fallback line of defense. If someone does forge a message that gets posted, Gmail always sends a copy of any posted message automatically to the person who it is supposedly from and, depending on the configuration, others as well. On a forged e-mail message, this does not go to the forger but to the person who is allowed to post, and that person will know that they did not send the message. Therefore, the hacker really has no chance for his work to go undetected.

Along with Gmail is an associated program called Gcal. Gcal is for managing a Gopher-based, calendar-of-events system via e-mail. Gcal enables multiple people to add items into the calendar of events also. Gmail can be acquired from

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools/gmail1.01.shar>

A couple of newer, stable beta releases with major improvements have come out since 1.01, which is what is listed here. If you can not find a later release on boombox and you want it, you will need to ask Prentiss Riddle, the primary author of Gmail, by sending him e-mail at [riddle@is.rice.edu](mailto:riddle@is.rice.edu).

## **Forms Gateway**

Previously, it was stated that Gopher Forms required the ASK block facility and Gopher+. This is true if you want to use the officially supported way of doing forms in Gopher. Well, of course, there is another way to do forms through a gateway. This is useful for people who are not running Gopher+ servers.

The forms gateway package is called GoForm and is available as

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools>

GoForm works by having the user telnet to a specific port. On this port is a GoForm server, which is the form. Each form requires its own port and that GoForm server be running. This may sound painful, but it works quite a bit better than it sounds. Also, in any case, it is really your only clean option of doing forms in Gopher without having to use Gopher+.

## Security Issues

While the Gopher protocol and system do not lend themselves to security problems, some of the implementations of Gopher have had some security holes. Also, even with software that has no security holes, bad security practices on the part of the Gopher administrator can lead to security problems.

An important good practice is to keep track of current security discussions. Security-related discussions involving Gopher may occur on `comp.security.announce`, `cert-request@cert.org`, or `gopher-announce@boombox.micro.umn.edu`.

### Insecure Versions and Practices

The only Gopher clients and servers with known security problems are the UMN UNIX Gopher clients and servers. Any version before 1.13, and any version between 2.0 and 2.012 inclusive, has known and serious security problems. These version numbers apply to both the client and the server code.

However, it is generally in your best interest to use the latest version of these programs whenever possible. The authors are currently in the process of rewriting significant portions of the code to greatly reduce the possibility of a security hole being present. The changes mainly involve screening user input going to the client and information from the server that could cause a security problem on the client end.

The largest remaining security problem with the client side of Gopher, as well as with any other information retrieval system, is that some document types are insecure by their nature and should be viewed only with great care. One example is PostScript. It is possible to imbed file system commands in a PostScript document. If you retrieve such a PostScript document with Gopher to view it, Gopher will pass the document to your PostScript interpreter. At that point, it is up to your PostScript viewer whether or not to process these commands. If it does, you could lose files—or worse. On the positive side, to my knowledge this has never been exploited, but that does not mean it will not happen sometime in the future.

The server has a much greater potential for security mistakes being made. First, the Gopher server should never be run as root or any other account that has special system privileges. Also, you need to be sure that any shell or Perl scripts you write and put on your Gopher server are secure. If they are Perl scripts, you may want to run them using Taintperl instead of normal Perl. Taintperl makes sure that you never pass user input to a shell without really meaning to. These kinds of programming problems have been the most frequent cause of security problems in the Gopher system to date.

The standard way to install the Gopher server is so that it runs under a chroot environment. The basic concept here is that the server can see only a tiny fraction of the files on the server machine. These files are the only ones it needs to operate. Many other files that could cause serious security problems for your server if viewed or changed will not be in the chroot environment to cause problems. However, the chroot environment is difficult to work with in many ways. Gopher lets you run in a normal (non-chroot) environment also. This option should be used only with extreme care and only when absolutely necessary because it greatly increases your potential security risks.

Also, many of the Gopher gateways have suffered at various times from security problems. Older versions of the Veronica server and go4zgate, for example, had some serious security problems. Both of those have been fixed, but others may not have.



**Note:** There aren't really version numbers that signify fixed vs. unfixed. In both cases, there is a version

that may or may not have the hole.

---

Another good practice is to always log your Gopher transactions, which will enable you to have an audit trail if problems occur. In addition, this is a useful tool in debugging some misconfigurations of your or other Gopher servers.

## Controlling Access to Your Server

In the `gopherd.conf` file on the UMN UNIX Gopher server are some `access:` lines that specify what access the Gopher server allows to specific hosts and/or domains. Some example `access:` lines look like the following:

```
access: default !read,!browse,!search,!ftp
```

```
access: .utdallas.edu read,browse,search,ftp
```

```
access: .metronet.com read,browse,search,!ftp
```

```
access: .unt.edu !read,browse,!search,!ftp
```

In this example, by default, access on the Gopher server is granted to nobody. The default rule applies to anyone who is not covered by a specific rule. Next, people who are on machines in the `.utdallas.edu` domain are given have full access to the Gopher server.

Users in the `.metronet.com` domain have full access, except for FTP. I have found this to be an extremely useful feature. The Gopher server acts as an FTP gateway for Gopher clients. From my experience, the impact of this on the server can be very negative. The FTP gateway is resource-intensive and has been prone to bugs. In almost all cases, you will want to allow local users to use the gateway, but as far as remote users go, it needs some careful thought.

Finally, a neighboring university (the University of North Texas, or `unt`) is given browse access only to the server. The effect is that UNT users see the menus on the server, but they do not have access to read any item on the Gopher server. This practice is known as teasing and at GopherCon 94 was recognized as a bad practice to use. It is annoying to the users of the Gopher system.

Search is the last option. It controls whether or not the user can access any search item or WAIS database that resides on the server.

## Public Access Clients

Some Gopher administrators want to run public access Gopher clients so that they do not have to assign accounts to all users who need to run Gopher. This can be a huge security hole if it is not done absolutely correctly.

First, you need to install special versions of `tn3270`, `telnet`, `more`, and other viewers that have been checked for security holes and do not allow access to a shell. Many of these can be found at

<ftp://boombox.micro.umn.edu/pub/gopher/Unix/GopherTools/securegopher.tar.Z>

It is critical to make sure that only these can be run by the public access client. Otherwise, you are effectively giving everybody access to a UNIX prompt on your public access machine.

Another important step is to run the Gopher client with the `-s` flag. This flag disables many Gopher features

such as printing, saving files, and choosing viewers.

The public-access Gopher client needs to be in its own account, which should not be used for anything else. In addition, it should not run a shell of any type or call the Gopher startup script or program that you want to use immediately upon login. Also, the account should be configured so that the user is logged out immediately upon exiting Gopher.

## AdmitOne Authentication

From its beginnings, Gopher was developed based on the concept of open access to information. IP-address-based security options were soon added, because it is useful to restrict access to certain types of information. However, Gopher still did not have a way to identify a user on a multiuser host machine or handle a user who moved around to different machines at different times.

Because the Gopher server is stateless, this made things more difficult. Obviously, a system that asks you for a password on every transaction is too painful to use. For this reason, UMN created the AdmitOne authentication schemes. When using AdmitOne, the client and the server run through the password transaction once and then use authenticated tickets to keep the user authenticated with the server for the remaining transactions.

This section does not delve into the depths of the authentication system. Some security experts have criticized AdmitOne as being *light-weight authentication*; it does make it harder for hackers to break in and steal the data they want, but a determined effort will succeed. The critics say that if you go to the trouble of implementing a light-weight security scheme, you might as well use a heavy-weight system such as Kerberos instead. The defenders of AdmitOne say that light-weight has its place by being easy to implement and protecting some sensitive but noncritical data; the amount of effort to break in is greater than the value of the data itself. You can make your own decision.

## Future Developments

Many WWW supporters feel that Gopher is useless and needs to be replaced by the Web. However, as many users know, the Web requires too much hardware horsepower to be a solution for everyone. To effectively use the Web, you need a fast CPU, VGA-resolution or higher graphics, and a fast Internet connection (faster than 14.4K). Meanwhile, Gopher was designed to work on very low-end machines with 2400-baud SLIP connections. Over the next few years, Gopher will continue to have a place just because the Web is too resource-intensive (though the Web will probably eventually supercede Gopher's role).

Some Gopher supporters believe that the best future option is not the unstructured chaos of hypertext documents currently on the Web. They feel that Gopher's role is to provide a table of contents for the world of Internet information.

## 3-D Gopher

People are now working on 3-D interfaces to Gopherspace. It is said that the CPU horsepower now available with Pentium and PowerPC chips means that it can be done effectively. The implementation plan is that the Gopher+ protocol would remain the same; only minimal data would be sent across the network.

All the graphical displays would be calculated and shown by the client. Therefore, it would be possible for different clients to have different graphical interfaces. In the current prototype, the item types are shown via shapes. All kinds of interesting ideas, like beaten-down paths for highly-used items, are being discussed. Some of us who suffer from motion sickness have our doubts as to the benefits of a 3-D interface; it may be bad for

health reasons.

Another part of the eventual 3-D interface will be interactive communication facilities, so that users of the Gopher system will be able to interact with each other while looking for information. The many useful applications for this facility include the ability of one person to show another person where information is in Gopherspace.

## Archie

In addition to indexing FTP sites, Archie will soon be indexing Gopher sites too. The key difference between Archie and Veronica will be that Veronica indexes everything unless specifically told not to. Archie will index only what it is told to index.





- [Part V](#)
  - [Tricks of the Professional Internet Gurus](#)

## Part V

# Tricks of the Professional Internet Gurus





- [19](#)
  - [Internet in The Classroom: How Educators Use the Internet](#)
  - [--by Dave Kinnaman](#)
    - [Introduction](#)
    - [Why Educators Have High Hopes for the Internet](#)
      - [How the Internet Changes Learning](#)
      - [Both Time and Space Seem to Disappear](#)
      - [Self-Reliant Learning](#)
      - [The Net is Color-Blind and Does Not Discriminate](#)
    - [Janet Murray](#)
      - [—by Tod Foley](#)
      - ["Nancy, Tell Mommy What You Found on the Internet Today!"](#)
    - [The Educator's Netiquette](#)
    - [Common E-Mail Boo-Boos](#)
    - [A Pedagogue's Internet Tool Kit](#)
      - [Emergency Procedures](#)
      - [Whistle While You Wait: Remote Access at a Snail's Pace](#)
      - [Giving a Courteous Farewell](#)
      - [Connect to the Planetary Clock](#)
    - [WAIS: Wide Area Information Servers](#)
      - [Single Source WAIS Searches from Your Local Gopher](#)
      - [Searching Many Databases at One Time](#)
      - [WAIS Indexes](#)
      - [Performing a WAIS Search](#)
    - [Online Information: Current Newspapers and Magazines](#)
      - [Electronic Newsstand](#)
      - [Online Magazines: Uncover by CARL](#)
      - [Electronic Text, Newspapers, and Magazines: American Cybercasting](#)
    - [Online Coursework](#)
      - [Online Instructional Methods](#)
      - [Online Learning Resources](#)
        - [The Georgia Institute of Technology](#)
        - [Kansas State University](#)
        - [The National Technological University](#)
        - [New York University](#)
        - [Purdue University](#)
        - [The Rochester Institute of Technology](#)
        - [Empire State College](#)

- [Syracuse University](#)
- [The University of Alaska at Fairbanks](#)
- [The University of Arizona](#)
- [The University of Colorado at Colorado Springs](#)
- [The University of Idaho](#)
- [The University of South Carolina](#)
- [The University of Washington School of Medicine](#)
- [Electronic University Network \(EUN\): America Online](#)
- [Schools Using the Net](#)
  - [Grand River Elementary School: Lansing, Michigan](#)
  - [Claremont High School: Claremont, California](#)
- [Online Database Access](#)
  - [CD Plus Technologies \(formerly BRS Information Technologies\)](#)
  - [DIALOG Information Services and Data-Star](#)
  - [GTE Network Services](#)
  - [Online Computer Library Center \(OCLC\)](#)
- [Online Research Services](#)
  - [DialSearch](#)
  - [Knowledge One](#)
- [Communicating with Teachers](#)
  - [Educator's Guide to E-Mail Lists](#)
  - [Ednet Guide to Usenet Newsgroups](#)
    - [Usenet Newsgroups](#)
    - [Commercial Newsgroups \(\\$\)](#)
- [Searching Listserv Archives](#)
  - [Submitting a Search](#)
    - [The search Command](#)
    - [The index Command](#)
    - [The print Command](#)
  - [Using InfoMagnet to Search Listserv Archives](#)
  - [Personal \(Filtered\) News](#)
    - [HeadsUp: INDIVIDUAL, Inc.](#)
    - [Netnews Filtering Service](#)
- [Public Education and the Internet](#)
  - [When Should the Internet Enter the Classroom?](#)
  - [Forms of Education on the Internet](#)
  - [The Internet's Educational Potential](#)

# 19

## Internet in The Classroom: How Educators Use the Internet

[--by Dave Kinnaman](#)

When the student is ready, the master will appear.

*--ancient mystical epigram*

### **Introduction**

*This chapter presents you with new ideas about education and about the Internet. Much of education faces imminent reform, and the Net could be the best thing that has happened to public education this century. Provided that access is affordable, fairly provided, and universally accessible.*

*This chapter will help you sharpen your Internet skills, provide you with some new skills, show you how to use powerful search programs to find educationally relevant materials, demonstrate that education is moving onto the Net in many academic disciplines, and help you find ways to use the Net to enrich and perhaps redesign your curriculum.*

*So, here's what's ahead:*

*Why educators have high hopes for the Internet*

*The educator's netiquette*

*A pedagogue's tool kit*

*WAIS: Wide Area Information Servers*

*Online information: current newspapers and magazines*

*Online coursework*

*Communicating with teachers*

*Searching Listserv archives*

*Public education and the Internet*

### **Why Educators Have High Hopes for the Internet**

*Everyone seems to be talking about the Net, the "information super-highway," the infobahn, the National Information Infrastructure (NII), and the Internet. Why is this so?*

*It seems that we see this new technology, the Internet, as a ray of sunlight, the first light of a new Information Age. It does hold great promise for our whole society, especially for educators and also for learners or students.*

*Teachers are asking "Why should I use the Internet in my home or classroom? Is this a good way to help children and adults prepare for a better future?" This next section is intended to begin to introduce to you the many ways that education is already changing because of the Internet.*

### **How the Internet Changes Learning**

Let's face facts—the Internet is exciting. Learning while using the Net is a powerful releaser of emotion, a motivator, and a new form of engagement for many students. Students gravitate to the Net like nothing before in their lives.

While the Net is not likely to replace after-school athletics or music lessons very soon, for some students it is the best learning opportunity ever made available to them. These children soon realize that their rewards on the Internet will be in proportion to the effort they invest in learning Net skills, and learn all the more for it.

Contacts in far away places around the world make any project more dynamic, and more interesting. Frequently, the Internet has the most up-to-date information available anywhere. Both teachers and students are energized by the freshness and immediacy of the Net.



**Tip:** This section cannot be a comprehensive review of all the reasons why every school should be using the Internet. Instead, it will briefly list in broad strokes how the Net is being used to accomplish important educational goals—in some cases, goals that *cannot be achieved* without Internet access.

The Net *is* the future. It costs less than a high school sports program, and Internet competence will stay with students for life, because these skills are intellectual, not physical. Keyboarding is only the beginning. Kids who are connected to the Net learn to ask better questions, to make better arguments, and to present themselves more positively, by using the Internet as a new form of communication.

Let's start with e-mail. Children concentrate harder to express themselves when sending an e-mail message to another country or state. They work on vocabulary and clarity as never before. When they know another student in Stockholm, or an engineer at NASA, will be writing back, children recognize the difference between slang and formal language. Some evidence has been found that kids express themselves more carefully and demonstrate better communication skills while using e-mail than they do in written tests for their regular classroom teacher!

On the topic of formal language—learning computer languages, that is—learning to *program* computers is a tremendous mathematical learning opportunity. Computer operation has its essence in math, and truly understanding computers requires some fairly sophisticated mathematical and scientific concepts. Kids who become ensnared in the Net tend to "automatically" excel in other areas of math, science, and often surprise their teachers and parents with new knowledge far afield from math and science.

The Internet is rapidly becoming the largest reservoir of knowledge ever extant on this planet. Wise students grow adept at finding and retrieving remote Net information. They then go on to develop more sophisticated search and retrieval strategies. Soon, they begin to really admire librarians for their access and retrieval skills, no longer as magical *sources* of books and information.

Once students have gathered together masses of new information, their teachers are expected to know how to help sort out the information and deal with it constructively. This is the beginning, for students, of learning the arts of analysis, evaluation, and application. These writing, thinking, and knowledge skills learned on the Internet are applicable to almost every curriculum area.

Also, don't forget that the Net's very best resources are human beings, not computers or databases! Old friends, new friends, and experts of all types—colleagues, specialists, and fellows of all stripes are on the Net, anticipating your arrival. Teacher, parent, and student isolation can become a thing of the past!

## **Both Time and Space Seem to Disappear**

The normal limitations of time and space do not apply to the Net. New Zealand is just as close as your state capital, in Net time. E-mail arrives in minutes, and huge files are copied and transferred thousands of miles in mere seconds.

Teachers use e-mail to do consultations, confident they won't have to play telephone-tag. Students can exchange several volleys with keypals before a surface mail letter can make a single *one-way* trip.

Sending mail on a Listserv to hundreds or thousands of people is no more work than sending a single message—the Net does the extra toil for you!

## **Self-Reliant Learning**

Because it is like a worldwide personal library, the Net inspires students, teachers, and parents to find and use new

information.

Powerful Internet tools allow everyone to find and retrieve new information, data, graphic images, and software that is personally interesting, almost instantly. Often the information is simply not available except on the Internet.

Students also learn the importance of educational independence and intellectual autonomy from using the Net for school work.

The Net is more current (as in *instantaneous*) and more dynamic than any old-fashioned library could be, so students and teachers learn that using the Net can save them time. Because the Internet exhibits many of the signs of life and is so responsive to our information needs, some people even become attached to the Net, as they would to a dear friend.

### **The Net is Color-Blind and Does Not Discriminate**

Socially induced prejudices like appearance, gender, race, age, and behaviors fade away in the world of the Internet, because students learn that they are judged solely on what they say and how they say it.

Hearing-impaired people may actually have an *advantage* on the Net, because they are not distracted by outside noises.

Class, race, ability, and disability are removed from consideration in Internet communication. They simply do not matter.

## **Janet Murray**

—by **Tod Foley**

Margaret Honey surveyed educators who were using telecommunications for professional and instructional activities in the spring of 1992. Respondents were described as highly self-motivated, with little support from the school or district level. A high percentage of the teachers reported that they were self-taught, learning from their online colleagues, and a substantial majority are "conducting professional networking activities from their own homes, suggesting that much of their telecommunications work is now done on their own time, at their own expense, and with a high level of personal commitment." A reasonable conclusion is that this significant national movement, which only recently has gained the attention of the popular media, has sprung from grassroots activities fostered and sustained by dedicated professionals who see this movement as extraordinarily beneficial to themselves and their students.

—Janet Murray, "Schoolkids and the Net," from *The Internet Unleashed* (Indianapolis: Sams Publishing, 1994, pp. 874-75).

Janet Murray is the librarian at a comprehensive public high school serving 1,500 students in Portland, Oregon, and SysOp of the HI TECH TOOLS for Librarians BBS (FidoNet 1:105/23). She is also a cofounder of K12Net, a virtual network of 37 educationally-oriented conferences devoted to curriculum, language exchanges with native speakers, and classroom-to-classroom projects. Since obtaining her MALS from Rosary College Graduate School of Library and Information Science in 1981, she has been an active advocate of online programs, making appearances and speeches at national conferences, automating circulation systems in school libraries, and bringing innovative access systems to her school district. In the following e-mail interview, she answers questions about K12Net and the future of the electronic school.

**TF:** Over the last few years, the word "education" has become a sort of rallying cry for the technologically progressive; from Internet-savvy librarians <g> to Hollywood "Edutainment" producers and Silicon Valley software developers. Obviously, digital storage/retrieval and advanced telecommunications are ideal for educational applications, but the bottom-line question is: "Who pays for this stuff?" Has this general interest translated into more financial support from the private and commercial sectors?

**JM:** Not in my experience. But the government's recent initiatives in funding tend to encourage local collaboration and matching funds from the private and commercial sectors, so that may be an emerging trend.

**TF:** There are many who feel that the increasing privatization of the educational industry will lead to an eventual displacement of the public school system. Do you feel this is a real possibility? And is that good news or bad news?

**JM:** First, let me quibble over a semantic issue: Do you (and others) regard education as an industry? A recent issue of the *Phi Delta Kappan* featured the privatization movement with a cover caricaturizing children as "products" emerging from an assembly line... But to respond to your first question—No. Commerce is rarely altruistic. Corporations that provide education for a fee (and a profit) will attract those who can afford it to the ultimate detriment of those who cannot. I'm highly skeptical of

the promises made by those with a vested interest in privatizing education; if there were a panacea to the perceived problems of education, it would have been discovered by now. As for your second question—the U.S. Supreme Court rejected the "separate but equal" doctrine forty years ago; any educational reform movement that fosters inequality is bad news. The U.S. Postal Service may provide a useful analogy; its privatization increased competition as well as costs. Now those who can afford Federal Express or United Parcel Service may prefer those services, but the national public system has not been displaced, and few would assert that it offers "equal" services.

**TF:** The Internet is regarded by many as a sort of "panacea," not only for purposes of education but for all of society as well. Of course this picture may be changing; the NSF's upcoming withdrawal from management of the Internet backbone and the increasing "privatization of cyberspace," when coupled with the traditional underfunding of the public school system, seems to imply that educators must find new ways to utilize and cross-link existing systems. Do you see this happening in the field?

**JM:** In 1994, there have been several initiatives to promote cooperation among federal and state agencies, network service providers, and educational practitioners. The March 1994 report, "Building Consensus/Building Models: A Networking Strategy for Change," summarizes the recommendations derived from one such collaboration.

**TF:** How can public education make best use of the Internet (or the NII), and yet remain unbiased and egalitarian? *That* is the question—.

**JM:** The Internet/NII's capacity to offer small, isolated schools access to the same resources available in urban school districts is fundamental to questions of equity. Although 67 percent of U.S. elementary schools and 50 percent of our high schools serve fewer than 500 students, all of the currently funded NIE testbed sites are located in metropolitan areas. These smaller schools are typically least able to afford the high-speed networking configurations currently being recommended by the experts, and, paradoxically, most likely to benefit from access. We must promote networking solutions that will serve *all* schools, not just the elite few.

**TF:** This leads to K12Net, of which you are a cofounder. K12Net, which coexists on FidoNet-compatible BBSs and as a series of Usenet newsgroups in the k12.\* hierarchy, is demonstrating the capabilities afforded us *right now*—the cross-linking of networking systems for the common purpose. Since access requires only a working Fido BBS, students, teachers, and administrators from all over the world can make use of K12Net, even if they're located far from the nearest university or Internet access provider. What's going on these days on K12Net?

**JM:** K12Net has always described itself as a "network with training wheels" designed to acquaint K-12 educators and students with the benefits of telecommunications. As more schools obtain access to Internet, our participants will probably shift their focus to sharing information about efficient location of Internet resources. The statewide networks in California, North Dakota, and Virginia already provide their states' primary access point for K12Net through the Usenet newsgroups in the k12.\* hierarchy. However, K12Net's status as an *international* educational network is heavily dependent on its FidoNet roots. Nearly 50 percent of the participating K12Net bulletin board systems are located outside the continental U.S., where K-12 educational access to Internet is highly problematic. Therefore, I do not expect K12Net to rely solely on Internet distribution in the foreseeable future.

**TF:** Are there other differences? Do the Usenet posts tend to have different purposes than those which come in from the Fido systems?

**JM:** Posts from both types of sites focus on the useful exchange of information. An important advantage of crosslinking K12Net bulletin board systems with Usenet newsgroups is that it enables university participation. Student inquiries frequently receive valuable responses from correspondents in research-oriented institutions.

**TF:** In terms of the future of public education and equal access to what might be called "public knowledge bases"—the educational levels of the NII—what are the most important hurdles before us?

**JM:** Affordable access and adequate training are the most significant hurdles to be surmounted by Internet enthusiasts. Rapid developments in the provision of user-friendly interfaces over the past two years have created an environment in which one can successfully locate materials and information appropriate in a K-12 setting. Now we must educate and persuade the service providers and telcos that it is in their best interest to provide affordable access, and we must develop training materials that can be used effectively and independently at remote sites.

**TF:** And how is K12Net addressing these issues?

**JM:** We are promoting awareness of the need for inexpensive, locally manageable networking software and training materials appropriate for use in individual schools. (Commercial solutions developed by BBSs and e-soft are prohibitively expensive and require longterm commitments for technical support.) Several of us are participating in local and statewide initiatives to network K-12 schools.

K12Net is an all-volunteer effort, with participating BBSs located throughout the world. For more information regarding K12Net and its activities, contact Jack Crawford or Janet Murray at these addresses:

**Jack Crawford**

Wayne-Finger Lakes Teacher Resource Center  
703 E. Maple Avenue  
10 Eisenhower Hall  
Newark, NJ 14513-1863  
Phone: (315) 331-1584  
E-mail: jack@rochgte.fidonet.org

**Janet Murray**

Wilson High School  
1151 S.W. Vermont Street  
Portland, OR 97219  
Phone: (503) 280-5280 x450  
E-mail: jmurray@psg.com

For current information on the state of Internet/K-12 activities and plans, the two following IETF documents are of significant value:

Gargano, Joan and David Wasley. "K-12 Internetworking Guidelines." Working Draft, IETF School Networking Group: June, 1994.

Sellers, Jennifer. "Answers to Commonly Asked 'Primary and Secondary School Internet User' Questions." IETF School Networking Group, Internet FYI, RFC 1578: February, 1994.

**"Nancy, Tell Mommy What You Found on the Internet Today!"**

After a youngster has a significant vocabulary and adult grammar, there is little to keep him or her from conversing like an adult over the Net. Likewise, adults can masquerade as kids. This brings up "security" concerns in educators and parents alike.

Safe telecomputing for kids must be taught *before* the keyboard and modem are freely available in an unsupervised environment. However, local policies differ greatly. General guidance is available in RFC 1578 (aka FYI 22). It is available by anonymous FTP.

Access:

Anonymous FTP:

**North America**

site ds.internic.net or 198.49.45.10

directory /fyi

**Europe**

site src.doc.ic.ac.uk

directory /rfc

Unfortunately, there simply is no substitute for infusing issues of information safety, privacy, and ethics into the technology curriculum at early age. All teachers easily recognize safety issues in science classes and around machines and shop equipment. These and related privacy and ethical concerns also must be addressed around Net access and proper Net behavior.

The Gopher menu in Figure 19.1 demonstrates one of the danger zones educator gurus should be prepared for before a crisis

happens. This kind of material is absolutely not common or usual, but the Net has great diversity, and such things are available.

### Figure 19.1. Menus like this are not typical on the Internet, but they are there if one looks for them.

Establishing clear rules, and consequences for breaking them, is a school-wide or district-wide responsibility. Parental involvement and support is a must, especially in the earliest years.

---

**N** **Note:** One of the best ways to get parents involved is to treat using the Internet like it was a field trip. (In a very real sense, it *is* a field trip away from the classroom, into another physical and temporal dimension.) Teachers can send home an announcement and a parental release form, to make parents aware of why it's a good idea to use the Net, and what the "unforeseeable but possible" risks might be. Parents often take signature forms more seriously.

---

In RFC 1578, the Internet Q&A for K-12 schools Jennifer Sellers says "...schools need to exercise reasonable oversight while realizing that it is almost impossible to absolutely guarantee that students will not be able to access objectionable material."

---

**N** **Note:** One of the reasons that censorship and "authoritarian" control of communication on the Internet is difficult to accomplish is that the Net was *designed* to be de-centralized and survival oriented, rather than centrally controlled and obedience oriented.

---

Because the Pentagon required a communication system that would still work *after* a nuclear attack on the USA, the Net doesn't have top-down controls of the kind we have grown to expect in our media-information-consuming society.

*Time* magazine, in a cover story about the Net, said, "It's hard to block anything on the Internet, which interprets censorship as damage and routes around it." *Time*, July 25, 1994, p. 54.

Since it is *almost impossible* to prevent some lapses, it is vital to have clear standards up front, *before* you are faced with emotional administrators, students, and parents!

## The Educator's Netiquette

Even Net gurus have pet peeves. The difference lies in how a true guru responds to these everyday nuisances.

Here is a list of the commonest newbie e-mail and Listserv mistakes made on the Net, given in the order of how much they irk Net Gurus. In each case, the proper, appropriate behavior is indicated, so the mistaken newbie can learn and evolve into a higher Net Being, perhaps even a guru.

### Common E-Mail Boo-Boos

- *Inadvertently posting a **personal** message to an entire Listserv list (hundreds or thousands of people), often by using the Reply: function of a poorly understood mail-reading program.*

*Evolved Response:* Be sure you obtain, study, and get practice with the directions for your mail program and text editor. You are responsible for the results of your use of these programs. Any failure will reflect on *you*. Your readers out in cyberspace have already tamed their mail programs, and they know what you're going through. But they also expect you to keep improving, and not make the same mistakes over and over.

---

**N** **Note:** For practice, first mail a message to yourself, to see what your message looks like. Then, send a close friend a message and yourself a cc: copy of that same message. Slowly work your way up to more and more sophisticated mail processing, one small step at a time. Remember, even the most advanced net gurus always check the To: and cc: lines before

they dispatch a message!

---

- Whenever you use the Reply: function, look carefully at the outgoing address that is automatically filled in. Be sure the address shown is the person you intend to reach. Some mail programs mail your Reply: to everyone imaginable unless you notice it and delete that automatic line.
- *Not including your name and e-mail address at the end of every post.* Some mailer programs do not show the headers of arriving messages, which makes it difficult to reply privately to messages that lack a short "signature." Often this newbie error contributes to others making error #1.

*Evolved Response:* Always put at least your name and e-mail address on every piece of outgoing mail, unless you are absolutely sure all recipients know you personally, and your e-mail address. If you do not include your signature, you may unintentionally miss pertinent information because it's just too inconvenient for others to somehow find your address to reply.

---

**IMPORTANT!**

**Tip:** If you are one of the unlucky people whose mail-reading program does not show incoming message "headers," where all that important addressing and reply information is, here is a way you can trick the computer into telling you the address, anyway:

Save the incoming message in question as a text file. Find the option on your mail program for saving mail as a file. Then, exit the mail program and open the file in a text editor. Viola! All the complex address information was really there all along—your mail program just prefers to hide it from you. Why do you use such an unevolved program?

Ask around to see if there are any better mail programs available to you. PINE is one of the very best, and it is MIME-compliant, which means you can use it to send almost any data file *attached* to your e-mail messages.

---

- *Sending commands meant for the Listserv robot program to all the people on the list instead.* None of the people on the list can help you unsubscribe, subscribe, signoff, postpone, nomail, and so on. These clerical tasks are designed for *privacy*, for you to do them all by yourself. You tell the Listserv robot program what you want and it complies, without anybody else seeing or hearing of it.

*Evolved Response:* Don't send anything to all the people on the list unless it is on the defined list topic.

There are always two different addresses for the people and the robot, and sometimes there is even another address for a human being—the moderator or list owner. Use the correct address.

The address for the clerical robot varies, because there are several different kinds of software and hardware combinations that do the mailing list clerical work. Typically, the robot's address will start with something like

listserv@ listproc@ majordomo@ listname-request@

The address for the actual people on the list *never* starts with the previous examples.

---

**IMPORTANT!**

**Tip:** You can often get the correct address for the human moderator by sending the following message to the Listserv robot.

To: Listserv@host.domain

review listname(-1)

You should get back a message containing most of the addresses of all the other people who subscribe to the list, including the moderator or list owner.

---

- *Not saving the initial "welcome" message everyone receives upon subscribing to a new discussion list.* It contains vitally important instructions and information you *will* eventually need. No matter what kind of a list you subscribe to, you should save the welcome message, because you will need it later.

*Evolved Response:* Save the initial message you get back from every list you subscribe to. You may want to print it on paper and save it in a safe place, in case of disk crashes. Gurus know that every list has little quirks that make it slightly different, so they save the instructions, knowing they will someday need them.

You may need to temporarily unsubscribe from a list, for example, even though you intend to read that favorite list faithfully until the day you retire. Someday you may move and/or need to change your access account name (also called your login or login name). Or, if your employer supplies your Internet access, your employer may change the name of the host computer(s) at your access site. Before you move, or just before your host computer's name is changed, you should unsubscribe from all of your Listservs! Why?

Well, it's because the robot at the remote Listserv site will get confused if it starts getting mail from you at a new address. It won't believe it's you because the new address does not match its careful records.

If you don't unsubscribe and then resubscribe from the new address (or new account), the next time you try to ask the remote computer to hold your mail for a vacation, or to unsubscribe, or whatever, you won't be able to get the robot to cooperate, because it won't believe it's really talking to you.

So, even if your employer sets up a forwarding arrangement and you don't miss any of the incoming mail from your subscribed lists, you are still just waiting for problems, because the remote computer doesn't really know you changed accounts or host names.

Unsubscribe and then, later, resubscribe from the "new" address. It will save you lots of trouble later.

- *Using the wrong subject heading for the topic you want to discuss.* It's helpful and courteous to your readers if you fill in the subject line for every message. Readers who receive upwards of 50 messages a day come to necessarily rely on the subject line in processing their mail.

*Evolved Response:* Certainly change the subject line when you introduce an entirely new topic. But don't choose an entirely different subject heading when you are contributing to an ongoing topic or "thread." It is much easier for everyone else reading the list to process, file, and relocate your idea by its association with the subject heading.

- *Including all of someone else's message when you wish to respond to only parts of it.*

*Evolved Response:* Include only the few quoted lines necessary to give your reply context and meaning. Rarely are more than four or five lines necessary for this purpose. Why force everyone to wade through all that verbiage again? Many people, especially in foreign countries, pay exorbitant prices per line for their Listserv subscription. You can drive them off the list by posting unnecessary lines. The next error is related to this one.

- *Addressing several topics in the same message.* Mixing two or more relatively unconnected ideas into a single, complex message.

*Evolved Response:* Write two or more separate messages. Each one will be shorter and simpler, and your colleagues will praise you and say you express yourself clearly and succinctly.

- *Reacting to another list member's errors and breaches of Netiquette by sending off a message intended to discipline the miscreant—but sent to the entire list, thereby merely compounding the offense.*

*Evolved Response:* This is the reason for personal mail. Address the offender, but remember, we are all newbies in some sense, and the Net ethic is intended to encourage experimentation and learning. If you can't say something nice in your message, maybe you should just delete it and move on to something constructive?

The Infobahn Litter Crew is collecting and compiling your *constructive* responses to newbie errors. If you have an extraordinarily brilliant, or just plain pretty good way to help clean up the Net after the next incautious wave of newbies, please send them to Infobahn Litter Crew, P.O. Box 524, Austin, Texas 78767-0524.

## A Pedagogue's Internet Tool Kit

Pedagogical gurus achieve their prominence by deep self-understanding, and by having tools to get the job done. Here are some methods to deal with emergencies, to exit ungainly programs, to manage the wait during net-lag delays, and to reduce the likelihood of delays in the first place.

### Emergency Procedures

You can't hurt the Net. It was designed to withstand full-scale nuclear attack, and to resist even malicious crackers. Don't worry—your explorations, no matter how primitive, are simply not going to hurt the Internet.

As you become more and more skilled on the Net, your ability to surmount crises also increases. But most of the problems you run into are *still* human error. It's just a question of which human you end up blaming! The more you know, the more you know (what) you don't know, so to speak.

Most of the errors made by Internet gurus are the same things anybody could do, only on a grander scale. Anybody can mistype the word "your" as "you," but it takes a real guru to send it out to 5 mailing lists so that 2750 people get *you* silly message!

Here are some procedural tips to help you deal with the more common emergencies and inconveniences that gurus encounter on the Net.

### Whistle While You Wait: Remote Access at a Snail's Pace

Patience, as they say, is a virtue. If a remote computer doesn't respond quickly, sometimes a novice too quickly assumes that the software has gone awry, and locked up the computer.

For gurus, usually, it's just that a bigger job than yours has come along, and has shoved you out of the spotlight for a while. Or, if this happens during daylight hours, the business (such as NASA, for instance) that owns the remote computer gives its own local workers much higher priority than remote users.

If you're lucky, the big job that displaced you will run itself out in a minute or two, and you will be back in business. If that two-minute wait is too much for you, or you've *already* waited five minutes to no avail, try some of the following tricks to get the attention of the remote machine, or bail out of a nonproductive trial.



**Tip:** If you're a Windows user and you just know that the site you're about to visit *always* makes you wait a long time for a response, why not be ready to play a game of Cruel solitaire, that comes in the Windows Entertainment Pack (WEP)? Just open the game's window and minimize the dealt solitaire game, and it will be waiting when you get stuck in wait mode.

Here are some things to try if you run into the Internet doldrums. Depending on your hardware and software, some of these tricks can force you to reboot your computer and/or reestablish your remote connection if they do *not* work—so be sure to note the ones that do and don't work for your configuration, so you can choose wisely in the future. Software and hardware respond unpredictably to tricks designed for systems other than their own!

control ]            (press two keys at once)

Esc

End

control c (press two keys at once)

control ^ (press two keys at once)

control Break (press two keys at once)

control End (press two keys at once)

## Giving a Courteous Farewell

Because there are so many ways to exit a computer system or a software program, it's very difficult for even a guru to remember the exact method in every situation. It is best if the software or remote computer politely tells you the exit procedure—but sometimes they just don't say.

Then, what do you do to get out? Well, typing h or help sometimes gets you a few cryptic tips, but there are often enough situations where there is just no guidance given.

So, every guru unfolds their own personal mantra of exit sequences, but here are a few collected by highly advanced gurus, just to get you started:

### Exit Mantras

|            |                                |
|------------|--------------------------------|
| q          | then press Enter or Return, or |
| exit       | then press Enter or Return, or |
| quit       | then press Enter or Return, or |
| close      | then press Enter or Return, or |
| bye        | then press Enter or Return, or |
| Esc        | then press Enter or Return, or |
| End        | then press Enter or Return, or |
| goodbye    | then press Enter or Return, or |
| logout     | then press Enter or Return, or |
| logoff     | then press Enter or Return, or |
| ciao       | then press Enter or Return, or |
| disconnect | then press Enter or Return     |

## Connect to the Planetary Clock

Many Internauts and Net gurus have chosen to deal with the remote computer waiting problem by adjusting their on-Net work clocks. (Remember, I said the Net makes time and space disappear?) If you use a remote machine when the time at the remote site is, say, between midnight and 6 a.m., you usually have very little competition for access, and get much faster response times. So, Internet gurus try to choose their remote computer sites according to the following type of time shifted schedule:

| <i>Guru Position</i>          | <i>Preferred Remote Site</i> |                    |
|-------------------------------|------------------------------|--------------------|
| <i>Time in Chicago/Dallas</i> | <i>Location</i>              | <i>Remote Time</i> |
| 8 a.m.                        | Hawaii                       | 4 a.m.             |
|                               | New Zealand                  | 2 a.m.             |

|         |             |             |
|---------|-------------|-------------|
| 10 a.m. | New Zealand | 4 a.m.      |
|         | Japan       | 1 a.m.      |
|         | Hong Kong   | 1 a.m.      |
|         | Sidney      | 1 a.m.      |
| 12 Noon | Japan       | 3 a.m.      |
|         | Hong Kong   | 3 a.m.      |
|         | Sidney      | 3 a.m.      |
| 6 p.m.  | England     | 12 Midnight |
|         | Germany     | 1 a.m.      |
|         | France      | 1 a.m.      |
|         | Italy       | 1 a.m.      |



**Tip:** *Waking go West, evening go East.* The rule of thumb for North American gurus is: In the morning you want to connect to remote sites that are *West* of you, in Hawaii, New Zealand, and Australia and Asia.

But when evening comes around, you want to connect to remote sites toward the *East*—in England, Germany, France, and Italy, for instance.

In the late hours of night in North America—*use North American sites!*

Here's something to meditate on: Time on the Internet is a most relative concept. Think about this—right *now* it is every hour of day, somewhere on this spinning, spherical planet!

## WAIS: Wide Area Information Servers

So as fast as I could,

I went after my net.

And I said, "With my net

I can get them I bet.

I bet, with my net,

I can get those Things yet!"

Then I let down my net.

It came down with a plop!

And I had them! At last!

Those two Things had to stop.

Dr. Seuss, *The Cat in the Hat* (New York: Random House, 1957).

*WAIS* is a distributed natural-language indexing, search, and retrieval tool. Right! Now what does that mumbo-jumbo mean?

It means you can use your regular vocabulary to search for things instead of fancy computer jargon. *WAIS* lets you ask simple questions in your own words and get lots of useful results from many computers and databases. That's better than the alternative of using special query languages to search one computer at a time, getting smaller results with less assurance of relevance and quality. The final result of a *WAIS* query is a set of document references (and full text in some cases) that contain the words you asked for.

WAIS enables you to search many databases all over the Internet. It searches the full-text of databases you select (called *sources* or *source files*) by asking computers all over the Net to check their index files for words you're asking about. WAIS index files end with the extension .src, such as:

ANU-Demography-Publications.src

bryn-mawr-medival-reviw.src

directory-of-servers.src

document\_centr\_catalog.src

inet-libraries.src

jte.src

With some WAIS programs, you can also use Boolean logic (AND, OR, NOT, or ADJ, for instance) to ask for combinations of words, so you can get closer and closer to exactly the items you want.

Public-domain clients for accessing WAIS are available for the following platforms: Macintosh, MS/DOS, Windows 3.x, VMS, NeXT, UNIX, and X Window. These clients are available via anonymous FTP from the University of North Carolina.

Access:

### **Anonymous FTP**

site sunsite.unc.edu

directory /pub/wais

## **Single Source WAIS Searches from Your Local Gopher**

Gophers are now delivering the most common, and simplest WAIS searches. Many Gophers have a WAIS search option, under the options

Other Gopher and Information Servers/

WAIS Based Information/

The only disadvantage of this kind of WAIS search is that you can't search more than one database source at a time.

## **Searching Many Databases at One Time**

You can search *many* databases at one time using the telnet site demonstrated later in this chapter. Unless you install a WAIS search program on your own personal computer, for serious searching gurus use a telnet WAIS search, because it's most efficient in using the Net's resources.

The user interface on the free telnet version of WAIS is real chunky, but you may learn to love this gnarled Internet tool.

The WAIS (pronounced like waist without the "t" at the end) output from this site is rough, and a little hard to understand. And you must always do a search *at least* twice, to get anything really useful. So why on earth would anyone want to use WAIS?

Because it's based on the *actual contents* of the files—not file *names*, not directory *names*. Consequently, it won't be so likely to lead you to an empty directory. And as you get more sophisticated in constructing searches, you can eliminate wrong items and find only high quality "hits."

Well, maybe that's overstating the case for WAIS a little. WAIS doesn't actually look inside files, really. It looks at sophisticated *indexes* of database files.

Astonishingly, the real contents of the target file, the one you're looking for, is *not* limited to words, so a WAIS index can lead you to graphic files, fonts, video and sound files, just about anything that can be put on a computer disk.

The key idea here is that indexes can refer to anything that can be described (which is a pretty big universe, to start with). Indexes can even refer to descriptions of odors and smells, can't they? Ever wonder if there was a database of fresh fruit or

oven-baked pastry smells? Now that's an idea for a guru's meditation.

## WAIS Indexes

Archie searches FTP sites worldwide; Jughead searches some Gopher menus; Veronica searches Gopher menus worldwide. They all do valuable searches, but none of them looks *inside* the index files to find content words, groups of words, and phrases the way WAIS does. The others look at menus and directory names, but only WAIS looks into the contents of the files, using indexes.

WAIS is associated with a standard method (ANSI Z39.50) for how to construct indexes for searching. So, WAIS is dependent on the quality and thoroughness of the indexes it is searching through. (The same limitation applies to Archie, and to the Whois system.)

There are over five hundred WAIS indexes (of databases) already built and available on the Net. The number is growing constantly. They cover educational topics of every variety, including:

agriculture  
beer making  
computer science  
human genetics  
literature  
molecular biology  
religion  
science fiction  
social science  
special education



**Note:** A complete listing of the databases is available from the University of North Carolina.

Access:

### Anonymous FTP

site sunsite.unc.edu  
directory /pub/wais  
file wais-sources.tar.Z

---

The separate databases are organized in different ways, using various database systems, but you don't have to learn the query languages of the various databases. WAIS uses your natural language queries to find relevant documents for you.

The technical experts say that no semantic information is extracted from WAIS queries. Part of what they mean is that WAIS indexes don't tell you if the search words are in any particular order or proximity. A WAIS index just tells you the words *are* in the document. Words that are of key importance to the indexed database, like those in document titles, words in all capitals, and words with initial capitals are given a little more weight in the index.

And there are also lots of stop words that just don't get indexed or searched for—like the words *and*, *the*, *but*, *with*, and so on, because they are too common to mean anything important in an index or a search. But they are a vital part of *natural-language* searching, since they allow you to ask a questions like "Please show me things about the space program?" WAIS hears that question as "space program" or "program space" and just ignores all other natural language words.

## Performing a WAIS Search

There are many WAIS servers throughout the Net. The main directory-of-servers database is available at several sites. Here is how you can address a query to it.

The host computer named quake.think.com is a demonstration site; anyone can telnet to it. At the login: prompt enter wais; no password is needed.

Although the client interface differs between platforms and versions of WAIS, each variant of the WAIS interface requires queries to be performed in the same way, whatever the interface:

- The user selects a set of databases to be searched from among the available databases. This first stage of your search requires you to select extremely broad keywords, in order to select the WAIS databases relevant for your next, narrowed keyword search.

If you're really seeking legal discussions of recent U.S. Supreme Court opinions, you would start your first WAIS search by asking "tell me about law and government." This first search is to look through the database of databases, to find the right databases to use for the real search, which you'll do second.

WAIS will return a list of databases that have "law and government" in them, and you can look at each database's brief description, if you are in doubt about it's relevance. Now you are ready to tell the WAIS program which databases to use in the next search - you're ready to select databases for your real search.

Selecting the databases is slow and laborious. Each one requires you to highlight the database's line in the results of the first round of your search, and while the database name is highlighted you must type u for "use this database in the next search."

After highlighting and typing u for each database you select to search, you then must type s to go back to the source database selection screen, and again highlight each database to be used and this time, type an asterisk while the line is highlighted for each and every database you want the next search to be based on. Figure 14.2 shows an example of a WAIS "source" screen, with source databases available for searching shown.

### [Figure 19.2. Each database to be searched must have an asterisk, before it will be searched for the keywords you specify next.](#)

Next, formulate a query by giving narrow, specific keywords to be searched for, like "tell me about U.S. Supreme Court Opinions."

When the secondquery is run, WAIS asks for the keyword information from each selected database.

Headlines of documents satisfying the query are displayed. The selected documents contain the requested words and phrases. Selected documents are ranked according to the number of matches within them. Figure 14.3 shows you a sample of WAIS output, showing how well the "hits" it found matched the search keywords by ranking and scoring each hit.

### [Figure 19.3. WAIS scores start from 1000 and go down as the matched item corresponds less and less to your keywords.](#)

1. To retrieve a document, the user simply selects it from the resulting list. Some documents are very short, and amount to only an index card entry. But others are research executive summaries, and lead you to highly relevant sources of further information.
2. If the response is incomplete, the user can state the question differently or feed back to the system any one or more of the selected documents he finds relevant. Try your search again, with different keywords. You will lose all the work of your database selections when you type q to quit, so you should try the search at least twice while you are at it.
3. When the search is run again, the results are updated to include documents that are similar to the ones selected, meaning

documents that share a large number of common words.

Learning more about WAIS:

Bug reports, comments, and suggestions about this version of WAIS, may be addressed to:

Email George Brett [george.brett@cnidr.org](mailto:george.brett@cnidr.org)

You can also join a Listserv discussion of WAIS:

**Subscription:**

e-mail [wais-discussion-request@think.com](mailto:wais-discussion-request@think.com)

body [subscribe wais-discussion@think.com](mailto:subscribe wais-discussion@think.com)

There is also a Usenet newsgroup:

Usenet newsgroup: [comp.infosystems.wais](mailto:comp.infosystems.wais)

## Online Information: Current Newspapers and Magazines

The advantages of electronic text in education are stupendous. Once a student has used a word processor to search for important themes, or follow the development of a particular character in a Shakespeare play, that student will never again doubt the utility of learning how to use a word processing program.

Here are three sources of up-to-date electronic texts for classroom use.

### Electronic Newsstand

Each document on the *Electronic Newsstand* part of this Gopher contains a copyright notice. Be sure to read and understand the notice before you use these current events and policy papers in any inappropriate way. Figure 14.4 shows a sample menu, and a keyword search in process.

#### [Figure 19.4. Search for articles using keywords that interest you.](#)

There is a nice keyword search available here, so you can search for articles you are interested in, rather than being forced to peruse all of the articles.

**Access:**

Gopher <gopher.eneews.com> 2100

### Online Magazines: Uncover by CARL

*CARL* is the *Colorado Alliance of Research Libraries*. CARL is rapidly becoming the most advanced electronic library in the world.

The following extended quote is almost an entire "beta-issue" of the occasional Internet magazine *Net Surfing*. This issue focuses on CARL and gives good explanations and examples of how to use CARL to search and obtain Electronic texts.

Surfing #02 UnCover: the biggest magazine rack you ever saw

When I was a boy, every town had a place like Mulligan's, a neighborhood market filled with as much clutter as the proprietor could cram in. The haven in the clutter was often a back wall devoted to a vibrant collage of magazines screaming with info

about the latest craze. Always there was one to capture a new interest and after a few passes through the table of contents, you could tell if it was worth the hard earned coin.

Ever wish there was a place like Mulligan's on the Internet super highway? A place where you could check the latest issue of your favorite magazine or journal and decide whether it was worth purchasing? Better yet, why not have this fictitious Mulligan's mail you the Table of contents as soon as it hits the newsstand. And what about all those pet projects, when you need to search through a mountain of past issues. Wouldn't it be great if there were a place on the Internet where you could search for articles on any topic or by any author and then have those articles forwarded to you.

Guess what! It's there, on the Internet super highway -- the biggest magazine rack you have ever seen. It's called UnCover, and includes more than 5,000 magazines and journals. The service is vast, with an archive of well over 5,000,000 articles that can be searched on and then FAXed to you within 24 hours. And yes, you can even have the service e-mail you the table of contents of your favorite monthly, soon off the press. And what about cost? Well the searching is free and so is the Table of contents through e-mail. It does cost, though, to have articles FAXed.

This issue of Surfing will be devoted to helping you explore the services of UnCover.

Part I: Logging on to UnCover

At your Internet prompt type: Telnet database.carl.org

You will then be asked to select a terminal type from a list. For most, the selection will be item (5) VT100. If you are unsure,

check with your computer folks. Once the terminal type is selected, you will be at the opening screen for CARL.

CARL is an acronym for Colorado Alliance of Research Libraries, a not-for-profit organization formed in 1978 to create ways for libraries to share information. One division of CARL services is UnCover, an article delivery service. Until October of last year, this service was available to only member libraries, but can now be accessed free of charge by anyone with an Internet account.

Most of the services listed on the opening menu for CARL are restricted to member libraries. You can explore items 1 (UnCover), 7 (Other Databases, which includes the Journal Graphic Database) and 8 (CARL Systems Library Catalogs). These have public access. For our purposes, we will be using UnCover. To enter the UnCover services:

Type: 1 at the >> prompt and press <RETURN>.

You will then encounter a series of four different prompts. For now, keep pressing the <RETURN> key until you come to the screen pictured below. These early prompts are for member libraries to insert their password or for you as a single user to insert your Profile Account number. A Profile Account is not necessary to use the search functions of the service. Establishing a Profile Account provides UnCover with some basic information about you -- name, e-mail address, etc. You will need a Profile Account if you choose to have the Table of contents of various magazines forwarded to you. (You might later want to re-enter UnCover and establish a Profile Account for yourself.)

=====  
Main Menu for Uncover  
=====

Welcome to UnCover

The Article Access and Delivery Solution

UnCover contains records describing journals and their contents.

Over 4000 current citations are added daily. UnCover offers you

the opportunity to order fax copies of articles from this

database. Type ? for details.

For information about a new service, UnCover Complete, type ?C

Enter N for NAME search

W for WORD search

B to BROWSE by journal title

QS for QUICKSEARCH information

S to STOP or SWITCH to another database

Type the letter for the kind of search you want,

and end each line you type by pressing <RETURN>

SELECTED DATABASE: UnCover

ENTER COMMAND (? FOR HELP) >>

=====

Congratulations! You are now logged on to the UnCover system. The

service is all menu driven -- typing the letter or number of a

menu item takes you to the next level of service. You should feel

comfortable exploring, there is little trouble that you can cause.

As is true with most Internet services, things slow down during

the middle of the day, so you might want to play around with

UnCover at a time when traffic on the net isn't as heavy.

Part II: Using NAME and WORD search

Both the Name and Word search features work similarly. If you are

looking for authors, you would use NAME search and if you are

looking for words in titles you would use the WORD search. WORD

search will only search for words in titles and in the brief summaries that are contained on the Table of Content pages of magazines; it will not search for words in the actual articles themselves (too bad).

Here is an example to show how the features work.

Type: W at the >> prompt

Press <RETURN> (to enter WORD search)

At the next prompt, you can enter the words you are searching for.

You can enter one or several words depending on your criteria. For our example:

Type three words at the prompt: Liberal Arts Job

Press <RETURN>

A list of four articles that meet our criteria will be displayed (pictured below).

=====  
List of Articles  
=====

LIBERAL + ARTS + JOB 4 ITEMS

Set of 4 will display on one page -- proceeding with display...

1 LaMarco, Terri (The black collegian. 03/01/94)

Preparing for Today's Job Market: Concrete advice fo...

2 Chesler, Herbert (Journal of career planning & empl..3/1/94)

Tell Them That a "Pure" Liberal Arts Degree Is Markd...

3 (National review. 05/11/92)

Books, Arts & Manners.

4 Sharp, Laure M. (The Journal of higher education. 9/1/89)

Early Careers of Undergraduate Humanities Majors.

<RETURN> to CONTINUE, Number + M (ex. 3M)to MARK article

Enter <Line numbers> to see FULL records

<P>revious for PREVIOUS page,<Q>uit for NEW search

=====

If your list is too long, you will be prompted to provide additional words to help narrow your search. Looking over the list, you will notice that item 2 includes our familiar "Journal of career planning and employment". This is pretty neat! UnCover provides an index to our own professional journal.

You can now preview a summary of each article in the list. In this example we will preview Laure Sharp's article (item 4 on the list)

Type: 4 at the prompt and press <RETURN>

The information pictured below will appear on your screen listing author, title, summary and location or source of the article. At the bottom of the information you are given the cost to have the article FAXed to you. This is as far as the service will take you; you are not able to preview the actual article.

===== Article Summary =====

AUTHOR(s): Sharp, Laure M.

Weidman, John C.

TITLE(s): Early Careers of Undergraduate Humanities Majors.

Summary: Data from the 1979 follow-up of the NLS-72 sample

suggest that the early career experiences (two to

three years post-baccalaureate) of undergraduate

humanities majors differed little from their

contemporaries who had majored in several other

liberal arts fields, but contrasted with those in

which job-major fit was better or pay higher.

In: The journal of higher education.

SEP 01 1989 v 60 n 5

Page: 544

This article may be available in your library, at no cost to you.

To have it faxed from UnCover, the following charges apply:

Service Charge: \$ 8.50

Fax Surcharge: \$ .00

Copyright Fee: \$ 1.25

---

Total Delivery Cost: 9.75

<R>epeat this display,

<M>ark article <Q>uit,

<H> for Search History, ? for HELP >

=====

For me, at this point, it's a trip to the local library. For one thing my library may subscribe to the journal and I can get it for free or they usually get a better rate to have the article FAXed - - many libraries pay an annual subscription fee to the service. Also, if you don't mind waiting for 6-10 days, you can probably get the article free through inter library loan.

If you choose to have the article FAXed to you, you would enter M at the prompt. A new item will be added to your command choices: <O>rder. Once you have marked all the articles that you would like FAXed, type O at the prompt. If you haven't established a Profile Account you will be prompted for your name, FAX number and credit card number. Then presto, the article will be at your FAX machine within 24 hours.

### Part III: Browsing by Journal Title

Another way you can use the service is to browse through the table

of contents of various magazines. This feature is accessed at the Main Menu for UnCover. If you are in a different section of UnCover, type Q and <RETURN> to trace your steps back to the Main Menu. Once there:

Type: B at the >> prompt

Press <RETURN> (to enter Browse by Journal Title)

You will then be asked to type the name of the journal that you wish to browse. For our example, type: Internet World

UnCover will then display a list of several journal titles.

Internet World should appear on top of the list.

Type: 1 and <RETURN> to access the Internet World entry.

The information pictured below will appear on your screen listing the name of the magazine, publisher and brief description.

=====  
Journal Summary  
=====

TITLE(s): Internet world.

Westport, CT : Meckler Corp.

OTHER ENTRIES: Internet (Computer network) Periodicals.

Computer networks Periodicals.

Continues: Research & education networking 1051-4791

The UnCover Complete service allows you to order articles from any issue of this journal, provided you have a complete citation.

Type ?C for details.

-----UnCover-----

<R>repeat display, <REVEAL>E-mail Alerting <C> UnCover Complete

<Q>uit,

<E> to Examine Current Issue Information >

<H> for Search History, ? for HELP >

=====

You now have several options available to you. Let me explain the REVEAL option. The REVEAL options allows you to create a list of magazine titles which are of interest to you. When the next issue is received, UnCover will automatically send the table of contents to your e-mail address. For this to work, you will need to establish a Profile Account. You can do that now:

Type: //Profile at the prompt and <RETURN>

UnCover will bring you through the steps. You could also establish a Profile Account when you first enter the program.

To continue with our example, use the E command to see a list of issues that are on file with UnCover:

Type: E at the prompt and <RETURN>

What you will see is the list pictured below.

===== Journal List =====

TITLE: Internet world

ISSUE STATUS CONTENTS

|   |          |   |   |   |   |           |             |
|---|----------|---|---|---|---|-----------|-------------|
| 1 | 08/01/94 | v | 5 | n | 6 | Expected  | Not present |
| 2 | 07/01/94 | v | 5 | n | 5 | Published | Present     |
| 3 | 06/01/94 | v | 5 | n | 4 | Published | Present     |
| 4 | 05/01/94 | v | 5 | n | 3 | Published | Present     |
| 5 | 03/01/94 | v | 5 | n | 2 | Published | Present     |

-----  
<RETURN>to continue, or <R> to repeat display

<C> UnCover Complete<Q> to Quit

=====  
As we did in the WORD search, you can now type the number of the issue to bring up the table of contents for that issue. Once you



|                               |            |
|-------------------------------|------------|
| Financial Times (London)      | Mon.-Fri.  |
| SCIENCE/ENVIRONMENTAL:        |            |
| Discover Magazine             | Monthly    |
| International Wildlife        | Monthly    |
| Mechanical Engineering        | Monthly    |
| Animals                       | Bi-Monthly |
| National Wildlife             | Monthly    |
| Sea Frontiers                 | Bi-Monthly |
| Ranger Rick                   | Bi-Monthly |
| SOCIAL SCIENCE:               |            |
| The Brookings Review          | Quarterly  |
| Foreign Policy                | Quarterly  |
| Insight on the News           | Weekly     |
| The New Republic              | Weekly     |
| The National Review           | Bi-Weekly  |
| FOREIGN LANGUAGE:             |            |
| Journal Francais D'Amerique   | Weekly     |
| Moscow News                   | Weekly     |
| MULTI-CULTURAL/INTERNATIONAL: |            |
| Beijing Review                | Weekly     |
| China Today                   | Monthly    |
| Islamic Affairs               | Monthly    |
| Jerusalem Post                | Daily      |
| London Daily Telegraph        | Daily      |
| France Today                  | Monthly    |

## Online Coursework

Many college and university courses and degree programs are already being offered on the Internet or have Net use as a vital course component. Some have brief, on-campus components as well, that require the student to be physically present at some particular location. Many offer videotaped classroom periods as one of the primary adjuncts to e-mail and telnet connections.

The samples of college and postgraduate programs below were chosen because they do not require *any* on-campus attendance, are open to the public, are not exclusively limited to students in a single state, company, or local broadcast area—and e-mail is mentioned as a primary course component in their entry in *The Electronic University: A guide to Distance Learning*, Peterson's Guides, c. 1993, Princeton, New Jersey.

Each entry in the Online Educational Resources section below has an e-mail address, telephone, fax, and surface mail address. Every student should carefully explore these opportunities before investing heavily in them. Make sure this method of learning meets your needs and fits your lifestyle *and your learning style* before you commit to a multi-year degree program!

The courses available *now* heavily emphasize continuing education and professional development for highly technical occupations—Computer Science, all kinds of Engineering, and Telecommunications. But you will also find listed below programs in Management, Business, Education, Food Science, Human Services, Nursing, Social Work, and Library Science. Soon other professional schools will begin to make offerings on the Net, to position themselves in this new arena of the educational marketplace.

## Online Instructional Methods

The diversity of instructional methods in these distance education offerings reflects the newness of this way of *doing the business of education*. Education has been done face to face, students and teacher all in one physical place all at one time, for thousands of years. That's the whole concept of a school, isn't it? Well, today there are many new pedagogical technologies available, including

Fiber Optic Data, Communications, and Video

Internet BBS, Computer Conferencing, and E-mail

ITFS—Instructional TV Fixed Service

Laserdisk

Microwave Transmission

Paper-based Correspondence

Satellite TV

Telephone, Fax

Video Conferencing and Compressed Video

Video and Audio Cassette

Video Broadcast, Public TV, Cable TV, Closed Circuit TV

Of course, most of the old limitations of time and space still apply to most of the preceding. And the options above also largely leave the "teacher" in control of events, rather than the learner/student being a self-reliant, adult, independent agent. Truly independent learning, guided by educators, may remain largely in the Internet's realm for some time to come.

Of the above technologies, those that are most under the learner's (time and space) control are:

- Internet BBS, Computer Conferencing, and E-mail
- Laserdisk
- Paper-Based Correspondence

- Video and Audio Cassette

## Online Learning Resources

The way we see education is changing. Once, a person expected to go to school, *get an education*, and after graduating go to work in a single long career, and then retire. Now, we are realizing that many other combinations and permutations of education, career, and retirement are possible, and probably desirable.

Why must education be a one-time event, with a single rite of passage at the end of it? This "tonsillectomy" model of education (once it's done, it's *done*) is losing favor. A new model is gaining attention both in schools of education and in corporate circles: A nutritional or *gourmet* model of education—learning is pleasant and necessary, and is best done regularly in prearranged and well designed cycles.

The online learning resources listed below presage the diminished importance of all central school campuses. Learning need not be at a single location—the Net is everywhere. The Net can be a tremendous boon to isolated rural schools. This is because if rural areas actually receive their promised equal access to the Net, they will be able to supply a much higher quality of education without the immediate physical presence of all the amenities, resources, and dangers of urban schools.



**Note:** Many states, provinces, and localities have excellent programs for their state or region. But programs are listed here only if they offer open admission to qualified members of the public *nationwide*.

---

But many excellent programs are geographically limited, primarily by the hardware they use. For instance, broadcast TV and cable TV are limited to locations within their signal and cabled service areas, and ITFS (Instructional TV Fixed Service) is limited to a broadcasting radius of around 20 miles.

Therefore, in addition to the nationwide resources listed below, to find Internet-related distance learning opportunities in your local area, check with your local library or bookstore. Or you may want to consult the book *The Electronic University: A Guide to Distance Learning*, by Peterson's Guides, Princeton, New Jersey, which contains dozens of state, local, and regional educational programs with Internet components.

Here are 18 education programs across North America that include an Internet component.

### The Georgia Institute of Technology

The Georgia Institute of Technology offers master's degrees and various postsecondary certificates, in Electrical Engineering, Computer Integrated Manufacturing, Engineering Testing and Evaluation, Environmental Engineering, and Health Physics.

For Information:

e-mail: joe\_boland@conted.gatech.edu (Joseph S. Boland)

Voice: 404/894-8572

Fax: 404/894-8924

Surface mail Continuing Education Videos

Georgia Institute of Technology

Atlanta, GA 30332-0240

### Kansas State University

Kansas State University offers a postsecondary certificate in Food Science for food technologists, managers, and supervisors.

For Information:

e-mail: henders@ksvm.ksu.edu (Linda Henderson)

Voice: 913/532-5686

Fax: 913/532-5637

Surface

Distance Learning Program

College Court Building #231

Kansas State University

Manhattan, KS 66506-6007

### **The National Technological University**

The National Technological University offers master's degrees in several computer-related subjects through corporate, government, and university sponsors in the U.S., Canada, and Mexico.

For Information:

e-mail: doug.yeager@ntupub.ntu.edu (Douglas Yeager)

Voice: 303/495-6414

Fax: 303/484-0668

Surface

National Technological University

700 Centre Avenue

Fort Collins, CO 80526-1842

### **New York University**

New York University offers a post-secondary certificate in Management/Computer Science in a program called The Virtual College.

For Information:

e-mail: vigilante@acfcluster.nyu.edu (Dick Vigilante)

Voice: 212/998-7190

Fax: 212/995-4131

Surface

Information Technologies Institute

New York University

48 Cooper Square

New York, NY 10003-7154

### **Purdue University**

Purdue University offers master's degrees and continuing engineering education in several engineering areas including: Industrial Engineering, Electrical Engineering, Mechanical Engineering, and Civil Engineering.

For Information:

e-mail: swain@ecn.purdue.edu (Philip Swain)

Voice: 317/494-0212

Fax: 317/496-1196

Surface

Continuing Engineering Education

Civil Engineering Building #1575

Purdue University

West Lafayette, IN 47907-1968

### **The Rochester Institute of Technology**

The Rochester Institute of Technology offers several programs. This first one is a master's degree in Software Development and Management.

For Information:

e-mail: epr@cs.rit.edu (Evelyn Rozanski)

Voice: 800/CALL-RIT ( same as 800/225-5748)

Fax: 716/475-2114

Surface

Information Technology

Rochester Institute of Technology

P.O. Box 9887

Rochester, NY 14623

The Rochester Institute of Technology also offers a master's degree in Telecommunications Software Technology.

For Information:

e-mail: lutz@Chase.DE (Peter Lutz)

Voice: 800/CALL-RIT (same as 800/225-5748) or 716/475-6162

Fax: 716/475-7100

Surface

Information Technologies

Rochester Institute of Technology

91 Lomb Memorial Drive

Rochester, NY 14623-5603

The Rochester Institute of Technology also offers postsecondary certificates in Data Communications. Several other programs are available in Applied Computing and Communications, Health Systems Administration, Solid Waste Management Technology, Emergency Management, and Applied Arts and Science.

For Information:

e-mail: smwetc@ritvax.isc.rit.edu (Suzy Warner)

Voice: 800/CALL-RIT (same as 800/225-5748)

Fax: 716/475-5077

Surface

Distance Learning

Rochester Institute of Technology

Rochester, NY 14623-6310

### **Empire State College**

Empire State College offers bachelors degrees in Business, Human Services, and Interdisciplinary Studies.

For Information:

e-mail: dgranger@sescvaesc.edu (Daniel Granger)

Voice: 518/587-2100

Fax: 518/587-5404

Surface

Center for Distance Learning

Empire State College (SUNY)

2 Union Avenue

Saratoga Springs, NY 12866-4303

### **Syracuse University**

Syracuse University offers a master's degree in Nursing.

For Information:

e-mail: rmcolley@suadmin.syr.edu (Robert Colley)

Voice: 315/443-3284

Fax: 315/443-1928

Surface

Independent Study Degree Programs

Syracuse University

610 East Fayette St.

Syracuse, NY 13244-2140

### **The University of Alaska at Fairbanks**

The University of Alaska at Fairbanks offers several programs, including a bachelors degree in Education.

For Information:

E-mail: ffrjb@aurora.alaska.edu (Ray Barnhardt)

Voice: 907/474-6431

Fax: 907/474-5451

Surface

Off-Campus Programs

School of Education

Gruening Building #706 C

University of Alaska, Fairbanks

Fairbanks, AK 99775-6480

The University of Alaska at Fairbanks also offers a bachelors degree in Rural Development.

For Information:

e-mail: rfbmj@aurora.alaska.edu (Bernice Joseph)

Voice: 907/474-6432

Fax: 907/474-5451

Surface

Gruening Building #707 B

University of Alaska, Fairbanks

Fairbanks, AK 99775-6500

The University of Alaska at Fairbanks also offers a bachelor's degree in Social Work.

For Information:

e-mail: ffgsb@aurora.alaska.edu (Gerald Berman)

Voice: 907/474-6516

Fax: 907/474-5451

Surface

Behavioral Sciences and Human Services

Gruening Building #702 B

University of Alaska, Fairbanks

Fairbanks, AK 99775-6480

## **The University of Arizona**

The University of Arizona offers a program in Library Science in association with Mind Extension University and the Western Cooperative for Educational Telecommunications.

For Information:

e-mail: merri@ccit.arizona.edu (Merrilyn S. Ridgeway)

Voice: 800/955-UofA (same as 800/955-8632)

Fax: 602/621-3279

800/955-8632

Surface

Extended University

University of Arizona

1955 E 6th Street

Tucson, AZ 85719-5524

### **The University of Colorado at Colorado Springs**

The University of Colorado at Colorado Springs offers a postsecondary certificate in Early Reading Instruction in Association with Mind Extension University.

For Information:

e-mail: lkaaker@uccs.edu (Linda Aaker)

Voice: 719/593-3597

Fax: 719/593-3362

Surface

CU-Net

University of Colorado at Colorado Springs

P.O. Box 7150

Colorado Springs, CO 80933-7150

### **The University of Idaho**

The University of Idaho, through their Video Outreach Program, offers a master's degree in Engineering.

For Information:

e-mail: bwillis@uidaho.edu (Bruce Willis)

Voice: 208/885-6373

Fax: 208/885-6165

Surface

Engineering Outreach

Janssen Engineering Building #40

University of Idaho

375 S. Line St.

Moscow, ID 83844-1014

### **The University of South Carolina**

The University of South Carolina offers a master's degree in Library and Information Science. This program is being conducted only in South Carolina, West Virginia, Georgia, and Maine at this time.

For Information:

e-mail: [ilibs16@univscvm.csd.scarolina.edu](mailto:ilibs16@univscvm.csd.scarolina.edu) (Gayle D. Sykes)

Voice: 803/777-3858 or 803/777-5066

Fax: 803/777-7938

Surface

Nancy C. Beitz

Admissions/Placement Coordinator

College of Library and Information Science

University of South Carolina

Columbia, SC 29225-0001

### **The University of Washington School of Medicine**

The University of Washington School of Medicine offers American Medical Association continuing medical education units on the Internet. See Figure 19.5 for the home page, as it was announced to the Listserv Health Matrix. The text of the announcement is also given here.

#### **[Figure 19.5. The home page for the Department of Radiology at the University of Washington School of Medicine.](#)**

Date: Tue, 16 Aug 1994 18:16:14 -0500

From: (name and address removed)

Reply to: [hmatrix-1@ukanaix.cc.ukans.edu](mailto:hmatrix-1@ukanaix.cc.ukans.edu)

Subject: Continuing Medical Education

=====

Continuing Medical Education (CME) credits are now available online!

The University of Washington School of Medicine has designated the digital teaching file on the UW Radiology Webserver for credit hours in Category I of the Physicians Recognition Award of the American Medical Association.

The URL for this server is: <http://www.rad.washington.edu/>

Other learning modules are currently available on this server and will also be available for CME credit soon. This server includes the following items:

1. Radiology Teaching File
2. Anatomy Teaching Modules
3. Radiology Exhibits from UW

4. Information on UW radiology residency and fellowship programs

5. Image processing software written by UW faculty

For further information, contact:

Michael Richardson, M. D.

mrich@u.washington.edu

## **Electronic University Network (EUN): America Online**

The Electronic University Network is affiliated with several degree-granting higher education institutions, and is the exclusive provider of higher education on America Online, one of the largest commercial computer networks in North America.

EUN on America Online has a "virtual" campus, including admissions, finance, continuing education, and registrars offices, a library, and a student center.

For more information:

e-mail: eunlearn@aol.com

## **Schools Using the Net**

Here are just a couple of samples to show you that elementary, middle schools, and high schools are already using the Internet. Although Gopher is probably a more practical and inexpensive program for a school to operate, these examples focus on World Wide Web (WWW) home pages, simply because they are more graphic and *visually* interesting. Transferring graphics, and sound files, can be extremely slow and taxes the Net's bandwidth, so that some feel WWW may weaken the Internet rather than show off its best possibilities.

### **Grand River Elementary School: Lansing, Michigan**

Notice that the fifth-grade class at Grand River Elementary prepared this WWW home page (see Figure 19.7).

**[Figure 19.6. The WWW home page prepared by the fifth-grade class at Grand River Elementary school in Lansing, Michigan.](#)**

Access:

URL: <http://web.cal.msu.edu/JSRI/GR/BradClass.thumb.html>

For more information:

e-mail: Brad Marshall grsch21@pilot.msu.edu

### **Claremont High School: Claremont, California**

This site includes links to academic resources for Claremont High School students, current research data and publications in knot theory, and access to their anonymous FTP site, along with descriptions of the school and its computing resources. Students have created their own homepages (see Figure 19.8).

**[Figure 19.7. Home page created by Claremont High School students.](#)**

There is also a link to the Claremont Colleges' Library system. This WWW server also includes links to Pasadena's Jet Propulsion Lab, Harvey Mudd College's Web site, and Pitzer College's new WWW server for K-12 access.

Access:

URL: <http://www.cusd.claremont.edu>

For more information:

e-mail: [rmuir@chs.cusd.claremont.edu](mailto:rmuir@chs.cusd.claremont.edu) (Robb John Muir)

voice: 909/624-9053

## Online Database Access

Here are four companies that sell access to information databases. They usually charge either a flat fee per year or month, or a per hour connect charge, with additional varying fees for access to certain databases. Each of these services sells access to dozens of different databases.

All of the services in this section provide access to databases over the Internet, but they do not perform any searching for you. You have to learn to search the databases, and perform your own queries. The next section provides two search services that will do the search for you.

## CD Plus Technologies (formerly BRS Information Technologies)

This group has very strong coverage of Life Sciences and Medicine, and strong coverage of Business, Education, Patents, Sciences, Reference Sources, and Social Sciences.

Access:

e-mail: [alex1@cdplus.com](mailto:alex1@cdplus.com)

Voice: 800/955-0906, x1400

800/950-2035

212/563-3006

Fax: 212/563-3784

Surface 333 Seventh Avenue

New York, NY 10001

---

**IMPORTANT!**

**Tip:** Both DIALOG (following) and CD-Plus Technologies (preceding) offer patent databases. These are for historical purposes. Starting this year, new patent information is available on the Internet by e-mail subscription for free.

---

Subscription:

e-mail: [patents-request@world.std.com](mailto:patents-request@world.std.com)

body ascii all news

Patent information is very valuable for finding out what others are doing, for locating new technologies to license, and to measure rates of advancement.

For more information:

e-mail: [patents@world.std.com](mailto:patents@world.std.com) (Gregory Aharonian)

Voice: 617-489-3727

Surface

Source Translation & Optimization

P.O. Box 404

Belmont, MA 02178

### **DIALOG Information Services and Data-Star**

Because these two former competitors are now one company, they have a broader coverage than either would provide on its own. Data-Star brings strong European and Foreign Trade coverage, while Dialog has strong coverage of Business, Patents, Marketing Research, and Biotechnology.

Access:

telnet: dialog.com

voice: 800/334-2564

415/858-3785

fax: 415/858-7089

surface:

3460 Hillview Avenue

Palo Alto, CA 94304

### **GTE Network Services**

This group has very strong coverage of all Educational topics, and strong coverage of Health and Human Services topics.

Access:

e-mail: info@gteens.com

Voice: 800/927-3000

214/518-8500

Fax: 214/751-0964

Surface

5545 MacArthur Boulevard #320

Irving, TX 75038

### **Online Computer Library Center (OCLC)**

This group is a reseller of database access, and can provide access to DIALOG (mentioned previously), training in search strategy and database use, and other database services.

Access:

e-mail: amigos@utdallas.edu (Tonya Derrick)

Voice: 800/848-5878

614/764-6000

800/843-8482

Fax: 614/764-6096

Surface

6565 Frantz Road

Dublin, OH 43017-0702

## Online Research Services

Here are two commercial research firms. They will do research and deliver the results quickly and over the Internet!

### DialSearch

In order to arrange for billing, you will need to fax DialSearch to open an account, but they will accept search requests from existing accounts and deliver the results by e-mail. DialSearch uses Dialog and Data-Star, two online databases mentioned here, so they have good coverage of business news and marketing trends, current events worldwide, biotechnology, patents, industrial analysis, and much more. Searches average \$400.00 to \$500.00 each, and you should receive a price quote before you are obligated.

Access:

e-mail: 72262.1132@compuserve.com (Cindy)

Voice: 800/634-2564

### Knowledge One

Knowledge One is another fee-based information service for when you need information on any topic. Like reading glasses at the shopping mall, your research is ready in about an hour! Time and space become distorted and irrelevant when your mind travels through the Net to this Sonoma, California company (please see Figure 14.8).

Access:

WWW

URL: [http://KnowOne\\_WWW.sonoma.edu/](http://KnowOne_WWW.sonoma.edu/)

For information:

e-mail: knowone@sonoma.edu

**[Figure 19.8. Knowledge One is a commercial research service now available on the Internet.](#)**

## Communicating with Teachers

A teacher new to the Internet will want to start making inquiries among colleagues about which discussion lists are the most valuable. Listserv discussion lists and Usenet newsgroups are not all the same. Some are laconic, with little activity all year, while others are verbose and pedantic, or manic and terse, and have heavy activity every single day, even on weekends. Each of us has to follow our own bliss, so we accept that not all newsgroups or Listservs will be to our liking.

Here are two excellent resources for Educators and aspiring Net gurus to choose discussions of their own interest.

### Educator's Guide to E-Mail Lists

Prescott Smith of the University of Massachusetts at Amherst prepares and distributes two excellent lists of lists for educators. He founded and is list owner emeritus for one of the all-time best educational Listserv discussion groups, *Ednet*—a forum exploring the educational potential of the Internet.

Subscription:

e-mail: listserv@nic.umass.edu

body: subscribe Ednet *your\_name*

Here are the major headings for *An Educator's Guide to E-Mail Lists*, under each of which you can find dozens of Listserv

discussion lists. After obtaining the list (access instructions are below) you simply find the lists that are the most interesting to you and check them out!

Education - Adult

- Higher, Administrative
- Higher, Research
- Higher, Student
- Higher, Teacher & Faculty Development - General
- Higher, Teacher & Faculty Development - Subject Areas
- Higher, Teacher & Faculty Development - Media
- International Related
- K12
- Multicultural Related
- Special

Arts

Computer

Computer - PC Bibliography, Text

Future Studies

Help

History & Classics

Language - Communications Studies

- International
- Linguistics
- Literature
- Writing

Library & Information Retrieval

Philosophy

Science

Social Science

Statistics

Women's Studies

This guide is over 100 kilobytes, so please be sure you have available file space in your account.

Access:

**Anonymous FTP**

site nic.umass.edu

directory /pub/ednet

file educatrs.lst

## **Ednet Guide to Usenet Newsgroups**

Prescott Smith of the University of Massachusetts at Amherst also prepares and distributes this list of newsgroups for educators. Each topic entry below has from a dozen to several dozen newsgroups under it. You may find newsgroups you want to participate in. At the end of the document, Prescott includes several dozen newsgroups that are commercially available for a fee.

### **Usenet Newsgroups**

Education - Adult

Education - Higher, Academic

Education - Higher, Administrative

Education - Higher, Student

Education - Instructional Media

Education - International Related

Education - K12

Education - Multicultural Related

Education - Special

Arts

Computer

Future Studies

Help

History & Humanities

Language - Communication Studies

Language - International

Language - Linguistics

Language - Literature

Language - Writing

Library & Information Retrieval

Philosophy

Science

Science, Social

Statistics

Women's Studies

### **Commercial Newsgroups (\$)**

You may obtain the *Ednet Guide to Usenet Newsgroups* by anonymous FTP.

Access:

Anonymous FTP

site nic.umass.edu

directory /pub/ednet

file edusenet.gde

## Searching Listserv Archives

Chapters 5, "E-Mail," and 12, "Listservs and Mailing Lists," have already prepared you for some of the advanced topics to be covered here. The harried life of an educating guru requires time-saving devices. One of the fastest ways to do practical research on the Net is to search the vast archives of the Listserv discussion lists.

There are well over 5,000 Listserv discussions and newsgroups ongoing now, with more popping up all the time. In Chapters 5 and 12 you learn to manage the Listserv discussions that are valuable for keeping up with various parts of your intellectual-spiritual-physical life.



**Tip:** Here's a way to use Listservs to do your research! For the official documentation on Listserv archive searching, send e-mail to any valid listserv@ address with the message

info database

You will receive back a copy of *Listdb Memo, about Listserv database functions*. This is the authoritative source for Listserv archive searching, and you'll need it to do things like case-sensitive searches.

---

## Submitting a Search

All batch Listserv database searches are built inside of the job command language structure shown in this box.

```
//Scan JOB Echo=Yes
```

```
DATABASE SEARCH DD=Data
```

```
//Data DD *
```

```
    {your search goes here}
```

```
        . . . .
```

```
        . . . .
```

```
/*
```

The remainder of this detailed discussion of Listserv archive searching relies on you remembering that the preceding job structure must be around your search lines. The discussion below does not show you the job structure of every search, but all searches must be sent inside of that structure to work.

## The *search* Command

Probably the simplest search command is

```
search * in listname
```

which would result in the selection of all the messages in the archives of the list listname.

This search can be modified to include only messages sent after a given date, before a given date, or between two dates.

```
search * in listname since 92/01/01
```

```
search * in listname before 92/01/01
```

```
search * in listname from 92/01/01 to 92/06/30
```

The above messages will select all messages sent to the list, since January 1, 1992; before that date; and for the first 6 months of 1992.

Specific words can be searched, also. For example, the following search seeks two words, facilitated and communication.

```
search facilitated communication in listname
```

This search will identify a list of all messages containing the words facilitated and communication.

The search rules can be modified to send messages that contained words that appeared, for example, since the beginning of 1993.

```
search facilitated communication in listname since 93/01/01
```

It is also possible to select messages on the basis of the sender of the message. The following command returns a list of all the messages sent by Dave Kinnaman.

```
search * in listname where sender is kinnaman@world.std.com
```

In combination with a date command, only messages from Dave Kinnaman for the last half of 1992 will be returned by the following request.

```
search * in listname where sender is kinnaman@world.std.com from
```

```
-92/07/01 to 92/12/31
```

(Please note that the - character is used to indicate a continuation of the command on the next line.)

It is also possible to do *phonetic* searches:

```
search * in listname where sender sounds like cinnamon
```

The full complement of Boolean keywords (AND, OR, NOT, CONTAINS, and so on) are available in the Listserv database functions. The *Listdb Memo* provides complete documentation, with examples.

## The *index* Command

The index command provides you with a list of the messages selected by your search. The Index includes the number of the item, and its date, time, number of lines, and subject.

```
search * in listname
```

```
index
```

### Index of Database Search

| Item #            | Date          | Time  | Recs | Subject              |        |
|-------------------|---------------|-------|------|----------------------|--------|
| -----             | ----          | ----  | ---- | -----                |        |
| 000001            | 92/04/25      | 07:12 | 25   | A few words          |        |
| 000002            | 92/04/27      | 08:56 | 40   | A few words from me  | 000003 |
| 92/04/27 10:33 83 | Introductions |       |      |                      |        |
| 000004            | 92/04/27      | 11:30 | 21   | Re: A few words from | 000005 |

92/04/28 13:14 36 17 strong and still

This is an index of messages that meet the search criteria. It serves as a guide to which message should be retrieved.

The list command can be used to include the sender of the message in the index. The following commands will result in an index:

```
select * in listname
```

```
list sender.9 index
```

The list sender.9 part of the command, prior to the word index, will result in the first nine characters in the userid of the person who sent the message being added to the output fields usually contained in the index command.

List/Index Output

| Sender     | Item # | Date     | Time  | Recs | Subject                      |
|------------|--------|----------|-------|------|------------------------------|
| DRZ@SJUVM+ | 000001 | 92/04/25 | 07:12 | 25   | A few words                  |
| RJKOPP@SU+ | 000002 | 92/04/27 | 08:56 | 40   | A few words from me too!     |
| DOC@VTVM1+ | 000003 | 92/04/27 | 10:33 | 83   | Introductions                |
| jmwobus@M+ | 000004 | 92/04/27 | 11:30 | 21   | Re: A few words from me too! |
| RJKOPP@SU+ | 000005 | 92/04/28 | 13:14 | 36   | 17 strong and still growing. |

### The *print* Command

The search and index commands result in your receiving e-mail containing the index of "hits." If the command print *N* is included, such as

```
select * in listname
```

```
list sender.9 index
```

```
print 3
```

the text of message 3 will be sent to you.

Several messages can be requested at one time with the following command:

```
select * in listname
```

```
list sender.9 index
```

```
print 27 30-35
```



**Tip:** The following advanced search design is really worth saving.

```
-----Clip and Save-----
```

```
//ListSrch JOB Echo=no
```

```
Database Search DD=Rules OUTLIM=2000 f=mail
```

```
//Rules DD *
```

```
S word in listname
```

```
index #.5 date.8 sender.17 subject.46
```

```
print
/*
// EOJ
-----Clip and Save-----
```

If you replace word above with the keyword or words you are looking for and listname with the name of Listserv list, you receive back a numbered list of hits, including message dates, sender names, and subject lines.

---



**Note:** Listserv commands are always included in the body of your e-mail message, *not* the subject line, and are always sent to the Listserv robot clerical program—never to the people on the list.

---

## Using InfoMagnet to Search Listserv Archives

A new Windows-compatible program called InfoMagnet is so powerful it not only maintains your Listserv subscriptions for you, it also searches Listserv archives for you. It uses the same procedures just described, but automates them so they are almost invisible to the user! See Figure 19.9.

**Figure 19.9. This screen appears after you click on InfoMagnet's icon and as it is loading into the Windows graphical environment.**

The version demonstrated here is clearly a beta version, but it is quite impressive already. Figure 14.10 shows the New Discussion Group List selection screen, which enables you to choose from about two dozen major topic areas for further investigation. For all the remaining figures in this chapter, you should imagine we selected "Education" at this choice point.

**Figure 19.10. Based on your interests, you select an area for detailed exploration.**

As you can see, the Listserv discussion topic areas that InfoMagnet can show you include Apple Computers, Art, Biology and Chemistry, Business, Computers, Databases, Ecology, Education, Food, Graphics, History, IBM, and many more major areas.

Figure 19.11 shows you just the top of an alphabetical list of about 180 education-related listings already in this beta version of InfoMagnet.

**Figure 19.11. After choosing the main subject of Education, the next screen shows one education discussion list name per line.**

After choosing the main subject of Education, the next screen shows one education discussion list name per line.

Over 700 people currently subscribe to EDPOLYAN - the Education Policy Analysis Forum. Would you like to listen in? Or perhaps you'd prefer to participate actively? Figure 19.12 shows the details about this discussion list.

**Figure 19.12. InfoMagnet's "peek" feature enables you to see the details about each discussion lists.**

When you double-click a line in the Education list selection screen, the first detail screen, which InfoMagnet calls a "Peek," is shown for whichever list you clicked on.

As you can see from the InfoMagnet Peek at the list EDPOLYAN screen, a lot of information is provided on each of the Listserv discussion lists covered in InfoMagnet. Besides Peeking at each list, in the upper left of the screen, you'll see that you can initiate a search of the list's archives, or have InfoMagnet send the messages for you to join the list. Also available on this screen are several details about the list: Whether the list has archives or not, whether it is a public or a private "moderated" list, an estimate of the number of subscribers, the list's physical location, and a complete set of the Internet addresses needed to manage all business with this list is provided on the InfoMagnet Peek screen!

InfoMagnet really does handle everything for you. Figure 19.13 shows the settings you choose as defaults for how you want to

subscribe to Listservs. Every choice that is normally available to you is handled by the program—automatically! These choices include: Whether to subscribe as a full member, digest member, index member, or to subscribe but not receive any mail; whether to conceal your name from the "review" command; and how you want acknowledgment of your own posts to the list handled.

**[Figure 19.13. Set your default subscription preferences here.](#)**

Here are four more Peek screens, detailing some popular Education discussion lists. Figure 14.14 is for EdTech.

**[Figure 19.14. The details on the EdTech discussion list.](#)**

Figure 19.15 gives details on an Italian list called Hyperedu, which discusses uses of hypertext in educational settings.

**[Figure 19.15. Details on the list Hyperedu.](#)**

The Middle-I list is very popular among middle school teachers and administrators (see Figure 14.15).

**[Figure 19.16. The Middle-I list is very popular among middle school teachers and administrators.](#)**

And Figure 19.17 is for VOCNET, out of Berkeley, California.

**[Figure 19.17. "Peek" details on the discussion list VOCNET.](#)**

## **Personal (Filtered) News**

There are several news filtering services now operating on the Net.

This section looks at one that is fee-based but has a 30-day free trial period to get you started, and another that is free, with your subscription period a number of days that you may choose.

### **HeadsUp: INDIVIDUAL, Inc.**

HeadsUp is available free for 30 days. This is a filtered news delivery service based on around 12,000 articles per day from 300 sources. You select 5-10 topics (from 700) and get 2-sentence briefs by e-mail. Full articles can then be ordered from an 800 number or via e-mail, for delivery within 30 minutes. After the trial, it costs \$695 per year, or \$29.95 per month plus \$4.95 per full-text via e-mail.

Access:

E-mail: [headsup@enews.com](mailto:headsup@enews.com)

Voice: 800/417-1000

800/414-1000

617/354-2230

Fax: 617/354-6210

### **Netnews Filtering Service**

Here is an excellent *free* news filtering service. It was designed with funding from the Pentagon, so you know it has some serious horsepower behind it. It is being upgraded regularly, so you can expect this already fine service to continue to improve.

The brief instruction manual is included next, because you may need these details to manage your news filtering subscriptions. This is such a valuable service, you should have the complete instructions right from the beginning!



**Tip:** This filtering service *does not* use Boolean searching. It doesn't know what the words AND, OR, and NOT mean.

It also does not know that employer, employee, employment, and employ are related words. So your search *should* use all the variations on the same root word. Use them all and you will get more "hits."

---

Here's a quick example of how it works. For each filter topic you define, you send a list of words to the service—each list of words is a new subscription. This example would ask the service to search hundreds of newsgroups for messages that contain the words education rural funding schools teacher.

Subscription:

e-mail: netnews@db.stanford.edu

body: subscribe education rural funding schools teacher

The service will acknowledge your subscription and begin sending you by e-mail each evening the first 20 lines of any "hits" it finds. Hits are messages the service finds that have most of your search words in them. If the first twenty lines of a hit look good to you, you should ask the service to send you the whole message.

---



**Tip:** This service does not save messages for very long, so ask for any "hits" you really want within 48 hours of being notified of the message by the Netnews Filtering Service.

---

There are several more detailed options described in the following instruction manual. Again, this service is *free*, so what have you got to lose? Make up a search and let Stanford do the fishing for you!

Netnews Filtering Server

Department of Computer Science

Stanford University

Comments to: tyan@cs.stanford.edu

February 1994

( A ) I N T R O D U C T I O N

The Database Group at Stanford is providing a filtering service for NetNews articles. A user sends his profiles to the service, and will receive news articles relevant to his interests periodically. Communication to and from the service is via e-mail messages.

A user's profile is, in the style of WAIS [1] queries, just a plain piece of English text; e.g., "object oriented programming," or "nba golden state warriors basketball." Based on the statistical distributions of the words in the articles, scores are given to evaluate how relevant they are to a

profile. The highest possible score given to an article document is 100. The user can specify the minimum score for an article to be delivered.

Below we first look at a sample message from the user and a sample notification from the service. Then we describe details of the user message format.

( B ) E X A M P L E

A user who has sent this message to the service

=====  
From tyan@cs.stanford.edu Fri Feb 19 21:48:24 1993

To: netnews@db.stanford.edu

subscribe object oriented programming

period 3  
=====

will receive periodically (every 3 days, as specified) notifications such as this one:

=====  
From netnews@db.stanford.edu Mon Feb 22 10:40:35 1993

To: tyan@cs.stanford.edu

Subscription 1: object oriented programming

Article: comp.object.10836

From: eric@diku.dk (Eric Jul.)

Subject: I-WOOS'93 Call for Papers/Participation

Score: 83

First 15 lines:

CALL FOR PARTICIPATION

1993 INTERNATIONAL WORKSHOP ON

OBJECT-ORIENTATION IN OPERATING SYSTEMS

I-WOOS '93

The Grove Park Inn and Country Club

Ashville, North Carolina, U.S.A.

December 9-10th, 1993

Sponsored by

The IEEE Technical Committee on Operating Systems  
and Application Environments (TCOS) (pending)

....

=====

The user can then request the whole article sent to him if he thinks it is relevant.

( C ) U S E R M E S S A G E F O R M A T

User messages should be sent to netnews@db.stanford.edu. The subject field of the message is ignored. Each message is a request to the service. Each request consists of a number of commands. Each command must start with a new line with no leading spaces. Continuation lines begin with a space or a tab. All commands are case-insensitive.

Requests are associated with the return address of the user message. Service replies and deliveries will be sent to that address.

The usages of the commands are as follows.

( C . 1 ) S u b s c r i b i n g

To subscribe for articles, use these commands:

SUBSCRIBE word word ...           Subscribe for articles relevant to the profile specified by <word>'s. Maybe followed by PERIOD, EXPIRE, and THRESHOLD commands.

PERIOD period                       (Optional - default 1 days) Specify <period> as the period between notifications (in days).

EXPIRE days                         (Optional - default 9999 days) Specify <days> as the length (in days) for which

the subscription is valid.

THRESHOLD score (Optional - default 60) Specify <score> as the minimum score for an article to be relevant. The most relevant article is given a score of 100. <score> must an integer between 1 to 100.

For example, to subscribe for articles related to "information filtering," valid for 200 days, send this:

```
=====
% mail netnews@db.stanford.edu
Subject:
subscribe information filtering
expire 200
=====
```

The service will acknowledge your subscription, returning a subscription identifier (sid).

( C . 2 ) G e t t i n g A r t i c l e s

After receiving notifications of articles that may be relevant to your interests, you may decide to see an article in its entirety. You can get the whole article with the GET command:

```
GET article article ...      Get the articles specified (by their article
                               ids).
```

For example,

```
=====
% mail netnews@db.stanford.edu
Subject:
get news.announce.conferences.3670
=====
```

( C . 3 ) R e l e v a n c e F e e d b a c k

After reading the articles, you may find some that you like. You can provide feedback using these commands:

FEEDBACK sid                                    Provide feedback to subscription <sid>.

LIKE article article ...                    Specify relevant article(s) by their ids.

For example, this message says that article news.announce.conferences.3670 is relevant to subscription 1:

=====

% mail netnews@db.stanford.edu

Subject:

feedback 1

like news.announce.conferences.3670

=====

With feedback information, the service may be able to better match future articles against your subscriptions.

#### ( C . 4 ) M a n a g i n g S u b s c r i p t i o n s

You can manage your subscriptions with these commands:

UPDATE sid                                    Update subscription with id <sid>. Must be followed by one or more of PERIOD, EXPIRE, THRESHOLD (see C.1.), or PROFILE commands to specify the parameter(s) to be updated.

PROFILE word word ...                    Specify the new profile for the UPDATE command.

CANCEL sid                                    Cancel subscription <sid>.

LIST                                            List all your subscriptions.

For example, to update the period and the threshold of a subscription:

=====

% mail netnews@db.stanford.edu

Subject:

update 3

period 1

threshold 60

=====

( C . 5 ) S e a r c h F o r P a s t A r t i c l e s

Besides providing the subscription service, the service also allows you to search for recent articles that are already in the database:

SEARCH word word ... Do a search of the database with the given query. Maybe optionally followed by a THRESHOLD command to specify the minimum score for an article to be retrieved.

For example, to search for articles related to "information filtering":

=====

% mail netnews@db.stanford.edu

Subject:

search information filtering

=====

( C . 6 ) O t h e r C o m m a n d s

END End the request message. Useful for preventing the processing of signatures.

HELP Get help information on server.

( D ) R E F E R E N C E S

[1] B. Kahle, et al., "An Information System for Corporate Users: Wide Area Information Servers," ONLINE, Vol. 15, No. 5, p. 56-60, September 1991.

Your subscription to this free service will be for over 27 years (9,999 days) by default. You can cancel a subscription at any time, or set the expiration to a shorter period if you like. But, hey, its free!

## Public Education and the Internet

As we near the close of this chapter, we will first provide a starting point for the youngest Net users, and then review several ways of organizing the many uses schools are now making of the Net. Finally, a few glimpses of the Net's educational potential will be suggested.

### When Should the Internet Enter the Classroom?

As usual, this depends on your own comfort level, and on the equipment and support available. Watching adults learn is very helpful for kids, so you shouldn't necessarily wait until you're a guru to start using the Net in school. It is very comforting for some children to see an adult being thwarted by this new keyboard-thing in the same way that they are.

The least expensive alternative starting place is e-mail. Color is not required on the computer monitor screen, and practically all Internet service providers have some kind of e-mail. And *even the youngest children*, preschoolers and kindergartners, can learn lots of things from e-mail:

- *Geography*. Mapping the locations of e-mail friends.
- *Language Arts*. What words do people use to describe themselves? How are formal and informal speech different? What happens if the keys aren't pressed in the "right" order?
- *Communication*. What parts of the message are vital to understanding it? What parts are required for it to be sent and received?
- *Cultural diversity*. What holidays are celebrated by the remote person(s)? And what are their favorite foods and what hobbies do they enjoy?

As the target audiences grow older, and more sophisticated, the whole Net opens to their keyboards. Teachers around the world are using this new communication medium to nurture learning in every conceivable curriculum area.

### Forms of Education on the Internet

Now let's explore the ways that students, parents, and teachers can use the Net to improve and enhance learning. Here are some of the dozens of ways people are now using the Internet for learning in schools, homes, and workplaces.

Only a few options are inherent in the machine-human combination:

| <i>Communication</i>       | <i>Net Tool or Activity</i>                        |
|----------------------------|----------------------------------------------------|
| Person to person           | e-mail, Internet Relay Chat (IRC)                  |
| Person to many people      | e-mail, Listserv, newsgroups                       |
| Many people to many people | e-mail, Listserv, newsgroups, IRC, MOO and MUSE    |
| Person to computer         | telnet, FTP, Gopher, WWW                           |
| Person to many computers   | Veronica, Gopher, WWW                              |
| Computer to person         | Personal news services, beeper on command services |
| Computer to many people    | Custom news services                               |
| Computer to computer       | Indexing, updating, and Net monitoring             |

How these combinations and permutations manifest with various Internet tools is fascinating to behold. Next are three ways to organize the educational uses of the Internet.

This first classification method groups Net tools with different combinations of students, teachers, and behavioral curriculum goals:

Professional collaborations among educators

Students' collaborative investigations

Students' and teachers' access to scientific expertise

Students' and teachers' access to information (libraries, and so on)

Students' and teachers' access to computers and software

Collaborative development and delivery of instruction and materials

Teacher education and enhancement

Electronic publishing of students' products

Next is a second practical method of organizing the options, from a different perspective.

#### Professional Activities

Research

Collaboration, sharing, exchange of information

Overcoming space and time to bring together groups with common goals

Around the school district or county

Around the region, country, or world

#### Information Access

Catalogs, databases, archives

News services

Software

#### Text-Based Virtual Reality

Muds and MUSES with carefully defined content, such as:

- Detroit—Factory town in the 1960s
- The Siege of the Alamo, or of Vicksburg
- Mountain meadow ecosystems
- Writing and signing of the Magna Carta
- Historical development of bridge construction
- Discovery of planet Mars, DNA, or continental drift
- Life in a frontier outpost
- Designing a media campaign to promote something patently worthless (garbage, nuclear waste, and so on)
- Congressional or Parliamentary functions

And here is a final method of organizing the types of uses being made of the Net, particularly in K-12 classrooms.

## E-mail Contacts

Electronic visitors

Electronic mentoring

Global classrooms

Keypals

## Making Collections

Growing databases

Pooled data projects

Remote field trips

Electronic publishing

## Projects

Action projects

Asynchronous writing

Information searches

Parallel problem solving

Sequential creations

Simulations



**Tip:** If you need some quick ideas on what to do in class, or some sources of lesson plans on the Net, get the new book *Education and the Internet*, by Dr. Jill Ellsworth, and look in Chapter 2, "Internet Resources by Curricular Area." Choose a topic area you or your kids would be interested in, and several resources will be at your fingertips!

---

## The Internet's Educational Potential

For decades there have been outcries and reform movements about the quality of "free" public education in the U.S. But the problems of classroom educators have *not* been solved, even though dozens of popular attempts have been made to reform public education.

Universal access to the Internet could provide the final panacea for education, *if* it is truly universal, affordable, and if Internet training and technical support are easily available to all.

In August 1992, then United States Senator Al Gore said "In the past few years, we have witnessed the democratization of the Internet. Today, the network connects not only to the top research laboratories and universities but also small colleges, small businesses, libraries, and high schools throughout the country."—quoted by Tracy LaQuey in *The Internet Companion* (Reading, MA: Addison-Wesley, 1992, p. vi).

As we have already seen, *elementary* and *middle schools* have also now joined the writhing throng on the Net. Why are educators at *all grade levels* so hopeful about the Internet?

It may be that a paradigm shift is occurring in the way our society deals with information and learning. With limitations of time and space swept away by open, affordable access to the Internet, anybody can learn whatever they want, wherever they want to learn it, at any time of day. Telephone companies installed over 2,300,000 miles of fiber optic cable in 1993 in order to prepare for their role in delivery of Internet quality services throughout the U.S. These cables can change our schools and our lives.

U.S. hospitals have seen many of their services rapidly shift in the past few years to out-patient clinics. So also we may soon

see colleges, high schools, and middle schools being less important as the sites of education. Is there a fundamental reason why all learning must occur with several students and a single teacher in a particular room all at the same time? Gurus say no, there is not.

There is also no reason why all students must experience the exact same presentations of new ideas, or the exact same drills and practice. Why not have advanced software to keep track of each student's learning style and adjust the pace, sequence, and level of detail for that one precious student?

This use of computers would usurp some of the teacher's role in the current educational system. What's left for the teacher to do?

Teachers will find that their role does change dramatically with increased use of the Internet for learning. The best teachers will be those who are *themselves* good at learning and at guiding others to the right next bit of learning. Teachers, too, will need to be more concerned with individual learning styles, and much less concerned about classroom control. They will need to concentrate much more heavily on asking the right questions, rather than being the *source* of the right answers!

Students quickly discern which teachers are aware of and involved with the new technology. Teachers who do not adopt the new may become increasingly obsolete and unpopular. Librarians, on the other hand, are likely to rise in power and respect simply because of their advanced skills at information processing, search, and retrieval. Because many perceptive librarians have embraced the Internet and are already revered among its greatest gurus, their profession will likely return to, and perhaps exceed, its former esteemed grandeur.

This may be the last, best opportunity this century to really accomplish a major step forward for broad public education. If the National Information Infrastructure (NII) fulfills its promise during our lifetimes, *every* literate person can have lifelong learning and ongoing personal development as a matter of policy, not as a *one-time* dose of education or training.

Because the Internet frees us from the bonds of time and space, we can all use educational materials at the Ivy League schools, or at the top research laboratories, without even "graduating" from a high school, much less passing the entrance requirements and paying the fees of an Ivy League college.

Knowledge, experience, and personal skills may become more important in the workplace than credentials like graduations and advanced degrees. Access to advanced technical information can be at the fingertip control of every individual human seeking knowledge, rather than being controlled by a "sage on the stage," which characterizes the old educational regime.

Now we close this chapter with three short quotes from a great leader and philosopher, the third President of the United States. The framers of the National Information Infrastructure (NII) would do well to listen carefully and heed his wise counsel.

"By far the most important bill in our whole code, is that for the diffusion of knowledge among the people. No other sure foundation can be devised for the preservation of freedom and happiness. If anybody thinks that kings, nobles, priests are good conservators of the public happiness, send [that person to Europe.]"

Thomas Jefferson (*Writings*, v. 5, p. 394)

"If a nation expects to be ignorant and free, in a state of civilization, it expects what never was and never will be."

Thomas Jefferson (*Letter to Col. Yancey*, 1816)

"I know of no safe depository of the ultimate power of the society but the people themselves; and if we think them not enlightened enough to exercise their control with a wholesome discretion, the remedy is not to take it from them, but to inform their discretion by education."

Thomas Jefferson (*Letter to William C. Jarvis*, Sept. 28, 1820)





- [20](#)
  - [Business and Commerce](#)
    - [Demographics](#)
      - [Results From The First World Wide Web User Survey](#)
      - [Internet Info](#)
      - [The Internet Business Center](#)
    - [The Internet Business Advantage](#)
    - [What About Security?](#)
    - [Business Already on the Internet](#)
      - [Internet Shopping Network](#)
      - [Kaleidospace](#)
      - [Hewlett Packard](#)
      - [Grant's Florist](#)
      - [CommerceNet](#)
    - [A Guide for Getting Your Business onto the Net](#)
      - [Electronic Mail](#)
      - [Putting Up A Web Server](#)
      - [Putting This All Together](#)
    - [You Can Put Electronic Commerce to Work for You Immediately!](#)

## 20

# Business and Commerce

--by [John Katsaros](#)

The economic worldview is one in which all human interactions can be described in terms of economics—in the supply-and-demand terms of commerce. It was inevitable that the electronic subculture of the Internet would become the base of a dynamic economy, in which legitimate business is carried out every day. Not unexpectedly, such transactions have, in some circles, become as pedestrian as buying a pretzel from a street vendor. Like all markets, though, the Internet has a unique character that must be understood to be exploited.

## Demographics

Electronic markets are similar to all other markets in one key respect—the winners understand their audience. Doing business on the Internet requires an understanding of who is "out there" and what they need. In the parlance of the marketer, what are the demographics? Of course, with a population as large as the Internet, there is no single answer to this question. (The broadest community today is 25 to 50 year old males with technical degrees, although this is becoming less true each day.) Getting the raw information yourself can be cost-prohibitive. Fortunately, several universities and businesses are asking the right questions and may be able to help you better understand usage characteristics as they affect you.

### **Results From The First World Wide Web User Survey**

Leading the list of Internet usage information is the *Results from the First World Wide Web User Survey*, prepared by the Graphics, Visualization, & Usability Center of the College of Computing at Georgia Institute of Technology. In January, 1994, James Pitkow and Mimi Recker received results from over 4,000 Internet users. Their questions included primary computer platform, hours spent in front of a computer, age, gender (94.9 percent male) and occupation. This study is a good first step toward improving knowledge of who is on the Internet. They plan to run this form of study twice each year. Future studies will include several valuable additions: 1) adaptive questions so that, as the user answers a questions, follow-up questions are selected that relate specifically to the user's environment; and 2) a results database, so custom queries may be produced from study data. You can view the results of the first study in HTML via: <http://www.gatech.edu/pitkow/survey/survey-1-1994/survey-paper.html>.

Figure 20.1 shows the percentage of Internet users by age group.

### [Figure 20.1. Internet users by age group. From Results from the First World Wide Web User Survey.](#)

## Internet Info

Another interesting source of Internet usage information is available from Internet Info. They continuously look at who is on the Internet from a commercial standpoint. In addition to doing custom reports, Internet Info also provides free lists of companies registered on the Internet segmented by industry type and geography. Their free New to the Net service provides a monthly listing of some of the more interesting new commercial domains and their business owners. The report also tracks the monthly growth rate of commercial domains. You can reach Internet Info at [info@internetinfo.com](mailto:info@internetinfo.com). You can view their free service information in HTML via <http://tig.com/IBC/>. The following is a sample report from Internet Info.

Cities with the Highest Concentration of InterNet Connected Companies

April 1994

Connected

Internet

Population

Companies

|                       | Connected | in      | per   |
|-----------------------|-----------|---------|-------|
| City / State          | Companies | 1,000's | 1,000 |
| 1 Princeton NJ        | 39        | 12      | 3.25  |
| 2 Palo Alto CA        | 155       | 55      | 2.82  |
| 3 Menlo Park CA       | 77        | 28      | 2.75  |
| 4 Falls Church VA     | 23        | 9       | 2.56  |
| 5 Scotts Valley CA    | 20        | 8       | 2.50  |
| 6 Vienna VA           | 33        | 14      | 2.36  |
| 7 Boulder CO          | 194       | 83      | 2.34  |
| 8 Herndon VA          | 34        | 16      | 2.13  |
| 9 Mountain View CA    | 143       | 67      | 2.13  |
| 10 Golden CO          | 25        | 13      | 1.92  |
| 11 Santa Clara CA     | 175       | 93      | 1.88  |
| 12 Los Altos CA       | 47        | 26      | 1.81  |
| 13 Burlington MA      | 39        | 23      | 1.70  |
| 14 Los Gatos CA       | 45        | 27      | 1.67  |
| 15 Cambridge MA       | 159       | 95      | 1.67  |
| 16 Cupertino CA       | 66        | 40      | 1.65  |
| 17 Fairfax VA         | 29        | 19      | 1.53  |
| 18 Englewood CO       | 40        | 29      | 1.38  |
| 19 Sunnyvale CA       | 157       | 117     | 1.34  |
| 20 McLean VA          | 28        | 24      | 1.17  |
| 21 Los Altos Hills CA | 28        | 26      | 1.08  |

|    |              |    |    |      |
|----|--------------|----|----|------|
| 22 | Littleton CO | 34 | 33 | 1.03 |
| 23 | Sterling VA  | 19 | 20 | 0.95 |
| 24 | Beaverton OR | 50 | 53 | 0.94 |
| 25 | Reston VA    | 44 | 48 | 0.92 |

This analysis is based on the 14,276 companies registered with Internet as of April 1994.

Source: Internet Info, Falls Church, VA For more information send email to [info@internetinfo.com](mailto:info@internetinfo.com)

## The Internet Business Center

The Internet Business Center (IBC) provides information sources that help people conduct business on the Internet. IBC's home page is a good starting point to find commerce activity as well as other sources of marketing information concerning the Internet. The Internet Business Center is sponsored as a marketing service by The Internet Group, an Internet consulting firm. You can view the Internet Business Center in HTML via: <http://tig.com/>. Figure 20.2 shows the IBC home page.

### [Figure 20.2. The IBC home page.](#)

## The Internet Business Advantage

As Internet commerce activity increases, more and more information will become available to help you increase your business. A recent publication, The Internet Business Advantage, is a monthly newsletter devoted exclusively to helping businesses increase their success on the Internet. In addition to covering new Internet commerce related activity, the Internet Business Advantage runs special features providing insights and tips to help businesses use the Internet. In a recent feature covering advertising over the Internet, the Internet Business Advantage provided tips and techniques to help businesses improve their exposure. You can reach the Internet Business Advantage at [success@wentworth.com](mailto:success@wentworth.com).

## What About Security?

One of the most popular misconceptions about putting a business on the Internet is that as soon as you do so, computer and communications phreaks around the world will immediately try and bring down your network and, by extension, your business. Although like most myths, this is far from accurate, it indicates an important truth: One (if not the primary) concern for the Internet business person must be security.

What does that mean? Does security on the Internet mean physical security of the host and communications hardware? Does it mean being able to track all attempts, unsuccessful and successful, to

use resources on your business LAN or WAN? Does it mean a constant cycle of education for you as well as your peers, subordinates and superiors about how to "keep their end up" in the security obligations of your company? Yes—all that and more.

Question: How do you make my business network impenetrably secure?

Answer: You can't.

Short of powering down all your computers, sealing them all in a block of lead and shooting them into the sun, there's nothing you can do to make your network impervious to attack by someone really intent on doing so and well-enough armed to pull it off. Even if you're not on the Internet, many conditions—the radio-frequency interference from your displays; traffic broadcast on your network through unsecure spaces; old backups, floppies, tapes, and printouts in the dumpster; undercompensated but highly privileged users—can all be easily used to get sensitive information about you and your customers almost as easily as if you'd published it in the classifieds.

Does this mean you should hop in your car, drive to work, turn everything off, and start using index cards, green paper ledgers and Big Chief tablets? Not at all. Host- and network-based information can be the least secure information you have, but it can also be the most secure—certainly more secure than something sitting in a physical filing cabinet or on a standalone workstation. The difference is in where you place your installation on the continuum between a completely open and public system, and one in which no one—not even legitimate users—can get at sensitive information.

Very few, if any, users will find either pole of that spectrum acceptable. Persons who would try unauthorized penetrations of your security will find it rewarding only up to a certain point. The art of security risk management is in locating the point on the spectrum that is appropriate for you, your customers, and your company.

There are too many approaches to site security to summarize them here, but a few would make an interesting starting point on your journey toward a "secure-enough-for-now" site.

Much like the firewall in your car, site firewalls exist to partition you from the chaos outside. In your car, the firewall is just that—a wall to separate the passenger compartment from the engine compartment in the event of a fire. A site-security firewall exists to partition the relatively unsecure territory outside from the (hopefully) relatively secure territory within your site. It's useful to know about three major types of site firewalls: *router-based*, *host-based*, and *application-based* firewalls. Nowadays, hosts can be found performing router-type firewalling and so on, so these should be considered broad categories.

Router-based firewalls take advantage of the fact that all data communications coming to and from your site goes through a few routers. It makes sense, therefore, to take advantage of this fact to filter the traffic going into and out of your site, and instruct the router to prohibit traffic from certain networks or to certain types of applications. The advantage of this approach is that it frequently doesn't require any additional purchases—just the additional manpower and administrative commitment necessary to decide what kinds of filters should be in place, and how to implement them. Not all routers are able to do such filtering, though, and a prohibitively large investment in routers incompatible with this approach might be a good reason for going with another approach. Other reasons for not using router-based firewalls include the increased expertise required to maintain security, as well as usability when using a dynamic set of filters in your router configurations. For a network of any substantial size, the processing and memory load represented by doing such work on your main routers may also be prohibitively large. Certainly, a

measure of quantitative testing is called for when deciding if router-based firewalls are appropriate for you.

Another, often simpler, approach is a host-based firewall. This approach is popular with many companies because it involves directly connecting only one host to the Internet, loading it up with additional security safeguards, and requiring all internal and external users to log in to the firewall host before gaining access to the other side. The benefits of this include the fact that it's as close to a no-brainer as security ever gets, as well as the fact that it greatly reduces the number of points of possible entry to your system, permitting you to spend more resources on securing other access points such as dialups, interorganizational WAN links, and remote LANs. The impact of the downside falls mainly on your corporate culture. Making the Internet difficult or even inconvenient to use will discourage a large percentage of your organization from making it part of their daily routine. If you're intent on making the Internet part of your corporate culture, this discouragement may translate into a larger and less effective training budget to compensate for the learning that would take place in everyday Net-surfing.

The third type of firewall, application-specific firewalls generally "live" on each individual host they protect, directing their influence toward a specific set of problems associated with individual applications. The advantage of an application-specific firewall is that it affords single hosts much more security than exists in the surrounding LAN. It also permits over-extended support organizations to bring up security in an incremental fashion, starting with the most critical hosts, and working their way down a chain of relative importance. A disadvantage of application-specific firewalls is that although they might require a small-to-moderate amount of effort to install on any given host, that effort and resources can easily become quite large when multiplied by every host in the organization. Many sites prefer to use application-specific firewalls as a secondary measure to shore up the influence of sitewide firewalls that add protection to the entire site.

Numerous applications in common use have their own individual security concerns. The UNIX "r" utilities—rlogin, rsh and rcp, for instance—can represent a wholesale security compromise if an innocent user happens to put the wrong information in a .rhosts file in their home directory. HTTPD, or a World Wide Web server, can permit users to publish form- or script-based services on the WWW without realizing how they can be exploited by remote users to gain varying levels of access to your systems. SMTP-based mail systems, perhaps the most common Internet application, can be misconfigured and subsequently misused as well to diminish your efforts to secure your business network.

It's easy, after learning such things, to conclude that security concerns make the Internet much more trouble than it's worth. With a little forethought and an ongoing commitment, though, you should be able to find a happy medium with regard to security.

With a well-maintained site security system, using network resources outside your corporate network should be no more difficult than using resources within it. Legitimate outside users shouldn't have to jump through hoops to get out what they need. You should be able to provide your customers with peace of mind that confidential information about them has not been compromised. The only way to do this is to monitor all your networks and host systems with constant vigilance, and have trustworthy tools and personnel keeping records about all entrances into your hosts and networks, and from where the entrances originate; have a common understanding in your organization that security is *everyone's* job; and finally, think through every likely and many unlikely security contingencies that you could be faced with at the most inconvenient times, and prepare for them so that your response and that of your organization will be second-nature.

In short, the security of the Internet and Internet-attached systems is a constant arms race between legitimate users or providers, and persons who (just as often well-intentioned as not) see it as their sacred duty to expose all security weaknesses on the Net in an endless game of oneupmanship between themselves and the rest of the world. As the business community begins to offer more mission-critical services over the Net, such comparatively innocuous irritants may be replaced by the more serious threat of industrial espionage and trade warfare.

The good news is that unless you do something terribly naive, like making public claims that your network is bulletproof or challenging computer or communications phreaks to break into your network (although comparable challenges made to your own employees are often an excellent way to test your security), each successive measure of security you place on your network puts you that much further away from disaster.

Although it may sound Machiavellian, the fact that there are always businesses on the Net who aren't responsible about their own security will always work in your favor if you, by contrast, expend the effort to make security a central component of your service mission. If a car thief, after all, encounters two similar cars in a darkened, isolated parking lot, but only one is alarmed, guess which one won't be there in five minutes?

## **Business Already on the Internet**

Probably the best way to see how your business may be able to use the Internet is to see how others are using it. Here are some examples.

### **Internet Shopping Network**

"Welcome to the toy store of the global village. It used to be G.I. Joes. Train sets. Hot Wheels. Now you prefer to play with the hottest hardware and software around." With this introduction, the Internet Shopping Network's (ISN) brochure pronounces that it is ready to do business. Founded in June, 1993, the Internet Shopping Network is an online microcomputer software and hardware superstore available 24 hours per day, seven days a week, to anyone with access to the Internet.

Clearly, ISN has "set its sites" on the gold-collar worker of today—the Internet user. That is not such a bad idea. Think of the demographics that you already know about today's Internet users. Without knowing their ages, race, religion, or where they live, you do know that they own and use computers. They at least have a modem or another, faster, way to connect to the Net. Many represent the elite of the computer industry. They drive the fastest computers, influence lots of other computer decision makers and often design the next generation of products. ISN is targeting this group as the primary customers for its computer cyberstore.

The Internet Shopping Network is set up as a member-oriented organization; membership is free with an approved Visa or MasterCard. ISN offers more than 15,000 software and hardware products. In addition, the Internet Shopping Network has begun offering access to the past twelve months of *InfoWorld* articles. Their rationale behind *InfoWorld* access is to enable shoppers to browse past issues to aid selection of the products most likely to fit their needs. If you want to buy a disk drive, for example, you first go to *Infoworld* and get their reviews to find which products work the best and why. Then you browse Internet Shopping Network's catalog and make your purchase.

The "toy store of the global village" provides an interesting glimpse of electronic commerce over the Internet. The Internet Shopping Network can claim two distinct advantages for cyberstore retailing. First, it is possible to get lots of "browsers" with an extremely low cost per browser. ISN doesn't have to worry about location, shoplifting, heating, or parking. From their Web server, they are able to support thousands of people from all over the world browsing through their store each week. That is a retailer's dream. When it is time to order product, the customer selects the item and initiates the transaction. The cost per sale transaction is incredibly low—much lower than the \$5 order cost often cited by 800 telemarketer sellers.

By using a membership orientation, the Internet Shopping Network avoids the security problem that comes when someone places their credit card number onto a public network. The card number never appears on the network. The one-time membership registration is handled through their 800 number or via FAX. This way, the Internet Shopping Network is able to offer credit card security on par with other mail-order houses and telemarketing sales groups. Figures 20.3 and 20.4 show the ISN home page and the ISN second page, respectively.

### [Figure 20.3. The ISN home page.](#)

### [Figure 20.4. The ISN second page.](#)

The Internet Shopping Network wants to become everyone's computer superstore. People in areas far from the nearest superstore are able to have the same broad product selection and low prices offered by superstores. Plus, the Internet Shopping Network offers a home for hard-to-find items not normally available on the shelves of major retailers.

## **Kaleidospace**

Billing itself as "the world's most accessible art gallery," Kaleidoscope Media displays works of aspiring artists to Internet users worldwide. Kaleidoscope's gallery, cleverly named Kaleidospace, is open 24 hours a day and is available worldwide for browsing the latest works from member artists. Artists can display their works for \$50 per month (plus sales commissions).

Kaleidospace is divided into ten rooms:

1. *Art Studio*. An art gallery for people selling graphic and fine art. Small, independent art galleries can have their own showcase.
2. *Center Stage*. For performers (comedians, performance artists, and so on) to promote themselves while on tour (using performance film clips). Performers use this section to showcase themselves for future bookings, selling performance videos or letting people know where they are scheduled. Independent clubs also may use this section to showcase their upcoming performances.
3. *Cyberfaire*. This room is for artisans and crafts people making "made-to-order products." These artisans can showcase samples of their work (from custom-made dragon sculptures to joke writing) and to get from perspective clients information necessary to create custom-made art.
4. *Interactive Arena*. This is for showcasing interactive titles (games and CD-ROMs).

5. *Music Kiosk*. Independent musicians can showcase their CDs and tapes. Independent, small record labels also have their own pages in the Music Kiosk.
6. *Newsstand*. This is for announcements and news relevant to the entertainment industry.
7. *Reading Room*. Authors and small publishing companies can participate in this room. They can showcase their material for publication or their finished work for end users.
8. *Screening Room*. Independent film makers (as well as animation and music video producers) can showcase their videos.
9. *Spotlight*. Spotlight houses high-profile artists to showcase a series of their work. Science-fiction writer David Brin has a novel that he started, which will be finished by Kaleidospace's audience; illustrator P. Craig Russell has the first three panels of a comic that he began, which will also be finished by the audience.
10. *Tool Shop*. For tools where the artists are the customers. CD-ROM kits, multimedia tool kits, independent software developers with tools, and sound archives are showcased in this room.

Figure 20.5 shows the Kaleidospace home page.

### [Figure 20.5. The Kaleidospace home page.](#)

Kaleidospace supports artists in ways that traditional art galleries cannot. Since they are dealing in Cyberspace, promoting, screening, distribution and placement capability take on new meaning. For instance, when you go shopping for a new CD to add to your collection, Kaleidospace lets you download a 30-second sample selection, browse through reviewers' comments, and look through the artists' online biographies. The service seems to be off to a good start. Within the first few weeks of its introduction, Kaleidospace had over 40,000 log-ins from around the world.

## **Hewlett Packard**

Using the Internet to improve customer service seems to be the most widespread among computer companies. It is only natural that these organizations turn to the Internet since that is where their customers are. There already is a long list of companies, their World Wide Web servers loaded with customer support information.

In April, 1994, Hewlett Packard went online with Access HP for the express purpose of delivering customer information using the Internet. Hewlett Packard's philosophy is to deliver information to customers where and how their customers want it—at the right time and at the right place. With so many of their customers already on the Internet, Access HP, is a natural extension of their service/support mechanism.

Hewlett Packard found the Internet very appealing. It has a low entry cost; HP already had access to the Internet, Mosaic is free to the users, and the technical community truly has an interest in doing this. So far, its early work has been rewarded with customer enthusiasm.

Figure 20.6 shows the HP home page.

### **[Figure 20.6. The HP home page.](#)**

Hewlett Packard has many lines of business—computers, peripherals, medical products, test and measurement equipment—each with its own special needs. Access HP has been constructed so that each business division, as well as HP Corporate, can identify and fill customer needs through the Internet. Whether it is business-to-business communication, support for product catalogs, locating training courses, updating customer software, or providing technical support, Access HP can be used. Hewlett Packard is planning to make this service a mainstream part of their overall marketing, sales, and customer-support capability.

### **Grant's Florist**

Need a dozen roses? Instead of looking in the phone book for a florist, Grant's Florist is available on the Net 24 hours per day to serve you. You can go over to their home page, view images of their arrangements, make your selection (they list their arrangements by price categories) and place your order. Of course, your personal message is placed on the greeting card. All without having to leave the comfort of your desktop.

The feature I like best from Grant's Florist is their reminder service that comes through e-mail. Since I check e-mail every day, I expect this service will come in handy for those hard-to-remember occasions. This can lead to a whole host of Internet-enhanced personal services. In addition to reminding you of birthdays and anniversaries, businesses can alert you that your car needs scheduled service, that it is time for your annual teeth cleaning, and so on. Perhaps we will be getting college reunions set up through Internet mail—assuming your Internet address does not change for fifty years!

Figure 20.7 shows the Grant's Florist home page.

### **[Figure 20.7. The Grant's Florist home page.](#)**

### **CommerceNet**

CommerceNet's ambitious efforts take aim at creating the first large-scale market trial of electronic commerce on the Internet. CommerceNet is a nonprofit consortium of Northern California, technology-oriented companies whose goal is to create an electronic marketplace where companies transact business spontaneously over the Internet. In April, 1994, CommerceNet launched its World Wide Web server to provide users access to electronic commerce-related information, applications and organizations. The CommerceNet server hosts information in six primary categories: 1) information about the CommerceNet organization; 2) directories of participants, value-added third-party services, and Internet resources; 3) member registration and communications; and 4) tutorials and examples.

CommerceNet's directors feel it will be successful when its member companies are more successful

because they are on the Internet. For the present, CommerceNet is organizing member initiatives to look into expanding Internet functionality to simplify commerce activity. They have chosen security, directories, catalogs, payment methods, and simplified access as chief among Internet electronic commerce concerns. CommerceNet has formed several committees of its company members to develop methods around these topics: 1) Connectivity Working Group, 2) Network Services Working Group, 3) Payment Group, 4) Directory Group, 5) Electronic Catalog Group, 6) Internet EDI Group, and 7) Design and Manufacturing Integration.

CommerceNet was created and is operated by a consortium of major Silicon Valley users, providers, and developers. It seeks to revolutionize the Valley's core electronics, software, and information service industries by making interactions between customers, suppliers, and development partners as efficient as interactions among internal departments. CommerceNet will ultimately help to revolutionize the way most Bay Area companies transact business, regardless of their size of business focus.

Membership in CommerceNet is open to all businesses. Its original participant list reads like a who's who in Silicon Valley: Amdahl Corporation, Anthem Electronics, Inc., Apple Computer, Inc., AVEX Electronics Inc., Bank of America, Citibank N.A., Dataquest, Digital Equipment Corporation, Dun & Bradstreet Corporation, Electronic Marketplace Systems, Inc. (An International Data Group Company), Hewlett-Packard, Integrated Systems Solutions Corporation (a subsidiary of IBM), Intel Corporation, Internet Shopping Network, InterNex Information Services, Inc., Lockheed Missiles and Space Company, Inc., Nanothinc, National Semiconductor, Network Computing Devices, Inc. (NCD), Pacific Bell, PartNet, Inc., RSA Data Security, Inc., Solectron Corporation, Sun Microsystems, Inc., Synopsys Inc., Tandem Computers Incorporated, Trusted Information Systems, Inc., Wells Fargo & Co. and Xerox. This membership list is a cross between computer companies, software companies, communications companies, semiconductor manufacturers and services organizations. CommerceNet also has government affiliations including regional, state and federal agencies. CommerceNet was awarded a \$6 million matching grant over three years by the United States government's Technology Reinvestment Project (TRP).

Figure 20.8 shows the CommerceNet home page.

### [Figure 20.8. The CommerceNet home page.](#)

CommerceNet believes it can best benefit business by dramatically improving the productivity and competitiveness of its participants. Through an online global marketplace it can help its company members 1) shorten procurement cycles through online catalogs, ordering, and payment, 2) cut costs through competitive bidding, and 3) shrink development cycles and accelerate time-to-market through collaborative engineering and product implementation. CommerceNet's directors feel it offers an opportunity to build Northern California's information infrastructure, to influence the development of Internet technology and standards for electronic commerce, and to participate in joint marketing efforts.

The CommerceNet consortium wants to remove some of the barriers to electronic commerce over the Internet. They are already displaying a secure version of Mosaic, which uses public key cryptography for either securing information or digitally signing documents. CommerceNet is talking about providing a starter package to simplify the process involved with creating a World Wide Web server for a business. Under the title of Electronic Industry Pilot, CommerceNet members involved with electronics manufacturing are experimenting with creating an electronic web that will improve electronics

manufacturing processes.

---



## Real Work in Virtual Space

--by [Jon Callas](#) and [Margaret Russell-Gonzalez](#)

The Internet is important more because there are other people there than because there is a lot of data on the Net. The data could just as easily be sent to you on CD-ROM. What makes the Internet the Internet is that it is alive, it is immediate, and communities form and thrive in it. These communities are both valuable and important, and require a different look at how we use and build technology for the Internet.

There are a number of programs and systems to do work on the Internet. Among these are a class of programs that are communications programs. For example, mail, news, and IRC are all communications programs, built for no other purpose than letting people talk to each other. There is an overlapping class of communications programs that enable the people who use them, sometimes incidentally, sometimes by design, to form their own communities. If you're doing business on the Internet, you probably call these communities *teams*.

Meeting Space[tm], made by World Benders, Inc., is designed for people in different places to communicate and collaborate at the same time—to form teams. Business gurus from Tom DeMarco to Frank Sonnenberg tell us that teams function best when all team members work within a small area; when the distance between people grows beyond thirty yards, team communications break down and so does the team.

Often we don't have the luxury of getting everyone we need to work with within thirty yards of us. Networks like the Internet not only that us to work with people who are distant but, by their very existence and utility, also transform the ability into the imperative to work with people who *are* distant. The human need to be close to people is still there, though. Physical and psychological distances can be bridged by creating a virtual space—a place to work that provides communications, which create the comforting and necessary illusion of proximity. Meeting Space exploits the advantages of networks like the Internet to the fullest; it is software that is not only technology, but also sociology.

---



## An Overview of Meeting Space

Unlike most business tools, Meeting Space found its origins in a class of toy. These toys—called muds, MOOs, and MUCKs—were created by college students so that they could socialize and play games with other college students all around the world. Meeting Space grew from that playground. We have taken technology from its playground and developed it into a real tool for collaborative work. It was a natural growth, because much the same things that make collaborative play effective can also be turned into an effective tool for collaborative work.

Some of the necessary development work was easy and obvious. "Wizards" become "administrators." "Players" become "people." "Dungeons" become "office suites." We took the ugly but utilitarian text-only

user interface and added menus, icons, and pictures. We also solved the common embarrassment of whispering quietly into the wrong person's ear.

Our most arduous task was the one least visible. The muds used by college students run on UNIX systems. They are relatively expensive machines that are cranky to run and keep running. The real problem we solved is how to take a distributed social environment and make it so that it is easy to use, easy to manage, and requires little upkeep. Meeting Space runs on personal computers, not workstations, over mixed network protocols and mixed platforms.

Meeting Space consists of a server program running on one computer and a number of clients. The server need not be on a computer devoted to running it, but many people do so anyway, because it runs well on an older machine that was formerly gathering dust. A Meeting Space that supports a small work group can run with 1MB of disk space and 1MB of memory. The server can also act as a document library for the group. As of this writing, the server must be an Apple Macintosh II (or later) running System 7—but we are also working on a version of the server that runs as a Novell NLM, and Microsoft Windows NT. The server works on any AppleTalk, Novell, or TCP/IP network, and can support clients using all three networks simultaneously. The network protocols are robust and terse; if, for example, you are connected to the Internet via a PPP or SLIP line running on a 14.4KBd modem, you can support up to 40 people on your network connection before you risk saturating the line.

The client program can run on a Macintosh or an IBM compatible running Microsoft Windows. We have a client program under development for the Newton PDA, and we have plans to build a client and server that run on UNIX systems but have yet to begin development. Clients connect to meeting sessions on a server and can, in fact, connect to more than one meeting or server at a time through different sessions on different windows. Connections can be saved and reused, making the transition into virtual space easy. Once connected, the software is robust and handles network glitches so that what was, for example, an accidental disconnect is seen by the user as a mere pause.

At its core, Meeting Space is a small operating system. This is a key feature of the program, and our best trick. As an operating system rather than an application, Meeting Space has the framework to do more than one thing at a time, and provides means for managing and controlling resources such as memory, disk space, and network connections. An operation that takes a long time (say, more than a second), is interleaved with other things that users of Meeting Space are doing. Thus, you can transfer a file at the same time that you are watching and discussing a slide presentation.

Meeting Space is also an object-oriented database. All the items that you work with in Meeting Space are objects—from the offices you work in, to the documents you work with, to your own virtual self. Objects can inherit from other objects. This inheritance simplifies working with them. We make administration easy by creating collections of objects where this is appropriate. For example, when you create a "person" object in Meeting Space, you also create a space for that person to work in, a room for them to use as their office, and their own shelf for documents. We simplify expanding the space by creating doors when we create rooms. Thus, you have only to walk to where you want a new wing and select the menu item to create a room. Meeting Space itself will make the room object, complete with doors into it and out of it. This may sound laughably obvious, but it is a feature unique to Meeting Space.

We have created a number of tools for doing real work among our types of objects. Agendas are objects

that describe the structure of a meeting. Documents are actual Macintosh or PC files. Minutes are special types of documents that record conversation and actions. Shelves are directories that hold documents. Presentations are slide shows that you can show to other people. We are also working on other useful tools, such as a graphics whiteboard, a shared-text editor, and a video camera interface.

In our development, we have had a longstanding technical debate about client programs. On the one hand, a plain text interface has simplicity. Any person connected to the Internet can telnet to a system and work with other people. On the other hand, the real world has progressed to using computers like Macintoshes and Windows systems, and these are ubiquitous. Our earliest prototypes had the design assumption that text alone was important, but our preferred client program is one built with the server program, and designed to offload some of the work from the server. That text assumption has carried even into the finished product. You can work on a Meeting Space simply by telnetting to it, but there is great advantage to using our client program on a Macintosh or Windows computer. The following illustration of a Meeting Space meeting displays the advantages of a graphical as well as a textual program (see Figure 20.9).

---

### [Figure 20.9. A typical setup for Meeting Place.](#)

---



There are many features to the client program. People and other "interesting" objects show up in an icon window. When you double-click on one of these icons you see business-card information about people, or open a presentation window displaying the first slide, or walk through a door into an adjoining room, and so on. Browsers let you customize and manipulate objects.

---



### **Using Meeting Space**

Meeting Space creates closeness by presenting the virtual world to people in ways that are similar to real-life interactions. Not only can you merely type text that is spoken, as in our example and on BBS chat systems, but you can gesture, act, and provide subtle cues to your coworkers. Meeting Space inherits this variety of communication modes from its mud forebears. Teams are built by shared communications, a shared task, and the shared reality of performing that task. Communications that are rich in frequency, content, context, work, and play are the stuff of teams and the purpose of Meeting Space.

As mentioned before, Meeting Space has its beginnings in the muds that college students use to socialize and talk with their fellows in different parts of the world. They are wildly popular because they afford a virtual closeness that bridges physical distance. Two simple commands give two types of interaction to provide communication rich in content and context. The most basic is the say command. If I type

say Hello everyone.

all of the people in the virtual room will see the text:

Jon says, "Hello, everyone."

In Meeting Space, we enhance this by changing the offered verb depending on the context of the speech. For example, if I type

say Does everyone agree that we need to get this done by the end of the month?

Meeting Space cues from the question mark and displays

Jon asks, "Does everyone agree that we need to get this done by the end of the month?"

Similarly, an exclamation point has the speaker exclaim rather than say. This is subtle, but it helps create a comfortable space because it adds a conversational tone of voice that is normally heard, but in Meeting Space must be read. This additional, informational closeness helps people work together more effectively because they can virtually hear each other.

The other most common command is act. Acting provides nonverbal communications such as smiles, frowns, nods. In face-to-face communications, many actions are involuntary or subconscious, and are often subconsciously perceived and often misinterpreted. In virtual communications, you have control over what you choose to reveal with your actions. The presence of these nonverbal cues makes virtual communications feel more normal and closer to real. The act command simply appends text to the speaker's name. Thus:

act leans back in her chair and thinks for a moment.

results in the display

Margaret leans back in her chair and thinks for a moment.

The icons also add to the coziness of the environment. They give people a representation of themselves that they can tailor depending on mood or situation. If I am moderating a meeting that promises to be tense, I might put on an icon of a bear or a Clint Eastwood cowboy, to give people fair warning of my mood.

It is common in many muds that say may be abbreviated to a double-quote ("), and act to a colon (:). Meeting Space supports these abbreviations and also the single-quote and semicolon, respectively.

In addition to the public forms of communication, there are private forms, too. Only the person you are speaking with sees them. This person need not even be in the same room as you, but can be somewhere else in the space. Meeting Space provides a distinct window to enter these messages. This separate window avoids the embarrassment of whispering to the wrong person. Private asides, as we call them, can either be speech or actions, giving you the ability to negotiate or grimace behind the scenes.

---

## Summary

Virtual meetings enable people to interact in real-time, in ways that feel natural and are culturally conditioned; that create a useful and editable record of the meeting that can subsequently be used to track

projects; that act as a base for memos and other business correspondence; that enable participants to present themselves in a professional manner, as well as edit themselves before speaking or acting. The real people can be in their offices, simultaneously using another window for another task, on the phone, talking to a real person; or on a modem from home or a hotel room eating breakfast; or anyplace that offers computer access potential.

People attend virtual meetings by logging on to their virtual self, inhabiting and bestowing their personality on a data structure that has agency in the virtual world and can move from room to room, look around, talk to and hear others, gesture, carry other objects (for example, an attaché or stack of papers to hand out) fully participating in the business at virtual hand. The rapid changes in business styles, needs, and personnel, coupled with the need to do business worldwide, increases the pressure on business to adapt new methods of working, communicating, and team-building. Tools and technologies that provide a clear, understandable, and often measurable improvement in the ease of doing business are the first to be tried.

*World Benders, Meeting Space, and Instant Minutes are trademarks of World Benders, Inc.*

*Jon Callas is the Director of Technology for World Benders, and the principal architect of Meeting Space. He has been an Internet denizen since 1978 or 1979 (he can no longer remember which), and frequently acts as a native guide there.*

*Margaret Russell-Gonzalez designs human-computer interfaces and tests the usability of software for World Benders. She has been a professional writer for over 20 years and refuses to author anything. She considers the Internet a tool that darn well ought to be easy and meaningful to use.*

*For more information about Meeting Space, contact World Benders, Inc. by electronic mail at [wb-info@worldbenders.com](mailto:wb-info@worldbenders.com), by telephone at (603) 881-5432, or by physical mail at*

*World Benders, Inc.  
1 Chestnut Street, suite #333  
Nashua, NH 03060, USA*

## **A Guide for Getting Your Business onto the Net**

Now that we have looked at how other businesses already use the Internet, it is up to you to decide to make the most of it for your business. The really good news is that you can put the Internet to work for you immediately. You do not have to wait. Most companies use the Internet in either of two ways—electronic mail and the World Wide Web. Here is how you may want to do the same.

### **Electronic Mail**

Electronic mail broadly appeals to businesses because it has become ubiquitous—nearly everyone, everywhere seems to be using it. What is more, many e-mail users check their mail multiple times a day—it is the first thing they check in the morning and the last thing before they leave. Today, most large companies have their internal e-mail systems connected to the Internet (or plan to do so within a few months). Just like with a telephone, internal and external "calls" arrive; in the case of e-mail, new

messages get deposited right at your clients' desktops.

Perhaps the largest benefit of electronic mail is that it's asynchronous. Using e-mail, you can talk to anyone else on the same e-mail-connected network, 24-hours a day. The fact that e-mail permits your recipient to process your communication at a time of their choosing, not yours, means that you are more likely to get a well-considered reply than if you impose yourself and your particular issue on someone else with a phone call. Because of this dynamic, use of e-mail not only improves the time you spend communicating with others, but it permits you to deal with more of your interpersonal communication at once—improving, by extension, the quality of the time you spend focusing on projects other than your communication; you can queue up more of your correspondence to deal with at one time.

To use e-mail, the first thing to do is to get yourself an electronic mailbox on one of many systems connected to the Internet's Simple Mail Transport Protocol-connected network. There are a lot of ways to go about doing this. You may decide to use an online service provider like CompuServe or America Online because you already use their service. Instead, you may wish to use an Internet service provider. If your company has a lot of e-mail users, then you may decide to install an Internet gateway to your existing mail system so that mail messages flow through the same internal network. If you are a small company, you may even decide to not have your own internal mail system and, instead, put everyone onto the Internet. Not only does this give you Internet mail capability, it can eliminate the cost of having to run your own electronic mail server and the associated computer hardware.

One factor you may want to consider is the type of information you want to exchange (and with which users). For instance, you may already use a proprietary commercial e-mail system such as Microsoft Mail or cc:Mail, and want to exchange binary-format documents with users on your local e-mail system but only plain text with outside users. In this case, you might want to consider attaching your existing system to the Internet through its proprietary SMTP gateway, many of which will exchange only plain-text messages with the Internet. If you think you might want to distribute software, graphics, video, or audio information over electronic mail, then you might be better off building a new e-mail backbone in your company based on one of the many free or cheap combinations of e-mail transport agents and MIME-compatible mail user agents. This would permit you to send a variety of information, using standardized encoding methods with disparate software on equally dissimilar platforms.

No matter how you decide to do it, once you get your Internet mail address you are ready to go. Naturally, you can start sending and receiving mail across the Internet. This can serve as a great way to correspond with your client base. You may also want to join newsgroups and mailing lists with topics that relate to your business.

Even as just an electronic mail user, you can begin to use the Internet to stimulate interest in your company. With mail lists and newsgroups, you have the opportunity to post messages that can let people know what your business is and what areas of interest you may have. As you have probably heard, it is considered imprudent to blatantly advertise across all lists. Depending on the group, however, informative messages may be useful and are generally considered acceptable. The best way to create awareness for your company is to provide information that is useful to the group.

If you have a lot of clients already on the Internet, you may want to set up your own mail list, newsgroup, or bulletin board. This may be a terrific way for you to correspond with a lot of clients simultaneously. It may also be useful for letting your customers communicate with one another to share ideas. Any company can set up its own list easily and quickly. Basically, all you would need to do is set up a mail address to

serve as the subscribe/unsubscribe location and another that is the actual mail list address. You may decide to have a moderator on the list to read each message, post the message to the list if it is appropriate, and perhaps even respond to the message. Or, you may decide that it is best not to have a moderator for the mail list and post all messages sent in. It is as simple as that! Newsgroups and bulletin boards are also simple to initiate and manage, which is probably why there are thousands of them.

Whenever you send an informative message about your company, you may want to make it easy for people to request additional information by providing an information return address with your message. Your message might read, "We are ABC manufacturing; for more information, reply to info@abc.com." In the body of the message, you can tell interested users to include keywords like PRICELIST or DATASHEET so that they can specify what type of information they would like to get. This function, known as a mailbot, is easy to set up so that the responses are automatically generated from your mail server (or your Internet server provider). A few widely-used server packages, most of which run on UNIX, can provide a cornucopia of features to mailbot administrators, such as user-initiated subscriptions, file server and database functionality, archival storage of mailing list traffic, compression of traffic into infrequent digests, and so forth.

You can also get a surprising amount of functionality from your humble e-mailbox that extends far beyond sending and retrieving conventional electronic mail. Nearly every Internet-connected mailbox can use existing mailbot gateways to browse Gopherspace (including Gopher indexes), Usenet news, Archie, anonymous FTP, Gopher, and the World Wide Web.

## Putting Up A Web Server

There is much more to getting onto the Internet than just being an e-mail participant. I think the most important thing you can do to improve your business is to put up a World Wide Web server. This is easy and inexpensive to do.

There are two primary directions in which you can go in order to put up your own World Wide Web server. You can "roll your own," or you can get a little help from the Internet community. To roll your own World Wide Web server, you will need to know a few things you may not already know now—like how to write documents in HTML (HyperText Markup Language, used in the WWW for conveying hypertext documents using plain text) and how to configure and manage the system you've chosen to run your HyperText Transfer Protocol server. HTTP is the communications protocol usually used for conveying HTML documents and their attachments throughout the Internet.

Although most active HTTP servers are running on UNIX systems, HTTP servers are also available to run under VMS, Microsoft Windows, Macintosh System, VM, and Microsoft Windows NT. Although there may be validity to the conclusion that your HTTP server should run under UNIX, existing resources or features of a given server might indicate that one of the other servers might be more appropriate, at least in the short term. It's important to note, though, that NCSA's UNIX-based server is becoming the standard workhorse server of the WWW.

Chapter 10 has plenty of specific information to get you started as an HTML author and World Wide Web information provider. Neither of these tasks involves black magic or any particularly obscure skills—just a fair knowledge of the system on which you plan to run your server, and a clear idea of how you want to present your information. (If you are more inclined to "back-seat drive" your WWW server, you can get up and going with only moderate expense and the handful of steps in the next section.)

Here are four simple steps involved with getting your business onto the World Wide Web:

1. Decide what you want to publish on the Web.
2. Create your World Wide Web pages.
3. Put your pages onto the appropriate server.
4. Spread the word.

Deciding what to put onto your World Wide Web server is up to you. I hope the previous examples provided some good ideas as to the level of opportunities that exist. Use your imagination. Keep in mind that, for many of your clients, your WWW pages will define your business. Your Web server becomes the lens through which your business prospects view your company. As mentioned earlier, it is always a good idea to include information that may be useful to your clients and prospects. If you are trying to get new prospects to visit your home page, you may find it especially important to provide useful reference material. For instance, a business involved with computer networking may decide that a glossary of technojargon is a useful service to include within their World Wide Web pages. A law firm specializing in software licenses, on the other hand, may post some useful license agreement templates.

Once you have developed some ideas as to what you may put onto the Internet, the next step is to design and develop the actual HTML pages. There's a couple of options here: you can generate the HTML yourself, or you can get someone to "ghost write" it for you. It's probable that generating HTML is a great deal easier than you suspect, especially if you use one of the freely available HTML editors on the Net (for example, BBEdit extensions for the Mac, the NeXTStep HTML Editor, Emacs HTML mode, HTML Assistant for MS Windows, or The Hypercard Simple HTML Editor (SHE)). If dedicated HTML editors aren't your style, you can use your existing editor and run one of the available converters to do accomplish varying amounts of the work involved in generating HTML from formats such as MS Word 6.0/Word 6.0 RTF, Word for Windows 2.0, Word Perfect 5.1 and 6.0 for DOS, FrameMaker, troff, LaTeX, BibTeX, DECWrite, Interleaf, QuarkXPress, Scribe, and PowerPoint. Many of these filters, converters, and templates can be found at the following URL:

[http://info.cern.ch/hypertext/WWW/Tools/Word\\_proc\\_filters.html](http://info.cern.ch/hypertext/WWW/Tools/Word_proc_filters.html)

Information about the editors mentioned above can be found at the following URL:

<http://info.cern.ch/hypertext/WWW/Tools/Overview.html> .

If you find that your aren't inclined to generate your own HTML, it's probably time that you found yourself an Internet consultant to do this for you. You are in luck—hundreds of them are available to do this for businesses of all sizes. (If you are looking for the name of one, you can find listings of Internet consultants at MIT's commerce page—MIT <http://tns-www.lcs.mit.edu/commerce.html> and Galaxy <http://galaxy.einet.net/galaxy.html> .

When you are looking for a consultant to assist you, one basic decision you have to make is whether to use someone local (that can visit your business) or use someone in another city. You can do either. (I use Net+Effect, a local Internet consulting company at [jshunter@netcom.com](mailto:jshunter@netcom.com).) You may find that having a

local consultant can help improve your understanding of how to use the Internet as a business tool. A local consultant, by being able to visit your business, may develop a better understanding of how your business operates and be able to suggest ideas for using the Internet.

Checking the references of possible consultants should be easy enough. You will probably want to see pages the consultant has constructed for other companies. You may even want to call some of these other businesses and get their input. Since this is a pretty important decision, you will want to try your best to make the right decision. The prices for these services typically run between \$100 to \$200 per "average" page. Advanced functionality (such as downloading sample software to prospects) may take longer to implement and will cost more.

An Internet consultant can help you design your pages to best fit your needs. You will also be concerned with how these pages work over the medium. As with any other form of communication, the medium affects the message. When you are designing the pages, you will want to put yourself in your clients' positions and think how they will go about dealing with your business through the Internet. Look for ways to get to the information they will want quickly. Also, if you are expecting a lot of visits from people with low-speed Internet connections, you may want to hold back on the graphics. Many companies are giving browsers a choice at the home page of getting graphics-light or graphics-heavy pages, depending on their connection speeds.

Often, you will want your clients to contact you through your Web server. That brings up another set of choices—How do you structure responses and inquiries? First of all, you can have people contact you via electronic mail. You can do this simply by putting your e-mail address on your home page. Or, you may have a special response page added to assist with generating the appropriate electronic mail response. For another type of response, you may want to provide a form that people will print and FAX back to you (in the case of credit card orders, this may make your customers feel better about security).

Electronic forms on the Web are becoming very popular for organizing responses across the Internet. Using a form, your Web browsers can fill in the appropriate blanks. Data are collected on your Web server. You can either access these data through a database application on your server. You may prefer to have these data parsed and converted to an electronic mail message and automatically sent to your e-mail box (this can be done in the background, so your Web browsers do not deal with this—all they do is fill out an electronic form).

At this point, you have decided what information to put onto your pages, and you have your strategy on how to get your pages built. Now, you must plan on where your Web server will be located. Again, you have some choices. You may want to use space on another's server (for instance, your HTML consultant may also provide a place for clients' Web pages), you may want to use a service provider, or you may want to put up your own Web server. Let's look at these last two cases in a bit more detail.

Many companies offer Internet services for supporting your World Wide Web pages. Again, you will find these services inexpensive. You can find these listed at MIT's commerce server, Galaxy and Apollo (<http://apollo.co.uk/>). While the prices vary across different service providers, you can find entry-level prices that are in the \$20-per-month range for Web services that use 10MB of disk space and unlimited transfer.

If you prefer doing this yourself (or, if you started off using a service provider but have grown to use a lot of memory and require lots of data transfer), you can buy your own UNIX server and put up the

appropriate Internet software. A moderately priced system will cost between \$10,000 to \$12,000, including the appropriate application software. When you do this, you will also need your own direct Internet connection. Again, this is not at all difficult to set up. If 56Kbps is sufficient, you will be able to acquire this service for rates that are similar to the ones in Table 20.1.

**Table 20.1. A 56Kbps service connection.**

|                                                      |                    |
|------------------------------------------------------|--------------------|
| Router Cost                                          | \$2,000            |
| Telco Setup                                          | \$500-\$1,000      |
| Monthly Internet service provider fee                | \$125-\$250        |
| Monthly lease line fee (to telephone central office) | \$100 plus mileage |

In this example, the total cost for putting up your own Web server is \$15,000 for equipment, software, and installation, and \$350 per month for service.

When you have reached this point, you are almost ready to go—you have got your business up and running on the Internet. All that you have left to do is to let people know where they can reach you. Now it is time for some electric public relations. Once again, your Internet consultant should come in handy here. You will want to find places to post notices that your business is ready to go. Ideally, you will want to have references to your home page included on other Web servers, so that people can more easily learn about you and navigate over to your server. (As previously mentioned, putting up useful reference information is a terrific way to generate interest in your Web server.) There are general-purpose locations (like "What's new") and business-specific locations such as industry reference lists. You may even want to put the word out on appropriate mail lists and bulletin boards, informing people that your server is up and ready for business. Plus, the ever-growing community of magazines and publications about the Internet can serve as helpful PR.

## Putting This All Together

The Internet is a powerful tool for doing business in the '90s. Almost any business can take advantage of it.

Imagine, for instance, an executive search firm. Like many businesses, this imaginary executive search firm has two principal constituencies—its clients, and the professionals that it targets for recruiting purposes. Because it is in the executive search business, it does not have a lot of computer expertise beyond setting up personal computer systems and using basic applications. So, it decides to hire a local Internet consultant to form its Web pages, and it will use an Internet service provider to put up its Web server.

On the client side, this business is always looking to attract new business opportunities. It will find that putting up its own Web server is a good way to improve exposure. Even if it picks up only one new client per year, this executive recruiter has more than benefited from the costs involved in putting up a Web server. Because this is such a people-sensitive business, the executive recruiter may decide to have Web pages (complete with pictures) describing the principal partners within the business. Posting success stories and references from satisfied clients may also be a good idea.

After the company has started a new search, it may use the Internet to pass along candidate descriptions to its clients. (In this case, it may first want to review this process with its client to make sure the client is

satisfied that the confidentiality of the Internet mail exchange is within corporate policy guidelines.) It may also want to use the Internet to exchange status reports with their clients (who may find this a very useful service, since the status reports will be delivered right to their desktop computer and can be easily incorporated with other status information). This may also be more secure than doing status reports via fax machines, where it runs the risk of the report being seen by anyone near the fax drop basket!)

On the recruiting side of the business, the Internet is a wonderful resource. The mail lists and newsgroups offer opportunities to meet new contacts and even solicit potential hires. Electronic mail provides a good way to exchange information with candidates, including resumés. On a broader level, a Web server may also have a category for posting job listings, thus enabling interested parties to browse.

As a service to potential recruits, this executive recruiting company may decide to put up its own mail list, where people can share ideas and war stories relating to seeking new jobs. The agency may hire a monitor to run this mail list and make suggestions on how to improve job searching strategies. While this mail list may not directly relate to growing the immediate business opportunity, it can serve as an important goodwill tool—especially when you consider that people seeking jobs often end up in important positions within new employers and may eventually become a client of the executive recruiter.

## You Can Put Electronic Commerce to Work for You Immediately!

Marketing people use the expression "early adopters" for individuals and organizations who are the first to use new products and services. In just a short time, electronic commerce early adoption has gotten off to a great start. You might never think that HP, Grant's Florist, and a new recording artist could have something in common, but they all are relying on the Internet to increase their business opportunities. Electronic commerce over the Internet is happening.

**Table 20.2. Some electronic commerce Mosaic home pages.**

| Company                      | Home Page Location                                                                                            | Market Segment                  |
|------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------|
| Sun                          | <a href="http://www.sun.com">http://www.sun.com</a>                                                           |                                 |
| DEC                          | <a href="http://www.dec.com">http://www.dec.com</a>                                                           |                                 |
| HP                           | <a href="http://www.hp.com">http://www.hp.com</a>                                                             |                                 |
| Apple                        | <a href="http://www.austin.apple.com/">http://www.austin.apple.com/</a>                                       | Apple's World Wide Web server   |
| Internet Shopping Network    | <a href="http://shop.internet.net">http://shop.internet.net</a>                                               | Computer Retail Mall            |
| Grant's Florist & Greenhouse | <a href="http://florist.com:1080/flowers/flowers.html">http://florist.com:1080/flowers/flowers.html</a>       | Your Network Florist            |
| Bay Area Restaurant Guide    | <a href="http://netmedia.com/ims/rest/ba_rest_guide.html">http://netmedia.com/ims/rest/ba_rest_guide.html</a> | Restaurant Guide                |
| Financial Web                | <a href="http://riskweb.bus.utexas.edu/finweb.html">http://riskweb.bus.utexas.edu/finweb.html</a>             | Financial Services on the Web   |
| Commercial Services List     | <a href="http://tns-www.lcs.mit.edu/commerce.html">http://tns-www.lcs.mit.edu/commerce.html</a>               | Commerce List Maintained By MIT |
| Biznet Technologies          | <a href="http://www.biznet.com.blacksburg.va.us/">http://www.biznet.com.blacksburg.va.us/</a>                 | Business Network                |





- [21](#)

- [Finding Information You Want](#)

- [--by Joseph Janes and Louis Rosenfeld](#)
- [Introduction](#)
- [Characteristics of Internet Resources](#)
- [Characteristics of Internet Searching Tools](#)
- [How to Work with Internet Tools](#)
  - [Archie](#)
  - [WAIS](#)
  - [Veronica](#)
  - [Web Searching Tools](#)
  - [Overall](#)
- [The Importance of People](#)
- [Integration](#)
  - [Determine Your Audience](#)
  - [Determine Your Subject](#)
  - [Determine What You Already Know](#)
  - [Do a Quick-and-Dirty Search](#)
  - [Determine Human Sources of Knowledge \(Listservs and Newsgroups\)](#)
  - [Ask People for Help](#)
  - [Putting It All Together: Industrial-Strength Searching](#)
  - [Consider Recording the Whole Process](#)
- [Summary](#)

>/ul>

# 21

## Finding Information You Want

[--by Joseph Janes and Louis Rosenfeld](#)

### Introduction

We might as well come clean at the very beginning. We're both librarians. That admission is double-edged; while nobody really actively dislikes librarians and most people in fact think that libraries are good things—and the people who work in them do good work—once you say the word *librarian*, the old stereotype clicks into place. You can almost see it happen—people see the bun, the sensible shoes, hear the shushing. . . .

In fact, our perspective as librarians is very helpful in many ways when it comes to looking for information specific to a topic or question, as opposed to just surfing your way through Cyberspace for fun. Librarians typically are taught to examine a question, understand the information needed and motivations that underlie it, then use one or more information-seeking tools to try to find particular items and sources in response. This involves a great deal of strategy, knowledge of many particular sources along with their organization and search protocols, an ability to work with people, and an understanding that the tools themselves are less important than their appropriate use. It also requires the ability to integrate what you find using these tools to help people find what they're looking for.

This is the perspective that we bring to the Net. When you hear discussions about searching for information on the Internet, you'll hear people talking about how to use Archie, how to find people's e-mail addresses with finger or Netfind, intriguing ways to make Veronica work more effectively, and so on. Those are, of course, all very important, and without these tools it would be completely impossible to find anything out there. The truth is, though, that all of the search tools and protocols currently available on the Internet are very primitive and permit only very simple searching.

This chapter discusses how searching for Net stuff is different from searching in other domains you might be familiar with, why it's been difficult to construct more sophisticated search mechanisms on the Net, and it proposes some ideas and strategies to take a more comprehensive approach to looking for information.

### Characteristics of Internet Resources

Before we discuss the searching tools themselves, it's important to understand some of the nature of the resources being searched. There are a wide variety of these, including:

- Archives of text files and software available via FTP
- Full text documents available via WAIS

- Postings to Listservs and Usenet newsgroups
- Personal information from X.500 servers and other directories
- Hypermedia documents, incorporating sound, video and images on the World Wide Web
- Items of many types from Gophers

This variety is part of the richness and thus the value of the Internet as a communication medium and information resource—but it makes comprehensive directed searching much more difficult than is the case in more traditional systems such as library catalogs, indexes, and abstracting sources. Why? Several reasons:

*Different formats.* A quick glance at the preceding list shows several different kinds of formats in which text could be imbedded (Listserv, Usenet, Gopher, FTP, WAIS), and a number of nontextual formats (sound, image, moving image, directory). Several different search mechanisms have arisen to deal with many of these, but they have distinct interfaces, command structures, and access paths. There are still as yet no commonly accepted ways to search through the nontextual sources—shy of using textual tags or following hyperlinks—but this reduces to browsing.

*These sources are dynamic.* It's true. Anyone who's been on the Internet for any period of time has had this experience: a source you really like and have come to depend on all of a sudden isn't there any more. Perhaps the server is down, but it's also possible that the address has changed or the person who maintained it has graduated or moved or something, and so it's just no longer there. That's very frustrating for the experienced user, but imagine the reaction of a novice who runs across this situation. They might well imagine that it's their fault or that their individual computer isn't working right.

*Sources are sometimes unreliable.* The stories of the 95-element Periodic Table and the 1911 *Roget's Thesaurus* are perhaps only the best-known tip of the iceberg of information sources that are badly out of date or incomplete—or just plain wrong. This problem is exacerbated by the distributed nature of the Net: Once a "bad" source gets out there, it's a simple matter to point your Gopher at it or include a link for it on your home page, thus spreading its influence.

*Some of these are unique sources.* And as such are probably quite valuable, but uniqueness also means that they are perhaps unknown or foreign to a particular user, and thus less likely to be pursued. Suppose the perfect resource for your needs is available through WAIS, but you didn't know that and instead your search went through Veronica and some Web searching. You'll never find it—not because you couldn't but because you never thought to look there.

Many of these also arise in the "traditional" information searching world. There are many out-of-date or badly done print and nonnetworked digital resources, some of them are very highly specialized (or expensive) and thus not well known, and there are a large number of access mechanisms. However, they typically don't change overnight on the shelf; the access mechanisms are usually pretty well explained, and the work and reputations of authors and publishers help to allow users to decide whether or not what's there is any good. The special circumstances around Internet resources are, however, worth taking account of.

## Characteristics of Internet Searching Tools

Now let's talk a bit about the tools that we use to find things on the Net, how they work, and how they compare to searching tools used in non-networked environments.

There is a strong match between searching tools and resource types on the Net. Archie can only be used to search for FTPable files, and nothing else really works for FTP. Similarly, Veronica goes with Gopher, Web tools are used to search the Web, and so on. It is true that there are overlaps (WAIS through Gopher, Gopher through Web searching tools, and so on), but the connections are very strong. This is often the case outside the Net, but there are important exceptions. Many library catalogs also provide access to journal and periodical indexing sources and other types of resources. Commercial services such as DIALOG provide a common command structure and interface to hundreds of different databases of different types. Thus, it is harder in general to learn to search on the Net because there are so many different tools—even though many of them are extremely simple to learn and use.

You might even say they're too easy to use. Many of these search tools permit only very simple matching of single words or character strings without the benefit of Boolean operators, adjacency searching (looking for two or more words in order), or field searching (looking for a word in the title or author fields, for example). They seem so easy to use, though, that a novice searcher might be fooled into believing that the results they get are complete and comprehensive when, in fact, they may be far from it. Archie is particularly notable here, since it provides virtually no capabilities beyond simple character matching.

But this is because there's very little else it can do. Most of these tools (WAIS a notable exception) have far less information about resources to work with than do traditional tools. In general, there is no indexing or abstracting of networked resources, and no controlled vocabulary or generally agreed-upon index terms used to organize and search. In FTP land, we have brief (often extremely cryptic) file names; in Gophers we have menu titles that may or may not reflect well the contents of documents; and much Web searching is limited to text between <title> and <h1> tags.

WAIS at least operates with the full text of documents, which gives it much more material to work with. WAIS has other difficulties as a search tool, though. Although it does allow for "natural language" searching (for example, it will accept anything you type at it, discarding stop words), it then proceeds to search on all remaining words in the search statement, implicitly ORing them together, and constructing a ranked list of retrieved items based on frequency of occurrence of those words, and a few other things. Since the WAIS search engine's algorithms have never been adequately described, it's hard to really know exactly what is going on in there; most people who use it have experienced frustration since searches often produce no useful results or the ranking mechanism somehow puts the really valuable stuff towards the bottom of the list. WAIS is a worthwhile attempt to provide access to full text items, and the ideas that underlie are on the whole good, but the actual mechanism falls short in many important ways.

One further comment: These tools are also distributed, as are the resources. Thus any particular Archie or Veronica server or Web search site is only as good as the person who maintains or creates it, and it may be down or busy—as is the case with any networked resource. On the other hand, it's much easier to make one of these available than to mount a new commercial search service.

## How to Work with Internet Tools

The preceding chapters have given some good instruction on how to use searching tools; here we'd like to give some advice and tips not so much on specific strategy or the operation of individual tools, but rather strategy on when to use which tools, and how to decide which ones might be most effective.

### Archie

In many ways, this is the worst tool available. Since it is able to operate with only extremely limited information about FTPable files, there's only so much it can do—but its lack of features beyond simple character-by-character matching make it often frustrating to use. This frustration is made worse by its use of implicit OR—if you search for more than one word in a single Archie statement (for example, prog internet society), the results will be any entry containing *either* of those words or both, thus producing many more returns than probably desired. There's no easy or efficient way to do a Boolean AND in Archie.

Of course, if you're looking for the location of an FTPable file, it's about all you have to work with. We suggest starting a search for information using Archie only if you know that the resource you're looking for is an FTPable text file (like, say, the early edition of *Zen and the Art of the Internet*) or software file (Mailstrom, for example).

As a secondary source, you might try Archie if the topic you're looking for is relatively specific, can be expressed using only one unambiguous word (thus avoiding the AND problem), and has some potential to be found in an FTPable file. This is not the place to do free-form, let's-see-what's-out-there searching unless you have a lot of time on your hands.

### WAIS

We've already discussed some of the limitations of the WAIS retrieval mechanism. Again, this is the only place (so far) that you'll be able to search the full texts of the resources, so you might well have more luck searching here than with other tools. Please bear in mind, though, that searching in full text is very tricky, especially with words that are ambiguous or have more than one meaning. Searching using WAIS on the word "bush" when doing a search on shrubbery will also get you items about George and Barbara Bush, among other things.

This problem is mitigated (somewhat) by the ranking mechanism that tries to take word frequency at least partially into account, but that doesn't always work.

We advise a two-step approach to searching in WAIS. The first step is useful when searching in the Directory of Servers, a sort of catalog of the individual information resources available via WAIS. In this source, you are only searching descriptions of resources, not the actual documents themselves. These descriptions range from the quite good to the poor to the nearly nonexistent, and so we suggest searching for broader subject areas here. The results you get from such a search will be descriptions of individual resources, and you can explore these individually (more about this in a second) to see which ones might be most useful. For example, if you were looking for articles or criticism of the *Iliad* and the *Odyssey*, don't search for those words in the Directory of Servers—they won't be there. Search instead for words like poetry, poem, classical, or even literature, although that one can be used in many different ways.

Notice how we select terms that are conceptually broad but are not that ambiguous.

After you've identified potentially good resources, proceed to the second stage. (Experienced WAIS users may know resources well enough to skip the first stage.) Here, you are actually searching in the sources themselves, so you now can use the more specific words like Iliad, Odyssey, epic, and Homer; the likelihood of weird retrievals is lower here if you've carefully chosen resources to search in.

## Veronica

Veronica is probably the most sophisticated of the Internet searching tools; it supports Boolean searching, truncation (searching on `tornad*` to get `tornado`, `tornadoes`, `tornadic`, and so on), and enables you to control the number of results you'd like to see using the `-m` option. It still, however, only permits searching of a very limited amount of information about particular items: menu names or directory titles, at most a line each.

Nonetheless, it does allow you to do some quite effective searching. We advise searching on specific words if you have them and use them in sensible combinations, but don't get too specific right away. Take advantage of the Boolean search capabilities. If you're looking for information about the Internet Society, for example, try searching on `isoc` first, the Society's acronym, and a nice, unambiguous word. If it turns out that you get too few retrievals using this strategy, try broadening to a search like

```
(internet and society) or isoc
```

By the same token, a search on `statistic*` to pull up available statistical information will produce a lot of stuff—probably too much, in fact. Something more specific is called for, maybe

```
inferen* and statistic*
```

```
or
statistic* and software
```

```
or
statistic* and internet and (use or usage)
```

depending on what you're looking for. Veronica is really the only search tool that can support this kind of broadening and narrowing of strategies, such as is possible with many nonnetworked systems.

## Web Searching Tools

As we write this (late summer of 1994), this is the area that is changing most quickly and dramatically. At first, there were no tools for directed searching of the World Wide Web, just starting points and gateway pages. Then the first few started to appear. Then, as more and more attention was being paid to the Web protocol because of the appearance of NCSA Mosaic, the floodgates started to open. Now, there are pages devoted to providing front ends to many different search engines. An example of one is the Meta-Index maintained by the Centre Universitaire d'Informatique (CUI) of the University of Geneva. Its URL is: [http://cui\\_www.unige.ch/meta-index.html](http://cui_www.unige.ch/meta-index.html).

We'll discuss three of these engines in a bit more detail. More are being developed and released all the time; by the time you read this, there may well be many more available.

One of the earliest successful engines was JumpStation, developed by Johnathon Fletcher of the UK. Its URL is <http://www.stir.ac.uk/jsbin/js>, and it permits searching of text that appears in Web documents, but only small parts of it. These documents are created using a markup language called HTML (HyperText Markup Language). They have tags imbedded in them to distinguish between major parts of the documents (lists, images, links to other documents, and so on) JumpStation takes advantage of this by permitting searching within two kinds of tags: the titles of documents and the first-level headers (roughly analogous to the major, roman-numeral divisions in an outline). This can make for effective searching, but again there are no conventions for what people write in these documents, so you're never really sure what you're getting.

The Web Crawler, from the University of Washington (URL: <http://www.biotech.washington.edu/WebCrawler/WebQuery.html>), is also produced by software robots (as are many Web searchers) that scan through the Web, starting with an initial group of documents and following links throughout Web space. The full text of documents are included in the index—which permits more thorough searching—and some simple Boolean searching is permitted, but users are still faced with the pitfalls of full-text searching. It does, however, have many attractive features. A side benefit of this discovery process is the Web Top 25 List, the most frequently referenced Web documents found by the robots.

The World Wide Web Worm (WWW) is also built by programs that scan the Web identifying new documents. The Worm (URL: <http://www.cs.colorado.edu/home/mcbryan/WWW.html>) permits searching of URLs of documents, so it is easy to find all the Web sites in particular countries, or of particular organizations—a search for all Web documents containing the .se domain (from Sweden) or the syr.edu domain (from Syracuse University). Content-based searching is more problematic because there is no convention for indicating content within the URL.

In addition to these directed searching tools, there are a few other techniques we advise. We often find it useful to search through the text of the What's New page maintained by NCSA (URL: <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/whats-new.html>); the descriptions are usually reasonable reflections of the content of individual Web sites, and they provide a convenient way to search for recent resources. It may well require repeated searches on several different pages because the old ones are divided up by month, but it is still a useful strategy.

There are also a number of subject-oriented pointer pages, such as the WWW Virtual Library: <http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html>

the Whole Internet Catalog:

<http://nearnnet.gnn.com/wic/newrescat.toc.html>

and others. They all suffer from a similar problem: Once the number of resources they point to grows beyond a certain point, it gets very difficult to maintain these lists in a reasonable way. They can get quite cumbersome and harder to use. Again, though, they do provide another, more general approach to searching for Web-based material. These two lists, along with several others also maintained manually, can be searched via the W3 Catalog maintained by CUI in Switzerland (URL: [http://cui\\_www.unige.ch/cgi-bin/w3catalog](http://cui_www.unige.ch/cgi-bin/w3catalog)).

## Overall

Due to the nature of the searching tools available and the kinds of resources you find out there, we give you the following advice: if you're just starting out looking for information about a specific question, or on a particular topic, it's often best to start with Veronica or Web searching. There's a great deal of useful information available via both Gophers and the Web, and it's just plain easier to get at it. FTP and WAIS have their uses, but we see them as secondary approaches, in most cases.

A lot of this becomes more clear as you gain experience with the different types of resources typically found under different protocols. It does get easier as you spent more time on the Internet.

One of the other things you'll discover the more time you spend on the Net is that the most important resource that exists out there is the other several million people who coinhabit it. Sometimes they have the answer; sometimes they can point you to the person who does or the server where it sits. But once you've tried a bit of searching using these tools, it's often best to start asking around.

## The Importance of People

It's a natural leap to make. There's a global network enabling access to millions of computers. On those computers are located a lot of really interesting things. But there's no really good way to find the good stuff. What's your natural impulse? Same as ours—ask someone. Research has shown that that is the preferred information-seeking strategy of many people: ask your neighbor, ask your colleague, ask your friends. Well, it works on the Net, too. Ask anyone.

There are a couple of ways in which this takes place. Send your friends e-mail and ask them what they found that they like, or if they know of any keen stuff about *Babylon 5*. It might work, especially if they share similar interests. But, chances are, they might not know of specific things.

Your next best strategy is to go to Listservs or Usenet newsgroups. We haven't talked much about these because searching really doesn't come into play here. You can search within archives of old Listserv postings, and you can search within the subject lines of articles in Usenet (using many newsreader clients), but these are much more communication channels than information sources.

It's possible that your questions might be answered by an FAQ (Frequently Asked Questions) document maintained by a group. These are exactly what they sound like: questions that often arise in the context of a particular group and answers that people in the group have developed over time so that they don't clog up the group. Once you find one or more groups that appear to be in your area of interest, the best thing to do is to subscribe to the group or scan it and look first for the FAQ. Most high-traffic groups will have the FAQs posted every week or two, so it shouldn't be too hard to find. (See Chapter 17, "Discussion Forums," for more details on how to go about this.)

If the FAQ doesn't answer your question, or if it piques your curiosity enough to find out more, then just follow the flow of conversation of the group. You can learn a lot by just reading, and by participating in the discussion as appropriate. We have often found that just subscribing to a newsgroup or Listserv for a few days can be very enlightening.

At some point, though, you may find that the specific answer to your specific question (just what do you feed pet squirrels, anyway?) isn't being responded to on rec.pets, so you decide to ask the question yourself. Here's where a bit of caution and thought are required.

Most of the groups we've spent time with are composed of terrific people who will go to surprising

lengths to help other people. We've both had the experience of asking a question and getting several responses within hours: recipes for Siberian ravioli (seriously) or charts showing microcomputer sales over the previous three years broken out by manufacturer. It's amazing!

But sometimes you have to be a bit careful how you ask these questions. You must always remember that many of these groups have become communities where participants have histories and know each other, and have developed mores and norms for discussion. You wouldn't barge into somebody's seminar or living room, and without so much as a hello, start asking them questions. It's rude, and probably won't get you very far towards getting help or an answer. The same sort of thing applies here.

We advise subscribing to a group and spending some time just reading and learning about the group and what goes on there, and how the discussion goes, before posting anything including a question. There are some groups where people can get attitudes and be very touchy about how things are said and, if some of these people decide to get on your case about being a newbie asking stupid questions in a place you don't understand, it can be an unpleasant experience for you. Whether this is a good or appropriate thing or not is a question for another day. The fact is that it can and does happen, and you should be forewarned and prepared. Reading a group for a few days will give you a much better feel for the sensitivity level of people and, most importantly, whether or not your question can be dealt with usefully here. Maybe there's a better group like `rec.pets.squirrels` or something that can be of more help.

We want to emphasize: *don't be intimidated*. Most of the groups that we participate in are filled with terrific people who are more than happy to see new folks participating and help them out. You've just got to be a bit careful at first.

## Integration

This chapter and the preceding chapters in this section have discussed the individual searching tools, their usage, and their problems. While any individual tool might be good for quick and dirty searching, it can be quite a headache to *exhaustively* search for all of the Internet's information regarding a specific subject. After all, there are times when you will want to know *everything* that's out there on beer brewing. What techniques can you apply that go beyond asking around in a few Usenet newsgroups or doing a simple Veronica search?

Well, you might think about doing an exhaustive search as the equivalent of an all-out war: you have to use every tool in your arsenal. However, would it be effective to just throw each of your resources at the target all at once? You might find this approach to be frustratingly disorganized, hard to keep track of, and unfulfilling in all respects.

Instead, you need to plan an attack that *integrates* all the tools described above in an effective way. This means you'll want to use the appropriate tools at the appropriate times: certain approaches can be used to set up the search, others can harvest the bulk of the relevant information, and so on. This section describes a series of generic steps that you can use in most any exhaustive search for information.

## Determine Your Audience

You should begin by considering how the information you find is going to be used. If you are the only person who will use this information, your search will be much less complicated. If there is a larger audience involved, you will need to determine whether they share your perspective, vocabulary, and

access to the Internet. Are they a scholarly crowd, corporate researchers, high school students? Do they know how to use the Internet's tools already? Are they likely to have the same level of connectivity to the Internet as you do? The answers to these questions should help you determine such things as how to "publish" your "product" (for example, in print or via some Internet tool), what language to use, how much detail to provide, whether or not you need to explain what a Gopher client is, and so forth.

## **Determine Your Subject**

The nature of the subject itself can dramatically affect how your search will go. Because the Internet's information is quite spotty in terms of coverage, you might find that there is too much information related to your subject, or perhaps not enough. Although the process of information searching is iterative, you'll want to avoid changing gears in the middle of this process: backtracking means you'll lose time and perhaps interest in completing your search.

To avoid having this happen, think about both the broader and narrower versions of your subject. For example, if you are interested in finding the Internet's offerings on beer, you might consider the broader category of beverages. In the back of your mind, your narrower interest may really be information on brewing beer. Then keep your eyes peeled for information fitting both the broader and narrower categories during your search. This way you'll be prepared if you have to narrow or broaden your search later on.

## **Determine What You Already Know**

Record your current knowledge of the Internet's resources relevant to your subject in a single document. This document can serve as the starting point for your search. Ask yourself what you already know about each resource: how did you learn about it? Who maintains it? Is it valuable or not, and why? The answers will help you determine where else to look, whom to ask for assistance, and what characteristics to look for when evaluating a resource.

## **Do a Quick-and-Dirty Search**

Later on you'll be asking for help on various mailing lists and newsgroups. Before you do this, you'll want to demonstrate that you've already made at least some effort to get to know the subject at hand to gain their respect, show that you know what you're doing, and show you're serious. You'll also want to avoid receiving responses that mention the same old well-known resources.

Building on what you already know, a quick-and-dirty search proves to other Netizens that you're someone worth helping out. It can give you an initial sense of how much is out there on your subject; you may find this moment a good time to adjust the broadness or narrowness of your search.

The best tools for quick-and-dirty searching are Veronica and JumpStation. They don't require much expertise and (usually) return usable results.

## **Determine Human Sources of Knowledge (Listservs and Newsgroups)**

As described earlier, there are many techniques for identifying groups of people who might share your interest in a subject. What many people forget at this point is to consider such issues besides subject; namely, the level of appropriateness of a specific group's discussion to your information need. For

example, your investigation of artificial intelligence resources might find four or five relevant groups. As you lurk on these groups, you might find some to be high volume "general interest" groups full of unsophisticated ramblings, and worse, many questions. Remember, you're looking for a good forum for your question, and don't want to compete with so many others. So avoid these groups in favor of those that may be narrower or less on your topic but that, instead, provide a tighter and more responsive community.

## **Ask People for Help**

Once you've spent some time lurking and feel sufficiently comfortable with a few relevant communities, send them a message that briefly describes: 1) who you are; 2) what kind of information you are looking for; 3) where you have already looked; and 4) what you already know (this can be a short listing of resources identified in the steps above). And of course, be sure to state that you will make the results of your search available to the Internet.

Remember that this process can be frustrating in that it won't provide you with the immediate results that a Veronica search might. Nor will your query always be received favorably. On the other hand, remember that people are the best source of information on and about the Internet, and that your query has at least planted some seeds that might take some time to nurture before you can harvest their results.

## **Putting It All Together: Industrial-Strength Searching**

You've now completed all the necessary preliminaries for doing an exhaustive Internet search. You've thought about such issues as audience and subject, you've found some useful resources, contacted a few relevant communities, and ideally you've started getting some helpful pointers from those communities.

Now you can begin the heavy-duty searching you've been itching to do. Identify a set of keywords, plug them into the searching tools as described in the previous chapters, and see what happens. In fact, this is the best time to apply all the tricks you've now learned for Archie, WAIS, Veronica, and the various Web searchers.

Of course, keep in mind that, as mentioned before, some tools are less than useful for certain purposes, so you'll want to spend less time with them. For example, unless there is a lot of yeast-growth-tracking software out there, Archie probably won't be too useful for your search on beer brewing. Also, you know that certain tools are not your personal favorites. So don't spend too much time wrestling with WAIS at the expense of the rest of your search. You can always come back to it later.

## **Consider Recording the Whole Process**

You probably have found everything you're looking for—at least, for the moment. You might want to record what you've found as an FAQ or Internet guide so you can come back to it later. Also, good netiquette dictates that you send copies to the kind folks from various mailing lists and newsgroups who helped you out.

You also might want to keep current on your subject. Maybe beer brewing technologies will be revolutionized in the next six months (one can only hope!), so you might want to do the search again in the future. And until the bugs are worked out of Knowbots, filters, and other intelligent agents, you will have to pretty much start your search all over. For this reason, consider briefly recording what you had to

go through: which keywords you used, which groups and tools were the most helpful, what resources you found. A little work in this area now could save you a lot of time and hassle in the future.

## Summary

We've tried, throughout this chapter, to present some ideas and ways of thinking about searching for information over the Internet, from our perspective as people who deal with information for a living. The approach we advocate here—searching where appropriate and efficient, asking people for help when possible and feasible, and pulling varied sources of information together to form a coherent whole—is, we think, at the moment the most sensible one to take. It's not perfect by any means: sometimes haphazard, certainly incomplete and idiosyncratic, but it does produce useful stuff.

It is also the case that, at least in the near future, "traditional" searching tools such as we're accustomed to in commercial, library-oriented systems are not going to arise in the Internet environment. Two reasons, primarily: (1) these are complicated and large projects; nobody's paying for their development and so they probably won't be developed in a big hurry, and (2) the highly distributed and dynamic nature of the Net mitigates against their development. This is because nobody has sufficient control over what's going on out there to hold it still long enough for it to be "indexed" or "classified," or even described at all well, and besides—the traditional tools don't always work in the print world, either.

The problems of representing and searching for information are large and difficult ones. Centuries of work in libraries have produced mixed results, but on the whole, we have systems that work tolerably well. Over the last few decades, a number of ideas have been proposed from the domains of computer science, notably artificial intelligence, towards solutions of these problems. In our view, these suggestions have oversimplified the problem and thus have, on the whole, failed. The heart of the matter is language—the medium in which information is often bound up—and its ambiguity and flexibility. Information in forms of audio, image, or moving image may present even greater challenges, as we seek to develop vocabularies and structures to represent them.

This is an old story. As new technologies have been developed that affect our ways of storing information (writing, printing, scholarly journals, magnetic storage, and so on), new methods have had to be devised to find it again. Writing led to titles; mass printing led to book catalogs; journals led to indexes; magnetic storage led to data structures. Now we have globally distributed high-speed networks, and a new response is required. The ultimate form (or forms) of that response are at present unclear—but for the moment, a few basics from the library world may suffice.





- [22](#)

- [Art on the Internet](#)

- [--by Kenneth Greenberg](#)
- [A Thumbnail Sketch of Artists in Cyberspace](#)
  - [The Community of Artists](#)
  - [The Business of Art](#)
  - [Research and the Arts](#)
  - [The Art of Publicity](#)
    - [Interlinking Links](#)
    - [Letting Others Know](#)
    - [Following Up](#)
- [Internet-Based Art](#)
  - [Art Forms Native to the Internet](#)
    - [Image Files: ASCII, GIF, JPEG, and More](#)
    - [Sound and Media Files](#)
    - [Interactivity: OTIS, Muds, MOOs, and More](#)
    - [Creative Writing](#)
  - [The Internet as an Art Medium](#)
    - [Internetic Installations](#)
    - [There's No Place Like Home Page](#)
    - [Electronic Cafes and Cyber-Culture](#)
  - [Curating and Criticism on the Internet](#)
    - [Understanding What You See](#)
    - [Media and Cultural Critique](#)
    - [The Politics of Critique](#)
  - [The Audience Participant](#)
- [What Artists Contribute to Understanding of the Net](#)
  - [Visualizing Cyberspace](#)
  - ["Hello World!": Self-Image Online](#)
  - [me.com: Getting Your Work Out There](#)
  - [A look Toward the Techno Future](#)
- [Pointers to Art Resources on the Internet](#)
  - [General Art Resources on Gopher](#)
  - [General Art Resources on World Wide Web](#)

- [Images, Sound, Software, Databases, and FAQs](#)
- [Selected Arts Usenet Groups](#)
- [E-Zines, Databases, and FAQs](#)
- [Art Education and Other Art Related Information](#)
- [Cybercultural Links](#)

## 22

# Art on the Internet

[--by Kenneth Greenberg](#)

My involvement in interactive computer technology and art goes back to the early 1970s, in an experience that gives true meaning to the term "spaghetti code." Three weeks before Charlotte Moorman's Avant Garde Art Festival of 1972, sound artist Liz Phillips asked if I'd be willing to cook up 500 pounds of six-foot strands of spaghetti as part of her upcoming interactive/performance work on the Alexander Hamilton Ferry. "Electric Spaghetti" used capacitive fields to sense the audience's grabbing and handling of hundreds of six-foot-long strands of pasta pulled from a massive heap. Information from the sensing devices was translated into electronically generated sound. We did not know the word *Internet*. The word *computer* conjured up toting huge stacks of hole punched cards. For me, the allure of interactive electronic art has never disappeared.

Historically, art and invention tend to go hand-in-hand. Pythagoras derived his famous theorem while exploring the nature of music. The Greek architects made exquisite use of geometry. At the dawn of the twentieth century, as physicists explored the concept of relativity, artists underwent an abstract revolution. The underlying issues of art and science are more closely aligned than it might appear at first glance. Science seeks to observe nature and phenomena empirically, and then to propose a theory which explains the observation and reliably predicts future observations. An artist also observes nature. The artist seeks to create a structure or style that will convey the meaning of his or her observations. The work an artist produces must ring true for others if it is to survive. As the audience views a work of art, whether in one moment of immediate time or over centuries of cultural critique, people experience the artist's hypothesis. Each of us becomes the judge of the work's ultimate validity. There may be an idea presented as simple as "behold the beauty of the world around us" or ideas that are quite complex such as the movements within Stravinsky's "Rite of Spring."

Creativity has always existed in all of the realms of computer technology. Start delving under the hood of almost any well written code and you will find humor and personality—albeit often strange and esoteric. Program names such as BIFF, the mail-notification utility named for the dog who barked when the mail arrived; subroutine labels such as "should-patch-later"; or the Internet tools named "Archie," "Veronica," and "Jughead"; or the ever-present RTFM (Read "this fine" manual) are the markings left by the people behind the programs. These are the same markings of the soldiers who scrawled "Kilroy was here" or cave painters leaving vivid visual descriptions of their daily lives. These are the traces of that part of the human spirit that must express and create.

In the late 1970s, it wouldn't have been easy to predict the massive computer markets we see today. We are now poised on the moment of critical mass of the computer communications era. Internet, Multimedia, Virtual Reality, and Interactive TV are among the ideas being discussed on a daily basis across all disciplines. While art in its purest sense might have been confined to a few small special-interest groups on CompuServe or Delphi a

few years back, the presence of people online today whose lives are intimately connected to the arts is impossible to measure. As the issues of presenting oneself or one's business to the world at large begin to emerge, we see the impact of artistic judgment and quality take on even greater significance.

My discussion of art will have a definite bias toward visual and technology-based art as these are my primary domains. However, most of the information here can be equally applied toward any number of areas in the arts. There may not yet be as many archives devoted to dance at the moment as there are to visual display, but there are, nonetheless, dancers not only interacting through words on the Net, but dancers whose performances are on the Net. Many of the issues, such as presenting and promoting ones work, are the same. Finally, the nature of the creative process cuts across all boundaries and is, in fact, not exclusive to the arts. The thought processes that created the Internet go beyond rote and routine.

## **A Thumbnail Sketch of Artists in Cyberspace**

Artists wear many hats in our society, and it's no surprise that their varied roles turn up in many different locations and applications on the Internet. Whether we examine the active online communities created by artists for their own purposes or explore the importance of art for business and the world at large, art has been a significant force behind the development of the network and within its daily functioning. The first section of this chapter looks at the activities and sites where artists are working and communicating on a daily basis.

### **The Community of Artists**

Estimating the number of artists in the general population has been a hotly debated issue for years. While, for example, U.S. Department of Labor statistics place the number of working artists between 1 to 2 million or approximately 1% of the population, a 1988 Harris poll found numbers indicating that, for example, 30% of the population draws, paints, or engraves, 30% play a musical instrument, 51% have a serious involvement with photography, and so on. When the list is read completely, it would appear that very few people in this country are not involved with art. The Internet is a specialized community. One expects to see primarily technically oriented personnel here. However, it doesn't take much investigation to find that the mainframes contain massive amounts of disk space devoted to graphic image files. We find literary works and literary critique. There are repositories of sheet music and sound samples. Gophers devoted to crafts or theater exist. Finally, in the broadest sense, the emerging standards of graphical interface, and the many programs designed or being developed to support these standards, such as Mosaic, are in part dealing with the basic artistic issues of ways of seeing and ways of showing.

The community of artists on the Internet can be seen as four distinct groups, although any individual artist might be in one or all of these groups. Artists who are part of a specific discipline would be one of these major groupings. This would include, for example, fiber artists, ceramic artists, musicians, filmmakers, dancers, poets, light artists, painters, sculptors, and so on. What brings these people together is obviously their common interest in their media or styles. Traditionally these artists would converge either through local or regional organizations, exhibitions, magazine subscriptions, or interpersonal contacts with others in their field. What is unique to the Internet is the ability to readily target and access the world of people in a given field. Ceramicists are not necessarily listed in the phone book as such. I can't go to a town and easily determine where I might find cello players. Though we are not quite at this point with all areas of interest on the Internet either, in many fields it's possible to connect with others of similar background and interest quite directly. Whether by Usenet posting or mail list subscribing, artist users in a multitude of individualized areas are meeting online.

A second major grouping within the community of artists on the Internet can be viewed as artists involved with broader organizations or institutions that bring a diverse group of interests together. While in the previous example we looked at the branching out to all of the individual specialties, here we look at the ways that all of

these groups come together. One example might be the Arts and Humanities department of a large university. Another example might be the Smithsonian, which maintains a wide range of programs and exhibits covering a broad range of disciplines. A significant example in this area is Arts Wire ([artswire@tmn.com](mailto:artswire@tmn.com)), which is an Internet based organization that has a membership made up of artists, art administrators, art organizations, and art activist groups. Arts Wire regularly publishes information about governmental activities that affect the arts, grants and job openings in the arts, art criticism, exhibitions, and more. Among other interesting and unique areas, they maintain a database of art material hazards. A group such as Arts Wire becomes a meeting place for all disciplines involved with the arts. Additionally, it becomes a potential political force for artists who otherwise may not have had the ability to network so readily. I would predict that as the Internet grows, we will see greater reliance on large scale, discipline focused sites that act as pointers to and connections with the diverse groupings which otherwise would be spread too thinly.

My third grouping includes almost all of us. It is the realm of people who enjoy the arts—the audience. There is already art for sale on the Internet. There are magazines for art appreciation. There are listings of events. And with the rapid expansion of the World Wide Web and Graphical Interface software, we see thousands of clip art images being used on both professional and amateur home pages.

The Internet is becoming not only a conveyer of arts media but an art media in and of itself. This newly emerging field that I term *Internetic Art* would certainly have to include OTIS at [sunsite.unc.edu](http://sunsite.unc.edu/pub/multimedia/pictures/OTIS) (pub/multimedia/pictures/OTIS) and other locations. This energetic group has for several years initiated a number of interactive art projects where participants may upload an image for others to download and manipulate, and then re-upload to await further stages of manipulation. OTIS recently was home to Synergy-Corpse, an online version of the favorite pastime of the early 20th-century avant garde where one person draws the head of a person and folds the paper to conceal it; the next person draws the torso and again folds the paper. This sort of visual game of "telephone" continues until a composite character is created. Here we see an Internet site that attracts people from my first grouping. It's within a site that is of the second grouping and because anyone can download and view and, in fact, participate, it brings in users from the present group I'm discussing.

Finally, my fourth grouping includes people who are conducting business that involves art. As mentioned earlier, presentations and other forms of advertising on the Internet are destined to be possibly the most significant online areas of business. Selling and presenting on the Internet will require more than just a good idea. Just as there is a qualitative difference between a home movie and a professionally produced television commercial, marketing on the Internet will require an artistic vision combined with an understanding of how information finds its way to the user. Front ends will be successful not only due to typical software issues regarding efficiency, organization of information, bandwidth, and so on, but also in large part based on their artistic cohesiveness and appeal. The entertainment industry itself will become part of the Internet. We already have ongoing informational and entertainment features such as Internet Talk Radio with popular programs like Professor Neon. (Info is available via Gopher at [vortex.com](http://vortex.com).)

## **The Business of Art**

Art is one of the few businesses in our modern service-oriented economy where an individual actually physically makes something. Often, this same individual is seller, designer, purchaser of supplies, bookkeeper, and floor sweeper. In my small neon art and design business, I create promotional material, meet with clients, draft proposals and presentations, and engineer the project—all before I actually blow and bend the glass and take on the problems of assembling and installing my finished work. I am not unusual compared with other artists and, in fact, I have the advantage of working with an unusual and sought after medium. The point I am making is often overlooked: Every artist runs a business. We all possess a romantic image of the artist feverishly painting at night and sipping wine in French cafes by day. The image should really be amended to include contacting and visiting galleries and potential buyers, studying other works of art, making phone calls to commercial clients, and perhaps

waiting tables in that French caf[130] to have the cash to reinvest in materials to produce more work. It is no more fun and no less painful than any entrepreneurial experience. The Internet has opened up a completely fresh territory for career artists.

Art enterprises on the Internet today include basic graphic and photographic services, sound services, and production and writing services. Online galleries exist where art can be purchased and where artists can, for a fee, display their work. Periodicals and fee-based databases in the arts can be obtained by subscription. With the emergence of online catalogs and improved graphical interfacing, corporations will likely have departments oriented toward online art and presentation. Agents will broker online talents to industry.

The basic human activity of meeting and social networking is catalyst for further creativity and business. Individual artists enjoy relatively equal access to the Internet as anyone else does. Not only are there online exhibitions, concerts, and performances, but there is an entire cyber culture that has always been on the network. Conferencing groups such as ECHO are based almost exclusively on the idea of artists, writers, and cultural observers hanging out and chatting. University home pages are a stimulating way to connect with the Web's resources. Public art has found its way to the Net in the same sense that it exists in public spaces. Perhaps we will see corporate-sponsored art on the Net—neighborhood beautification or percent-for-art-styled programming along the information highway, where virtual-reality technology will bring artistic environments to the home or office.

## Research and the Arts

For all of the wonder and power that is the Internet, let's not forget that it's in essence a powerful research tool. Most tasks on the Internet are research. When I first became interested in art on the Internet, I performed rather simplistic whois searches of simple words like *art* or *graphic*, and that began to give me some idea of what was (and what was not) out there. As the network grew and my own skills improved, I created more specificarchie queries and could thereby get a glimpse of which sites might be fertile and useful. I also learned to open my news reader (I use tin) with wild-card combinations where I guess at strings such as *\*craft\**, *\*graph\**, and *\*art\**. Some searches will, of course, lead nowhere or hit dead ends. Occasionally, a search will open a previously unknown region where much more than expected is found. With the introduction of Veronica and Jughead, it became possible to find and enter news, Gopher directories, and other remote sites on the fly. Similar resources on WWW (World Wide Web), such as WebCrawler, enable this same ability to seek and connect.

I've found that it takes a combination of skill and intuition to use these tools effectively. You have to know not only how to evaluate potential choices of direction but you also need to know the "feel" of your connection. Is a directory really being retrieved or are we hanging? Is the file very big or is the site overloaded and running slow? Is this Gopher link temporarily down or has it been abandoned? If I come up empty handed, is this because there is nothing available or am I just not asking the right questions in the right locations?

Artists, particularly in the commercial field, perform an enormous amount of research. If a client needs to create a new campaign, for example, the artist does not merely conjure up an image. Countless hours are spent studying a wide range of ideas and solutions that others have tried in areas that may resemble or touch upon the particular project. Once an idea has solidified, there still may be research required to obtain the best source of an image or, as in the case of a radio commercial, sound samples. If the presentation will be network-based, as mentioned earlier, thorough familiarity with what network paths will lead a perspective audience to the work will become crucial.

Other specialized and technical uses of Internet research are purely academic and cultural. It's possible, for example, to search all references to pointillism or the works of Calder. Even the popular quest for recreational art and entertainment has its strong presence. One can easily uncover theater schedules in various towns or follow television plots or even find ones way around a botanical garden. The world of network computing has its own

in-house entertainment and culture. We have cookie servers, coke machines, electronic pubs and cafes, e-zines, muds and moos, flame-fests, and more.

## The Art of Publicity

Online communities are linked together by virtue of the electronic time and space that we share. Most individuals and organizations desire more than to simply be waiting to see what happens next. As is true in the business world and in socializing, promoting your existence and your abilities is an extremely important aspect of your life in the Internet community.

## Interlinking Links

Despite the vast and seemingly impersonal face of online networking, the Internet can be a rather close-knit community. This is because of the nature of individualized or selected linking. Think of a giant bookstore where people in the engineering aisles are likely to share a quip or an observation with others interested in this area. Similarly, the fiction area or the art area will draw in a specific clustering of people. They may or may not interact. The Internet has a remarkable ability to draw users from widely diverse geographic locations, and even vastly different cultural backgrounds, into exceptionally focused points in Cyberspace. The Usenet is one of the best examples, and we see interest areas that are seemingly very limited in scope, populated by strong core groups of users. We see news groups that range from rec.arts.marching.drumcorps to alt.alien.visitors.

The physical invisibility of other users, often thought to be the cold side of electronic communication, actually forces users to make more of an attempt to communicate. In some cases communication is the only activity available. Often, I think of the Internet as an ocean where thousands of us are bobbing up and down on our small crafts and there is a camaraderie that develops around this. We ask questions like, "How do you navigate around those rocks up ahead?" Users will frequently volunteer their assistance, having struggled with similar problems or with their own special circumstances. There is a sort of electronic karma, where your being brave enough to ask a question, frees you up later in life to help someone else.

Somewhere in this process, temporary and sometimes long-term friendships develop. These relations may sometimes be the result of a series of flames where everyone eventually will kiss and make up, or it may be around a very specific issue such as users of a specific software who come together to exchange program tips, macros, or complaints. Opportunities exist for promoting and networking which are not hindered by the normal constraints of time and space. This is not to say that one should barge into any area where you feel you deserve attention and start right in with, "Hi, I'm Howard. I make the best Widgets on earth. Here's my hypertext resume, complete with 24-bit, high-resolution photos." In some places you'll be booted or hooted out of the loop within microseconds. Nor am I suggesting that people more subtly worm or weasel their way into an area they feel will be fruitful for business.

The Internet is a community, and it also has obvious commercial potential. While there has been a de facto and almost perfectly anarchistic state of peaceful coexistence and mutual support among users, this does not mean that people are not interested in exploring and developing opportunities for making money. To me it's not merely an issue of effective targeting—it's an issue of artistic integrity—or, in a word, honesty. I am more likely to buy a computer, for example, were I to find a seller whose life truly revolves around how computers work, and this person is willing to help me make intelligent choices. On the other hand, it would be difficult to resist a straightforward statement that said "20 years in business, lowest price."

Promoting oneself as an artist has unique aspects to it. Unless you are a known artist whose work is in demand, there is not necessarily a group of perspective buyers specifically looking for your work. There is not always what one might technically call a "need" in the sense that a business needs a desk or a car, although this view has been strongly debated. (There are many reasons that a business or an individual may truly need art.) Even when

there is a strongly motivated client, the choice of what to buy can be subjective and at the mercy of changing desires. When you buy something like a printer, you may struggle over the look, the comparative features, and the like; but usually there is a set list of requirements that any number of products will meet. Like many artists, I have had the experience of having a particular work be in great demand, while the other works from the same series don't seem to have the same magical appeal. This is why the personality of the artist, the person behind the work, is often the most important part of the process. The same way I need to be reassured that the motherboard I am buying has been put together with thought and integrity, when art is sold there is often a need to hear the artist's own beliefs and to learn about the processes that went into the work.

The Internet is a potentially powerful forum for showing the person behind the work, whether you're seeking a grant to create a dance performance or you're selling Widgits. My home page (<http://gopher.panix.com:70/11/nyart/Kpage/> or <http://www.panix.com/kgreenb/kghome.html>), for example, has my name highlighted at the bottom. If the reader clicks on my name, they find not only a mix of my .plan and .signature files, but also further links within my various references. I write a column for *Internet World*. Clicking on Internet World takes you directly to the Electronic Newsstand where archives of this magazine and my work may be found. It's a comfortable way to promote myself because others can choose to make that link on their own. It hasn't been forced on the reader, yet the information is there for anyone who wants to see it. Sometimes, I receive mail where someone sees a particular interest of mine and they share a common interest. This may take the form of "I noticed that in alt.artcom you mentioned you make neon. I work with this medium, too. Some of my work uses electronic sensors[el]," and so on. I have had situations where I am very interested in the contact someone has initiated and situations where I am not interested. It is easy to respond quickly via e-mail and either encourage the dialogue or suggest other people or places they might consider more appropriate. Generally, if someone is sincere and the presentation is direct and simple, I find I respect the attempt to communicate with me, even if I am not interested or able to be helpful.

## Letting Others Know

The task of directing others to you or your work is not an easy one. The immensity of the Internet is intimidating. Getting your idea or message across can feel like you're tossing a bottled message into the ocean. Your chances of recognition might be slim, but they're made even more of a challenge by the thousands of others tossing their messages into the sea. Setting aside the idea of e-mailing to everyone (don't try this at home), there are strategies that build a strong foundation that will ultimately help your work to succeed.

The vast diversity of the network becomes comprehensible for each person when individual selections are made and individualized groupings occur. The Internet is a living example of the concept of parallel universes. When I ftp to a NASA's photographic archive because I'm interested in downloading a high resolution photograph of some satellite pictures of some earthly features to test my photo retouching package, I might be next to (in the virtual sense) a geologist who is retrieving the same file for an entirely different purpose. We may have arrived at this site through entirely different paths. The geologist may have been poking around on a university department Gopher. I might be FTPing after seeing an article in an art news group. Someone else might pass through the same site while Net surfing. The three of us have no knowledge of each other's coincidental interest.

Let us suppose I now develop a curiosity about geological formations. I'm inspired to paint a series or create a sculptural environment and I want more information. Now I begin to look for sites that are a little difficult for me to find while the geologist would be on familiar ground, and would understand the language, methodologies, and so on. Maybe the geologist has discovered a need to learn about photo-imaging because greater detail and isolation of physical features has become critical. The geologist now treads into the world of graphic resources, an equally unfamiliar domain. Our paths may anonymously cross again. And there goes that Net surfer.

We both, however, have fortunately interacted a lot with users within our more familiar domains. Owing to the diversity of interests in the department, the geologist finds someone who knows someone who can point a little

more directly toward the appropriate resource. I find someone whose home state has a university known for its geology program. I might even by strange coincidence connect with that very same geologist, and we're both thrilled to discover our mutual ability to assist each other. Within the seeming chaos of divergent paths on the Net, there is order where the many possible common interests intersect.

Strategies for effectively finding your best audience begin with your own small circle of contacts on the Net. Your personal mininetwork probably formed quite naturally and it therefore likely has the strongest foundation. But your group of resources can't help if you don't ask. Seeking advice isn't only a logical starting point but it's a no-pressure approach to letting friends and strangers know about your interests. In some cases it enables others to offer resources to you that they didn't realize you might need. Often as a result, your sphere of contacts widens. In the process of developing new relationships, you may have the opportunity to show a sample of what you do or your product to an entirely new audience. You then may be able to ask for advice about your product or service from their perspective. Does this person think the presentation is effective? Might they suggest the kinds of people or companies who might be interested in this service? Have they had any experience with this kind of service or product? You are not only receiving feedback and possible further potential contacts, but you have made yourself known to at least this person, who may in the future refer someone to you. This is all basic business common sense. The same rules of politeness and brevity apply. Be concise and to the point. Keep in mind that people do not want to scroll through screenful upon screenful of information.

After you have established a slightly larger network of people who have a direct interest in you and what you do, it's important to cultivate their interest. The down side of high speed communication is that things grow old fast. Try to stay in touch with what others are doing. Study online techniques or presentations in areas completely different from yours. Occasionally, through a news group or a mailing, announce a new service that you provide. You now have an online presence to maintain. I liken it to animated neon displays I create in my storefront studio window. I feel an obligation to keep things running, and periodically I make a refreshing change. I keep my outdoor neon clock set to the right time. I have learned over the years that people do notice and do appreciate it.

## Following Up

Don't forget to follow up the responses and requests that you receive. This provides further opportunities to present yourself and your work in greater detail. You also begin to accumulate information as to what approaches work and what information or aspect of your work needs further clarification. Your new contacts possibly provide more degrees of expansion for your growing personal network. And lastly, don't overlook nonnetwork related activities. We often get so involved in trying to locate someone or some service online that we fail to simply use the phone book. Similarly, the fact that you are working online does not preclude using the mail. A brochure that can be held and leafed through and grabbed off the shelf without having to boot up and then go online to retrieve it certainly makes up for its lack of Internet chic.

## Internet-Based Art

In the first part of this chapter, we examine the communities of artists and art users on the Internet and their activities. In this section, we look at artwork that is actually *on* the network and art that actually *is* the network. Again, as mentioned earlier, I am primarily focusing on visual art which tends to be a predominant area in online and off-line life. Keep in mind, however, that not all forms of art are represented on the network and that some forms of art are evolving. There may be art forms we have not yet seen which may eventually be so well suited to network interactivity that these forms overshadow the existing media. There are also areas where art forms intersect and overlap. GIFs, for example, may be used to display choreography. We will look at art forms found on the Internet and the Internet itself as an art form.

## Art Forms Native to the Internet

Many art forms that have existed in the world of desktop computing have made an easy transition to the Internet. Just about any file format can live on the Internet, and a vast array of file types and program structures can be found. The examples covered next are among the most common and most universal.

## **Image Files: ASCII, GIF, JPEG, and More**

It wasn't long after the first line printers were outputting text-based information that the desire arose for some type of graphic display. ASCII art was the earliest visual computer art form, and it has enjoyed a remarkable longevity. Art work that uses the shapes and locations of text and other ASCII characters forms the basis of graphic signatures, ranging from a simple smile-sized creation all the way to immensely complex pictures. Figure 22.1 is an example of a relatively simple ASCII art file.

### **[Figure 22.1. An ASCII art picture.](#)**

ASCII art has grown in its sophistication, as well. Programs exist, for example, that can sample a high-resolution GIF image and then choose ASCII characters that most closely resemble small clusters of pixels, and thus display a detailed gray-scaled image on a text based screen. Figure 20.2 is an example of this style of a gray-scale converted to ASCII file. Newsgroups and archives of ASCII files exist throughout the Internet, and it is an art where the only requirements are a text editor and some patience.

### **[Figure 20.2. A gray-scale drawing converted to ASCII.](#)**

A close cousin of ASCII art is ANSI art. Originally, this art form applied to desktop screen effects that used the extended character sets. These character sets redefined character keys as filled blocks, double lines, corners and so on. Also included are the ANSI escape sequences that cause a character to blink, display in higher intensity, clear a line of text, position the cursor, and so on. Because of the nature of how a vt100 screen refreshes its display, ANSI art has become the low budget animator of the Internet. Clever use of escape sequences (programs exist to facilitate this) allow the creation of animated moving text banners and ASCII cartoons.

CompuServe's GIF standard is a reliable and lossless format that allows the exchange of almost all types of graphic files native to specific computers or graphics packages. A user can, for example create a file in MacPaint and then either save it or convert it to a GIF. It can be uploaded to an archive and downloaded to an Amiga or a PC for viewing or further conversion. GIFs can be mailed through the use of uuencoding. With the proper software and a decent band width, a GIF can be viewed online. Mailers that send and receive MIME understand GIF files, as do Gopher and WWW browsers.

As the demand for visual display increases and as modems get faster and more people connect their computers to the Net via SLIP and PPP, the graphic standard is expanding. Enhanced text effects and fonts and a variety of graphic standards are being supported. Popular formats such as JPEG, compressed image files containing enormous amounts of information, are being exchanged.

## **Sound and Media Files**

Static graphic imagery and text aren't the only medium available on the Internet. Sound, and in particular, music has had an enormous presence on the Net. There are dozens of archives where sheet music and chords are available for downloading. There are hundreds of downloadable files containing sound samples for all systems including au files that can play on the Sun /dev/audio device. Sophisticated tools that browse the Web can play sounds embedded in a document in the same manner they display images. A new site named Cirque de la MAMA (<http://lancet.mit.edu/cirque/introduction.html>) is a multimedia and multidisciplined showcase area.

Artists who contribute to this site make use of combined sounds and imagery.

Full-scale compositions are stored as MPEG files, a fairly lossless form of compression. One of the most interesting applications for music is NetJam, a program developed largely by Craig Latta (finger latta@xcf.berkeley.edu for one of the most amusing and interesting, if not longest .plan files I have seen). NetJam allows a virtual real-time jam session between musicians in remote locations. A synchronization server times the information so that the musicians do, in effect, interact real-time. There are also archives of existing compositions where a user can work in a ones-own-time frame. NetJam information files and program documentation are available via FTP at xcf.berkeley.edu /pub/misc/netjam.

Video is emerging on the Net as well. There are several sources of video clip files and software for working with video. Programs such as CU-SeeMe allow real-time video conferencing across the network. This software (for Mac and PC) is publicly available via Gopher or FTP at gated.cornell.edu. A video artist online not only can present work to be viewed by others but also can interact with several other video artists through the use of reflectors.

### **Interactivity: OTIS, Muds, MOOs, and More**

It would be difficult to trace the early roots of interactive art because all art is actually interactive on some level. A Shakespearean actor makes an aside to the audience and it is, after all, the audience's response which drives the actors. In computer terminology, the distinction between multimedia and interactivity has become somewhat cloudy. To me, the concept finds its origins with artists such as DuChamp, who made works that were meant to be touched and handled or Julian Beck, whose Living Theater produced works that were meant to trigger individual responses from the audience. Sound and light pieces, where an observer's movements trigger sensing devices that then drive light, sound, or movement, began to evolve in the 1960s. Environmental playgrounds exist throughout the world today. Unique to the online culture are muds, MOOs, MUSHes, and interactive communities such as MUSE that can, and should, be viewed as an art form in and of themselves. (For a more complete discussion of muds see Chapter 11, "Online Entertainment.")

Multiuser interactive environments, Adventure, and MultiUser Dungeon, being perhaps the oldest and most famous text-based object-oriented games, create an internal virtual landscape and provide characters, objects, and situations that fill a temporal landscape of action, inaction, chance occurrence, and skill. In the Renaissance, an artist would have to capture the whole gist of possibilities within one static painting. Volumes have been written analyzing the various postures of the Saints in "The Last Supper," for example. The programmer still must create the characters and the scenery with care, balance, and insight. An entire culture then blossoms around the game where users exchange strategies. To a certain extent, the game is now beyond the hand of the author.

Not long ago in New York City, and in Chicago as well, there were art installations that were mini golf courses designed by artists and meant to be played upon. At the vastly exciting Jonathan Borofsky installation at the Whitney Museum in the mid-eighties, among the enormous body of static and kinetic works was a hand-painted Ping-Pong table complete with paddles and ball, and the museum guard's responsibility in this case was to retrieve or replace the occasional overshot or overworn ball. As more systems have allowed direct access to individual public directories, the individual home page has become a personalized interconnective tool. Hand selected favorite sites as well as local gossip/news combine with the author's thoughts, comments, and views to become a type of interactive environment to play within. Often they combine pictures and sounds.

MUSE (Multi User Simulated Environment) extends the multiuser role-playing game to where an entire community exists in real-time. Citizens may enter rooms, embark upon paths, manipulate objects, and create realms for others to explore. The participant doesn't merely play through a pre-programmed set of rules and locations. Objects in space become a medium for creativity, and it's possible for user-citizens to orchestrate scenarios as well.

From a commercial point of view, interactivity will be the primary means by which the consumer is pulled in, analyzed, fitted, given a chance to test drive the product around the block, and billed. All of the challenges that go into making a MUD or a MOO are contained in this growing field. Interactive artists have been dealing with many of the same issues as well, though usually from a conceptual or aesthetic perspective. How, for example, do I create sounds that when triggered by a person's movement, allow the person to know what particular movement at what moment triggered the sound? At the same time, how do you handle the overload of many people in a space causing the sensors to be constantly triggered? And if you successfully trim the sensitivity way down, how do you handle the times when one person or no one is on the site? The artist must not only conceive of the possibilities but he or she must also create a fluid, purposeful transition between these different states. Finally, it's not just the look and feel of the work, but an idea is also being conveyed.

## **Creative Writing**

Let's not forget the time-honored art of writing. The Internet forces each of us to be verbal. We "meet" people online on a daily basis who we may never actually physically meet. Most of us don't have full fledged video conferencing, and therefore we're forced to express ourselves through the written word. Paradoxically, as the technology has so rapidly grown, we have returned to the pre-twentieth century skill of letter writing. I have found that I am much more likely to e-mail a letter to a complete stranger than I am to phone someone I don't know. I'll more readily post my comments to an online discussion group than I would break into an actual group in "real" life. How many of us would hold a sign up in a busy train station reading, "Can someone help me with adding memory to my PC," and yet the relative anonymity of the Net allows us to admit our weaknesses to the world at large and ask for help.

## **The Internet as an Art Medium**

Artists find it hard to resist any new medium. Even the most traditional painter can become interested and even quite enthused over a new kind of paint—as was the case when acrylics were introduced. Artists have been involved with the capabilities and technologies of computers ever since this medium became accessible. Many of the oldest SIGs on the earliest online systems are art interest groups. Sound artists use MIDI. Video artists edit on computers. Special effects on stage are computer controlled. The communications media have also been extensively exploited. An interesting precursor to the interactive era was video artist Nam Jun Paik's New Years' Eve piece in the 1980's. 24 locations around the earth separated by an hour's time zone difference and communicated real-time via interactive uplinked video. Internet based art has been initiated, or rather instigated, by people whose fields are not generally art. John Romkey's "Internet Toaster," driven by its two Internet based commands "push" and "pop" (which are assembler program instructions) has spawned a genre of Internetic art. A work that which went online in June 1994, "The File Room"

(<http://fileroom.aaup.uic.edu/FileRoom/documents/homepage.html>) uses the Web to connect to and examine censored works of art in files created and stored by contributors in several countries throughout the world.

In the fall of 1994, The Electronic Cafe finds its U.S. home at New York's Kitchen (kitchen@panix.com), a venerable performance art and multimedia exhibition space that has survived since the early 1970's. The New York strong site, as it is called, joins several other cities around the world to create a setting where artists around the world can create and share interactive multidisciplinary works. A dancer in California, for example, may perform in real-time with a musician in Austria. Cyber-culture blossoms here as local patrons of the cafe have access to the teleconferencing systems as well. Countless virtual pubs and cyber-collectives exist throughout the world where users meet to discuss and collaborate.

## **Internetic Installations**

The issue of how to design, create, and install "internetic art" is a challenge. Even the most basic idea of

displaying one's visual work or offering a sound piece has to have purpose and intrigue for the users of the Net. Typically, I find that even in a text based environment, it's difficult to attend to every detail that an author has provided. At the moment, owing to the relative slowness of the average user's hardware combined with the large amounts of data that comprise media files, it is sometimes a torturous process to read a page that contains many graphics. This is made particularly painful when an accidental keystroke or misfired mouse click starts an irreversible retrieve of an enormous file. In the normal realm of publicly displayed art, even the largest piece is rarely an obstacle in this sense.

As the "groundskeeper" of New York Art Line, the Panix art Gopher (URL: <gopher://gopher.panix.com:70/11/nyart>), I have found that being on the "server" side is not easy. Before putting my work on Gopher, I experimented with the idea of an in-house BBS on Panix using Lynx as a browser. I had a number of users try out my menus, and I was amazed at how often the unanticipated path was chosen. I felt, for example, that a first line entry that said WELCOME DESK would be the obvious first choice and, therefore, the place for orientation info and the primary "hub." My beta users consistently bypassed this option and went directly to an area deep in the menu that I named "The Dada Base" and where, unfortunately, I did not yet have any material.

As my work moved to Gopher, I became aware of the need for consistency in an ever-changing world. If I was interested in having users become aware of local information that I served, I had to maintain a consistency to my directory structure or else any other established Gopher's link to that area would be broken. This meant that in many cases, you have to live with what you've done until you create a clever way to redirect to your new areas. At the same time, you need to let users know that your Gopher is not static. Some means for drawing users beyond the consistent look of the main menu is often needed.

My Gopher area now also has a link to my home page where I get to offer the freer form of the Web to users who are browsing with html readers. The home pages of the Web can, to some extent, get by on personality more than consistency. While things do change in the world of Gopher, there is an element of predictability once you're familiar with a particular type of Gopher, such as Arts Gophers. After a while, they start to reference each other and as I look for new links, I frequently find the same core group of Gophers and even similar menu headings. As I embark into the Web, it seems to be an ever-changing and growing resource. One is not better than the other. They are different. Gopher is more like the catalog of your library, while Web is like the magazine stand. The Web, Gopher, FTP, and telnet are the current primary Internet realms where artists can presently create online public installations.

## **There's No Place Like Home Page**

I've mentioned a few times already that the World Wide Web is growing at an incredible rate. On average, I learn about a new art resource or project on the Web just about every day. The Web has several distinct advantages over Gopher and even telnet and FTP. On the one hand, it's possible to present an attractive document complete with fonts, effects, imagery, and sound. Second, the organizational aspect of your material need not be as a static list. It is possible to place hypertext references within a freewheeling prose. Third, to some extent you have greater control over your unique way of presenting links to the Web. In Gopher, as an example, if I organize a directory where I've researched and found many Usenet groups that relate to art, any other Gopher can simply point to this directory, give it a new name, and effectively now "own" this directory, too. I don't happen to mind this because that's the nature of Gopherspace. The Web, however, has a little bit more ability to be proprietary—a reference to another document at a site can be nested so that the initial home page is the only path to the work. Of course, any URL reference that's accessible via the Web is up for grabs. Finally, the fourth advantage is that Web browsers can run a Gopher, telnet, or FTP session. Additional advantages exist in the individual software. Lynx, for example, has the ability to mail a comment to the owner of a document.

The home-spun nature of home pages have made them an attractive option to many people looking to put

information on the network. The only disadvantage I see is that the Web is mapped in a somewhat free form way. This makes it both an adventure and a challenge for those looking to present work and those looking to find work. It does seem, though, that good ideas have a way of getting through and surviving. Once a home page begins to be cross referenced by one or two major sites, the awareness and participation, if any is called for, grows geometrically.

## **Electronic Cafes and Cyber-Culture**

The comforts of personalized electronic space are not limited to an individual's home page. The concept of a community hangout on the Net is about as old as the network itself. After all, we are talking about a multiuser environment. Long-time users of the Internet have grown accustomed to and, in fact, have shaped the network so that it is almost impossible to spend all of one's time in isolation. Even the most hermitted lurker still must be aware of the ever present world of exuberant interacting personalities. It's difficult not to find a personal touch within a program or a site. Just as the salon and cafe have been institutions in the art world for the past century and well beyond, the Internet has always had SIGs, news groups, IRC, multiuser games, and so on.

ECHO uses the imagery or analogy of electronic salons where artists gather in virtual space to interact and converse. The WELL ([well.sf.ca.us](http://well.sf.ca.us)), which is the direct electronic offspring of The Whole Earth Catalogue, structured itself as a series of moderated communities. Delphi ([delphi.com](http://delphi.com)), the first major commercial provider to offer full Internet access, has had a variety of popular SIGs and a sizable collection of smaller SIGs individually configured by users. Arts Wire on [tmn.com](http://tmn.com) has structured itself as a series of conferences for its community of artist users. Borrowing from a host of concepts that have always existed, they have formalized the manner in which users can communicate with each other. A user may initiate a conference with one or several other users. The conference may be formatted as private or public. The larger commercial providers, CompuServe, AOL, Prodigy, and Genie, have had a host of community forum areas both large and specific in scope. The wonderful collection of local Freenets has always had its community centers and town halls. Mail listservers have provided still another venue for exchanging ideas and contacts.

A cafe is usually an environment where the subject matter is open. The draw is that a group of creative personalities are in one place at one time, and their interactions will generate at least an entertaining evening. Talk sessions and Internet Relay Chat have allowed a deeper enhancement of the slightly more static Usenet and other interest areas. However, the concept of interacting online has been expanded with facilities such as NetJam where musicians create combos and play through MIDI-interface across the Net. The OTIS project has allowed artists to interact by manipulating each other's work. These sites allow performers and audiences to share the archived works and to be immediate participants. They are comparable to open microphone night at a local performance space. A number of ECHO derived systems exist where energetic gatherings of artists, musicians, and cultural buffs meet and chat online on a regular basis. There are real-world events springing from network meetings. My local access group, Panix.com, has picnics, outings to movies, and occasional parties. Commercial enterprises such as pubs, Laundromats, and cafes are placing terminals on their premises. It would not surprise me one bit to see in the near future an Internet version of the diner table jukebox.

## **Curating and Criticism on the Internet**

We have looked at the people who make and appreciate art. We have also explored some of the kinds of art found on the Internet. An important area of interest that bridges the gap between making and appreciating art is the critical evaluation of art.

## **Understanding What You See**

I have found that one of the most challenging areas in the brave new world of the Net is any type of rational or critical evaluation. When I performed scientific research in a previous stage of my life, it was a fair guess that

through the use of abstracts combined with diligence and scholarly analysis, one could come reasonably close to what one might call an educated theory about some issue. In the sciences, it is necessary to take the next step of proposing a hypothesis and then setting out to show that empirical cases where your hypotheses are contradicted are extremely rare, if not nonexistent. The humanities and the arts have equally stringent techniques for understanding and determining the critical or cultural validity of a work, although the process uses comparative reason rather than mathematics. On the Internet, even with high speed and high sophistication tools, it is difficult to assign a set of critical values to the information received. If I search, for example, a variety of online databases for causes of the Russian Revolution or even the cause of a thunderstorm, I have no way of determining if my search exhausted what is on the network in these areas or if the material retrieved has "true" validity.

Evaluating the arts is equally challenging. I will leave the issue of good art/bad art to others. What works are presented online. How they persist is, for the moment, based on individual initiative combined with either the willingness of an access system to devote space to what the artist is doing or by the artist paying for disk space. The most basic form of relatively nonjudgmental art critique or curation begins with the activity of cataloguing. I have not yet found any source on the network that attempts to organize and categorize styles of art work and types of media accessible throughout the Internet. This field is in its infancy. There are some galleries and even some museums, notably Reiff II (<http://www.informatik.rwth-aachen.de/Reiff2>), that have repositories that accept images and proposals for consideration. The evaluation in this case is performed by the museum's curatorial staff. E-zines are online—they feature professional critique and news groups where popular critical dialogue takes place.

## **Media and Cultural Critique**

Every medium has its critics and the world of online business, pleasure, research, and culture has not escaped the critic's eye. In fact, the very public and shared nature of events, tools, and media on the Internet make it difficult for any well used tool or heavily visited site not to become a subject in and of itself for discussion, praise, or occasional ridicule. System administrators can be targets in much the same way our political leaders are lambasted and lampooned. Entire groups have formed around either appreciation, such as the Mystery Science Theater 3000 fan groups or deprecation, such as the Jihad Against Barney. Beyond the simple act of expressing one's momentary views about a system's service or a software's performance, there is an important contribution from the critic's corner. We are not only given some order and a preliminary filtering for our tastes, but some of the aspects that drive the creative process are sometimes revealed, and deeper questions about purposefulness, methodology, and ideology are raised.

## **The Politics of Critique**

Criticism almost always has a political stance beneath its surface. One of the more fascinating aspects of the Internet is the fact that we have a rather large and not so homogeneous grouping that extends well beyond physical and cultural boundaries getting by in a rather remarkable state of harmony. There is everything on the network from punk rock to philosophy. There are Christian groups, Islamic groups, Jewish groups, and so on. In my experience, the level of conflict is well below the expected norm, where all of these groups are physically as close as Cyberspace makes them appear to be. The explanation, I think, lies in the shared adventure of exploring and interacting on the Net and the protective layers of virtual space. Further, the mechanics of the multiuser environment make our interactions nondestructive to each other's virtual territory, and often we are able to enter a dialogue or some other form of contact within the safety of our own personal framework of time.

## **The Audience Participant**

Earlier, I offered four major groupings in the online arts community, and I referred to the third as all of us, or the audience. We are the end users. As the era of interactivity emerges, we are the players as well. Interactive artists

refer to participant—viewers or audience—participants. Now, not only do we, as an audience, influence the work by our paying attention to it, but further by acting upon it. It is an unusual predicament for many people. Often upon viewing a work of art, especially in the presence of others, we feel the need to understand or put a label on our experience. I have been to countless exhibitions where people attempted to express what they saw in an abstract work of art. Sometimes, however, the artist does not have a concrete figurative image hidden or contained in the work. This type of art forces the audience to reach deeper within for a less concrete and more emotional or spiritual, and sometimes even physical, reaction.

The responsibility of the audience in an interactive work can go beyond this internal experience. There are works that really do very little until a viewer-participant becomes more fully engaged. It is a little bit like that first time you go online on a UNIX system, and you have a blank screen with a sole prompt waiting for your input. Just as the creative process of the artist sometimes does not begin to flow until there is the direct working and manipulation of substance, a truly interactive work of art will derive its shape and ultimately its meaning as a result of user participation. The point is to enjoy and learn from the interaction. Whether we are discussing an esoteric virtual art environment or a presentation selling a new car, how the audience experiences the work and how the work then responds to the audience's actions will determine its ultimate success.

## **What Artists Contribute to Understanding of the Net**

We have examined the community of artists and audience, and we have explored the art that is on the Internet. We also looked at the Internet as an art form and some of the critical analysis that is a part of the network. In this section, we look at the deeper role that artists play in our ongoing development and understanding of the communications era.

### **Visualizing Cyberspace**

I've often wondered why an artist or a musician has not yet been sent to space. I am reminded of the late great jazz musician Sun Ra who said something to the effect that he did not mind, because he had already been there. Space has been a government research venture, and most of the tendencies to revel in the profound impact of space exploration have been relegated to creativity on the ground—with the exception of one or two astronauts who shared their deeper feelings. Fortunately, Cyberspace does not require a pressurized capsule and years of training, and for a growing number of artists, artistic vision goes beyond merely the look and feel of a software's decor. Creative vision is not simply the ability to find the most stimulating pose for a model standing next to a car. Artists frequently deal with issues of meaning and responsibility to our inner being. Art helps to elevate the mundane to where its place in the grander picture is at least confronted, if not understood. As artists experiment with the Internet as a medium, a reflective mirror is raised where we all may look at who and what we are trying to be.

When I first began to play with programming languages, I tried to methodically order each task toward an end. I soon began to realize that the same ideas or modules surfaced again and again and thus, I became an object-oriented programmer. As I learned about using libraries, I began to realize that I wasn't really a programmer, but rather a designer. The task at hand was no longer the logic; it was the usefulness and appeal of the interface. The sum total of all of the programs—both on network mainframes and on your desktop computer terminal—can be viewed as one large software interface. The modules are completely interdependent. Whether you use Eudora or Elm, the mail must use the same routing mechanisms. FTP, whether by Mosaic or in text alone, still must rely on a uniform directory and file structure and means of data transfer. The artistry of how the interfacing brings the various content and program elements together, and makes use of them, is an exceptionally significant aspect of our online activities.

### **"Hello World!": Self-Image Online**

Most people are initially shy on the Net. It is difficult at first to comprehend how and when you are visible. I remember thinking that all of Delphi could see me and that the Sysops spent their time watching their users input and output. It was a while before I realized that system administrators have their hands too full to be involved in voyeurism. For a while, I swung a hundred-and-eighty degrees the other way when I began to realize how invisible we are on the Net. The issues of personal and shared space—especially on a multiuser system—are persistent and important. It is, for example, a challenge sometimes to promote one's existence and abilities to the network community without appearing to boast and inappropriately use this very public domain.

I was reading the local news on Panix, my home base, and discovered a discussion around a proposal to have a new news group called panix.user.artist. Delighted to find others with whom I potentially shared an interest, I posted to the discussion my support of the news group. I then added a comment that artists might also be interested in checking out the art area (New York Art Line) that I created on Panix's Gopher. When I returned to this discussion a few days later, I was taken aback by a severe reprimand that this was not the place to promote my work. I dutifully apologized and felt somewhat embarrassed and somewhat indignant. I thought to myself "touchy person" and let it go. A week later I poked my head back into the thread and there was an admission that I had been treated a bit harshly. I then added "no problem" and further explained my motivation for posting the information.

This sort of thing is not new for artists. When you send out announcements for your exhibitions, concerts, or performances to friends and associates, you hope that they view your call, your card, or your brochure as an invitation to an enjoyable and thought-provoking event rather than an annoying display of ego. It is difficult to separate a work of art from its creator, and the creator usually does want and need an audience. Many great works of art possess a clear presence of the person behind the work. Hitchcock placed himself in all but his earliest films. Escher's face and hands were in his works. I think it is important that people see and feel the personality of the artist whether by actual commentary or by the markings left by the creative process. This is why I, and apparently many others, find an appeal in the often personal nature of the home page. We seek evidence of a human underneath things in almost all activities. Art is the discipline that works to bring that humanity to the surface.

### **me.com: Getting Your Work Out There**

We have already discussed a number of techniques for getting your work seen and known on the Internet. There are further issues about the quality of your presentation or installation. First and foremost is an idea that is as true on the Net as it is in real life—be good at what you do or at least be sincere. Kevin's Prairie Dog Town at gopher skynet.usask.ca seems to have had greater staying power than the infamous lawyers who made their mark (I can't help but think of something in bad taste) on the Net. Although their claim was that they were making a sincere attempt to bring certain legal issues to the floor, it is difficult to see their work as any different from Tony Shafrazi's mutilation of Picasso's "Guernica," which he justified as an expression of art.

Determining what belongs or does not belong in any publicly accessed space has never been easy. The argument that graffiti is art, a hotly debated urban issue of the seventies, has been vindicated by the influence of street style in current advertising trends. Pieces of the Berlin Wall, replete with years of politically intense imagery and words, are collector's items. Yet many of us would not be enthused to find our transportation system, our businesses, and our personal property covered with heavily painted scrawling.

A major debate in the art world took place around the "Tilted Arc" in New York City. This public work by noted steel sculptor Richard Serra was commissioned by the federal government for Federal Plaza, a process that entails greater competition and public scrutiny than a military bid. Once the work, a long curved and tilted wall, was installed, an immediate opposition emerged from workers in the Federal building. Arguments ranged from "it's ugly" to a fear that it would fall or that one could become a victim of a mugging behind the wall. For the artist, the issue went beyond simply pride. A major commissioned work by the federal government is an exceptionally

significant stage in even a well-known artist's career. Its permanence and prominence in a public space for time immemorial has an unfathomable value. The debate went to the courts, and the workers prevailed. This was seen as a victory for populist control over the content of art, and at the same time a severe blow to the right of free expression and an artist's right to protect his or her own work.

Graffiti on the Internet seems to be at a minimum. Occasionally, everyone on my access system will receive an unwanted "hey" from a naive user. When America Online began to offer limited Internet access along with limited user support, many long-time Internet users became infuriated with the busloads of "newbies" who appeared to think that every individual and every news group was their new user support. There have been flame gangs who have made it their pastime to attempt to pull the focus of a news group discussion way off of its track, purely for the fun of the exercise.

Some things to think about when seeking to place your work on the Internet include: What groups are interested in the material I am offering? Is it unique? Has the interest for this type of a site or project passed its peak? Has the presence and/or the structure of my work created an obstacle or a mild annoyance for users? Have I designed my project in a flexible manner so that if the need for a major revision arises, I can readily adapt and thereby minimize downtime and any other inconvenience?

## **A look Toward the Techno Future**

In a round table discussion on National Public Radio about the Internet, I caught a remark to the effect that to some, the word "interactive" means a button on your TV that says "buy." We are being inundated daily with commercials showing glimpses of soon-to-be telephone technologies. Radio shows are being cobroadcast on TV. A call-in interactive children's show, Moxie, where a cartoon character can respond spontaneously, has been offered by the Cartoon Channel. As the communications media of TV, radio, computers, and telephone merge, a host of issues (both pragmatic and philosophical) emerge. The organization of all of this information and resource is changing in nature.

We used to watch a handful of TV channels. In your viewing area, you would get to know the flavor of each one—though they all essentially tried to offer the same thing. The main qualitative differences were between local and network offerings. As cable grew, we began to see some degree of specialization such as *The Learning Channel* and *MTV*. We now see a high degree of specialization emerging. We have *The SciFi Channel*, *The Comedy Channel*, *The Romance Channel*, *The History Channel*, and so on. Similarly, on the Internet where Gophers used to point to just about everything, they are becoming specialized—for example, the Ceramics Gopher and the History of Science Gopher. On the Web this has advanced to highly specialized home pages such as the Comet Shoemaker-Levy 9 Collision with Jupiter Information home page (<http://seds.lpl.arizona.edu/sl9/sl9.html>).

It is difficult to predict the state of art on the Internet even five years down the road. While analogies can be drawn from the development of home and business computing over the last decade, much of what lies ahead has no previous foundation. It would be somewhat like being able to predict the existence of WinGopher years before Gopher or even Windows existed. The rate of expansion on the Internet has been accelerating so dramatically that I would not be surprised to see it as a subset or parallel set to an equal or considerably more powerful system. Many of the obvious commercial ventures are already being suggested. That ad on TV where a mechanic or a doctor receives long distance assistance can be easily substituted with an interior designer viewing your living room and instantly displaying custom tailored computer generated designs. A make-up artist or a clothing designer might work with your features and show the new you on-screen.

Will pure art have to compete with a self-indulgent, commercially-driven interactive media, or will channels exist to promote and support experimental ideas? Will the experience of art become removed from the public through confinement to a video display or a printed output, or will it become closer to the public by being more readily

accessible? The world has a handful of serenely beautiful sculpture gardens. These are very special places where children and adults can walk within or upon art works and thereby experience the artist's vision in an all encompassing way. Would a virtual tour through a garden satisfy or compel? The ultimate challenge for future artists in this realm will be to bring experiences that are real to what will be otherwise a cerebral plane or viscerally stimulated plane.

## Pointers to Art Resources on the Internet

These sites are of interest to artists, people who work in the art related fields, and people who enjoy the arts. Many of these sites are stimulating and enjoyable to just about everyone. Of course, as you probably know, sites change on the Internet and they are not always active. I even had a site close down and disappear forever while I was in the middle of transferring a file. The following are generally known to be in service at the time of this writing. This listing is by no means complete. I have included sites that will lead you to further arts resources, databases, images, sounds, and more. Also on this list are a few sites that are more esoteric and that demonstrate the diversity in this field. I apologize in advance for any sites that may leave you hanging in dead virtual space. I have made additions and deletions just before we've gone to press but I cannot predict which of these sites will be in operation at the time of your connection. The constant change on the Internet can be frustrating, but it also brings about new and interesting additions every day.

### General Art Resources on Gopher

`gopher://blackstar.com:70/1/`

`gopher://echonyc.com:70/1/1/Cul`

`gopher://english-server.hss.cmu.edu/11ftp%3aEnglish%20Server%3aArt%26Arch%3a`

`gopher://gopher.kspace.com`

`gopher://gopher.mountain.net:70/1/1/`

`gopher://gopher.panix.com:70/11/nyart`

`gopher://gopher.panix.com/1/1/photography`

`gopher://gopher.tamu.edu:70/1/1/.dir/comics.dir`

`gopher://gopher.well.sf.ca.us:70/1/1/Art`

`gopher://tmn.com:70/1/1/Artswire`

`gopher://gopher.uwp.edu:70/1/1/pub/music`

### General Art Resources on World Wide Web

<http://www.acs.appstate.edu/art/>

<http://www.art.net/>

<http://artn.iit.edu/gallery.html>

<http://fire.kspace.com>

<http://lancet.mit.edu/cirque/cirque.html>

<http://www.lib.virginia.edu/dic/class/arh102/>

<http://www.mcs.net/~wallach/arthole.html>

<http://mistral.enst.fr/~pioch/louvre>

<http://nearnnet.gnn.com/wic/hum.toc.html#histart>

[http://sunsite.unc.edu/expo/ticket\\_office.html](http://sunsite.unc.edu/expo/ticket_office.html)

<http://pubweb.ucdavis.edu/Documents/Quotations/web/pretzel.html>

<http://www.uky.edu/Artsource/artsourcehome.html>

<http://www.wimsey.com/anima/ANIMAhome.html>

## **Images, Sound, Software, Databases, and FAQs**

<ftp://oak.oakland.edu/pub/msdos>

<ftp://sunsite.unc.edu/pub/multimedia>

<gopher://allmusic.ferris.edu:70/1/>

<gopher://amanda.physics.wisc.edu:70/1/1/>

<gopher://gopher.archive.merit.edu:7055/1/1/atari/graphics>

<gopher://gopher.archive.merit.edu:7055/1/1/mac/graphics>

<gopher://gopher.archive.merit.edu:7055/1/1/msdos/graphics>

<gopher://cs4sun.cs.ttu.edu:70/1/1/Art and Images>

<gopher://cs4sun.cs.ttu.edu:70/1/1/Music and Sound>

<gopher://finfo.tu-graz.ac.at/11SIGGRAPHbib>

<gopher://merlin.etsu.edu:70/1/1/AmigaArchives/Aminet>

<gopher://sunsite.unc.edu/pub/multimedia>

[gopher://twinbrook.cis.uab.edu:70/1/asciarc.70](http://gopher://twinbrook.cis.uab.edu:70/1/asciarc.70)

## **Selected Arts Usenet Groups**

alt.artcom

alt.architecture

alt.binaries.multimedia

alt.binaries.clip-art

alt.binaries.pictures

alt.binaries.pictures.ascii

alt.binaries.pictures.fine-art.digitized

alt.binaries.pictures.fine-art.graphics

alt.binaries.sounds.midi

alt.binaries.pictures.fine-art.digitized

alt.binaries.pictures.fine-art.graphics

alt.binaries.sounds.music

alt.emusic

alt.exotic-music

alt.radio.networks.npr

alt.tv.liquid-tv

bit.listserv.film-1

misc.writing

rec.arts.dance

rec.arts.fine

rec.crafts.jewelry

rec.crafts.metalworking

rec.crafts.misc

rec.crafts.textiles

rec.crafts.quilting

rec.folk-dancing

rec.music.info

rec.music.compose

rec.video.production

sci.engr.lighting

## **E-Zines, Databases, and FAQs**

[ftp://dartmouth.edu/pub/ICMA\\_Library](ftp://dartmouth.edu/pub/ICMA_Library)

[ftp://ftp.netcom.com/pub/unitcirc/unit\\_circle.html](ftp://ftp.netcom.com/pub/unitcirc/unit_circle.html)

<ftp://rtfm.mit.edu/pub/usenet-by-group/new.answers>

gopher://echonyc.com:70/11/Music/MO/MWorks

gopher://gopher.cic.net:70/11/e-serials/alphabetic/s/sound

gopher://gopher.cic.net:70/11/e-serials/general/art

gopher://dewey.lib.ncsu.edu:70/11/library/stacks/arch

gopher://gopher.eneews.com:70/11/magazines/category/culture

gopher://gopher.tc.umn.edu:70/1/1/Libraries/magazine

gopher://gopher.well.sf.ca.us:70/0/0/Art/art.fbi

gopher://gopher.wired.com:70/1/1

<http://www.cis.ohio-state.edu/hypertext/faq/usenet>

telnet://ursus.maine.edu (Choose B then Choose 2)

## **Art Education and Other Art Related Information**

gopher://ericir.syr.edu:70/1/1/Lesson/Miscellaneous

gopher://fatty.law.cornell.edu:70/1/1/uslaw/copyright

gopher://tmn.tmn.com:70/1/1/NAEIN

## Cybercultural Links

<ftp://english-server.hss.cmu.edu/EnglishServer:Cyber>

gopher://metaverse.com:70/1/

gopher://techno.stanford.edu:70/1/1/raves

<http://kzsu.stanford.edu/uwi.html>

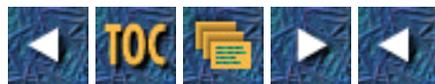
<http://www.panix.com>

<http://www.panix.com/kgreenb/kghome.html>

<http://sailfish.peregrine.com/ww/welcome.html>

<http://sccs.swarthmore.edu/jahall/index.html>

<http://www.lighthouse.com/~ake/DepthProbe/index/home.html>





- [A](#)

- [Uniform Resource Locators \(URLs\)](#)

- [--by Max Metral](#)
- [HTTP and FTP](#)
- [Gopher](#)
- [Telnet](#)
- [News](#)

- [Standard Fare](#)

- [The Home of the World Wide Web](#)
- [Mosaic Demo Document](#)
- [NCSA What's New Page](#)
- [The Best of the Web '94](#)
- [Weather Data](#)
- [Weather Images](#)
- [World Wide Web for Dummies](#)

- [Meta-Information: Finding What You Want](#)

- [CUI's W3 Catalog](#)
- [Destination Unknown](#)
- [EINet Galaxy](#)
- [Special Internet Connections](#)
- [The World Wide Web Virtual Library](#)
- [Yahoo: A Guide to WWW](#)

- [People's Hot Lists](#)

- [Christian Bartholdsson](#)
- [Jeff Van Dyke](#)
- [Andy Eskilsson](#)
- [Matthew Gray](#)

- [Web Robots](#)

- [WebCrawler](#)
- [World Wide Web Worm \(WWWW\)](#)
- [WWW Home Pages Broker](#)

- [Computer Companies and Information](#)

- [Apple](#)
- [Microsoft](#)
- [Novell](#)

- [Silicon Graphics](#)
- [Electronic Commerce](#)
  - [CommerceNet](#)
  - [List of Commercial Web Services](#)
  - [Internet Shopping Network](#)
- [Government](#)
  - [California Bay Area Governments Online](#)
  - [FBI](#)
  - [House of Representatives](#)
  - [Senate](#)
  - [Library Of Congress](#)
  - [NASA](#)
  - [The White House](#)
- [Graphics/Visual Exhibits](#)
  - [Dinosaur Exhibit](#)
  - [The Exploratorium](#)
  - [LEGO](#)
  - [Silicon Graphics](#)
  - [The Vatican Exhibit](#)
  - [The Virtual Hospital](#)
  - [The WebLouvre](#)
- [Online Publications](#)
  - [Books Online](#)
  - [Dr. Fun](#)
  - [Encyclopedia Britannica](#)
  - [Global News Network \(GNN\)](#)
  - [Mondo 2000](#)
  - [PowerPC News](#)
  - [Wired](#)
  - [VIBE](#)
- [Schools and Universities](#)
  - [Carnegie Mellon University](#)
  - [MIT](#)
  - [Stanford](#)
  - [Honolulu Community College](#)
- [Sound/Music](#)
  - [Internet Underground Music Archive \(IUMA\)](#)

- [RG](#)
- [Wired Radio Stations](#)
- [Artist Information](#)
  - [Tori Amos](#)
  - [Anything Box](#)
  - [Beastie Boys](#)
  - [Eric Clapton](#)
  - [Jane's Addiction](#)
  - [Billy Joel](#)
  - [Madonna](#)
  - [Nirvana](#)
  - [Wilson Phillips](#)
  - [Frank Zappa](#)
- [Sports](#)
  - [Sports on the Web](#)
  - [Baseball](#)
  - [Basketball](#)
  - [Fantasy Baseball](#)
  - [Football](#)
  - [Soccer](#)
  - [Surfing \(Water-Based\)](#)
- [Stocks/Finance](#)
  - [Security APL and Data Transmission Network](#)
  - [EDGAR Dissemination Project](#)
  - [quote.com](#)
  - [MIT AI Lab Stock Project](#)
  - [NETWorth](#)
- [Miscellaneous](#)
  - [Boston Computer Society Internet SIG](#)
  - [Cardiff's Movie Database](#)
  - [Fax Gateway](#)
  - [htMUD](#)
  - [Usenet FAQs](#)
  - [Why the Web Sucks](#)

## A

# Uniform Resource Locators (URLs)

[--by Max Metral](#)

*Uniform Resource Locators* are gigabytes of information available via the World Wide Web. The problem is, there's no master index, no yellow pages, and no road map. As a result, it can be quite difficult to find a specific piece of information. Furthermore, there is a lot of incredible work out there that may go unnoticed unless you know where to look. Here's a categorized list of some URLs that contains useful and/or entertaining information. As always, these are just starting points. The only way to really catch the wave is to jump from these pages to others you may find interesting.

Most of the URLs presented here reference HyperText Transfer Protocol (HTTP) sites. URLs can reference many other types of data as well. The general structure of URLs is as follows:

*Protocol\_Type:Resource Location*

*Protocol\_Type* can be one of the following strings (among others):

|        |                                |
|--------|--------------------------------|
| http   | HyperText Transfer Protocol    |
| ftp    | File Transfer Protocol         |
| gopher | Gopher                         |
| telnet | telnet                         |
| news   | Netword News Transfer Protocol |

Each protocol type has a slightly different format for its *Resource Location* portion.

## HTTP and FTP

*Resource Location* = *//server name/document path*

HTTP and FTP share the same format in almost all cases (see the URL specification for the differences). For example, to refer to document `people/bill.html` on server `biz.buzz.com`, you use the URL `http://biz.buzz.com/people/bill.html`.

## Gopher

<Resource Location> = *//<server name>/<type><selector string>*

The *type* field is a single character that is used by the gopher protocol to determine the type of the document. The default type is 1. The rest of the path is the document or resource selector string, as described in the Gopher+ protocol. For example, to access the root document at server `gopher.biz.com`, you use the URL: `gopher://gopher.biz.com/1`.

## Telnet

*Resource Location* = *//username:password@server name*

On most systems, opening a Telnet URL merely launches an external Telnet application and allows you to connect

from there. For example, to connect to biz.buzz.com as coolguy, you use the URL: telnet://coolguy@biz.buzz.com.

## News

*Resource Location = newsgroup:first article-last article*

Most browsers will have you set a preference that tells it which server to query against for News URLs. Each server has its own identifiers for each article, so it's usually best to first ask for a list of current articles by omitting the <first> and <last> keywords. For example, to ask for the current articles in alt.flame, you use the URL: news:alt.flame.

A more detailed description of URLs can be found at:

[http://info.cern.ch/hypertext/WWW/Addressing/URL/URI\\_Overview.html](http://info.cern.ch/hypertext/WWW/Addressing/URL/URI_Overview.html).

## Standard Fare

There are several sites that almost everyone who has browsed the Web has seen. Because of this, expect traffic on these sites to be heavy and access times to be slow.

### The Home of the World Wide Web

<http://info.cern.ch/>

CERN (translation: European Laboratory for Particle Physics) is the originator of the World Wide Web project. To find information about the origins of the Web and information about obtaining servers and clients, read the following.

### Mosaic Demo Document

<http://www.ncsa.uiuc.edu/demoweb/demo.html>

You just unzipped, untar'ed, or unstuffed your version of Mosaic and you want to see what this puppy can do. Go to the demo document and you'll be pleasantly surprised. Of course, you may be a bit disappointed that 100,000 other people are trying to be surprised at the same time (so be patient).

### NCSA What's New Page

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/whats-new.html>

The National Center for Supercomputing Applications was the birthplace of Mosaic, a huge catalyst in the popularity of the Web. It has become the standard place to "announce yourself." When new servers come online, many of them will show up in the What's New page.

### The Best of the Web '94

<http://wings.buffalo.edu/contest/awards/index.html>

The Academy Awards of the Internet, the Best of the Web, has only been around since 1994. However, this has been an explosive year in Web growth, and there are many incredible sites out there. Some of them are honored here. You can also find information about the founders of the Web project and the Mosaic project.

### Weather Data

<http://www.mit.edu:8001/weather/weather>

MIT's server provides textual data from the National Weather Service. If you don't have forms, you can ask for a city by appending ?CITY to the URL. For example, to retrieve San Francisco, you ask for <http://www.mit.edu:8001/weather/weather?SFO>.

## Weather Images

<http://rs560.cl.msu.edu/weather/>

WX-MAP can retrieve surface map images (in GIF form) for many stations across the world. Via Mosaic Forms, you can request different types of maps. Figure A.1 shows a sample map.

**[Figure A.1. A sample satellite image from rs560.cl.msu.edu.](http://rs560.cl.msu.edu)**

## World Wide Web for Dummies

<http://www.mit.edu/people/rei/wwwintro.html>

This page teaches you the basics of the Web, without all the jargon.

## Meta-Information: Finding What You Want

The biggest problem and asset of the Web is the fact that there is no central "root." There is no definitive source at which you can search the Web. However, several ways of finding information have arisen:

- *Naming conventions.* A simple heuristic for finding organizations is to add www to their domain name. For example, to find Stanford's web site, try www.stanford.edu. If you're not sure of an organization's domain name, you can use *netfind* or some of the other tools described elsewhere in this book.
- *Hand-crafted indices.* Several organizations are in the business of constructing indices to the Web. Also, many individuals have created their own favorite collections of pages.

## CUI's W3 Catalog

[http://cui\\_www.unige.ch/w3catalog](http://cui_www.unige.ch/w3catalog)

From the page itself: "W3 Catalog is a searchable catalog of W3 resources build up from a number of manually maintained lists available on the WWW. There is some redundancy amongst the various sources, but a bit of noise seems acceptable to increase your chances of finding what you're looking for."

## Destination Unknown

<http://www.vnet.net/desthome.html>

The only word that truly fits Destination Unknown is "hip." The art is crisp and modern and the interfaces are visually attractive. What's more, the information is pretty good, too.

## EINet Galaxy

<http://galaxy.einet.net/www/www.html>

Galaxy is another searchable index of Web documents made available by EInet. EInet supports electronic commerce by connecting organizations electronically. They also make Web browsers for the Macintosh and Windows.

## Special Internet Connections

<http://www.uwm.edu/Mirror/inet.services.html>

The SIC index contains pointers to Internet services in general, not just Web documents. They are indexed by category. While not an attempt to construct a complete index, it does have many useful and enjoyable links.

## The World Wide Web Virtual Library

<http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html>

The WWWVL provides a storehouse of information about other Web sites indexed by category.

## Yahoo: A Guide to WWW

<http://akebono.stanford.edu/yahoo/>

Yahoo contains about 10,000 entries in a searchable database. The entries are organized by categories, but also in other ways. You can browse "What's New," "What's Popular," and "What's Cool." You can also ask for a random URL, somewhat like a new word for the day.

## People's Hot Lists

Many people have spent months surfing the Web. Some have published links to their favorite pages for you to follow. Here are some of the best.

### Christian Bartholdsson

<http://www.csd.uu.se/~chris/links.shtml>

### Jeff Van Dyke

<http://ana-www.lcs.mit.edu/people/jvandyke/Hotlist.html>

### Andy Eskilsson

<http://jupiter.pt.hk-r.se/student/pi92ae/fun.html>

### Matthew Gray

<http://www.mit.edu:8001/afs/sipb/project/>

## Web Robots

Every day, dozens of automated systems are running around the Web following hot-links and classifying everything they see. They can then enable you to search these automatically built indices by keywords, subject, header, and so on.

## WebCrawler

<http://www.biotech.washington.edu/WebCrawler/WebQuery.html>

The WebCrawler wanders the Web and collects information about document content and structure. You can search by keyword, and a list of links is retrieved. You can also access the Top 25 List, which shows the most referenced links in the Web. There is also a list of other Web indices.

### **World Wide Web Worm (WWWW)**

<http://www.cs.colorado.edu/home/mcbryan/WWWW.html>

The Worm roams the Web and constructs a searchable index by title and keyword. If you publish your own page, you can tell the Worm about it so that it can find it quickly.

### **WWW Home Pages Broker**

<http://www.town.hall.org/brokers/Home.html>

The WWW Home Pages Broker is an index of over 5,000 WWW pages. The broker uses the Harvest protocol, which attempts to index pages by content. By filling out the form provided, you can search the known pages for specific keywords.

## **Computer Companies and Information**

If you're browsing the Web, you're likely doing it by computer. Therefore, computer companies have jumped aboard to help you get press releases, product information, and even support through the Web.

### **Apple**

<http://www.apple.com/>

Press releases, product information, user groups, and pointers to shareware sites are just a few of the offerings of Apple's server.

### **Microsoft**

<http://www.microsoft.com/>

Microsoft's server is fairly new, but has a wealth of information about the company. You can also browse the employment opportunities.

### **Novell**

<http://www.novell.com/>

Novell has made its technical supports databases available via the Web. They've also developed a very intuitive bookshelf interface to their server. The server presents you with an image of the bookshelf, and then interprets your mouse click to select the proper book.

### **Silicon Graphics**

See listing in the "Graphics" section of this chapter.

## Electronic Commerce

No information superhighway would be complete without someone trying to make money. Although the Internet has its roots in nonprofit organizations, commercial providers and services are sprouting up left and right.

### CommerceNet

<http://www.commerce.net/>

CommerceNet is an ambitious project to bring commercial services to the Internet. Eventually, CommerceNet will provide secure protocols to buy and sell goods using Mosaic and the Web. Currently, you can browse company information, product offerings, and product specifications of dozens of companies.

### List of Commercial Web Services

<http://tns-www.lcs.mit.edu/commerce.html>

This server from MIT maintains a list of commercial services on the Web. From flower shops to cellular phones, it's out there. You can also obtain a list of the newest commercial services.

### Internet Shopping Network

<http://www.internet.net/>

The ISN allows you to buy computer hardware and software using Mosaic. You subscribe to the ISN and supply your billing information over the secure medium of the telephone line. From then on, you can order products by supplying a membership number. Soon, ISN hopes to use CommerceNet technology to do secure transactions over the Internet.

## Government

With the arrival of the first technologically literate White House, the government is beginning to come online. Several local governments and much of the federal government are discovering that the Internet is an amazingly effective way to reach their constituents.

### California Bay Area Governments Online

<http://www.abag.ca.gov/>

Being the home of Silicon Valley, you would assume that the Bay Area Governments would be highly wired. You would be correct. In fact, in the police station in Mountain View, there are public terminals connected to the Web. For an excellent example of what government information on the Web could be, see San Carlos' home page:

[http://www.abag.ca.gov/abag/local\\_gov/city/san\\_carlos/schome.html](http://www.abag.ca.gov/abag/local_gov/city/san_carlos/schome.html)

### FBI

[http://naic.nasa.gov/fbi/FBI\\_homepage.html](http://naic.nasa.gov/fbi/FBI_homepage.html)

That's right, the FBI is online. Currently, they are using the Web to disseminate information about unsolved crimes and offer rewards for information.

### House of Representatives

`gopher://gopher.house.gov/1`

## Senate

`gopher://gopher.senate.gov/1`

With the arrival of the first technologically literate White House, the government is beginning to come online. Several local governments and much of the federal government are discovering that the Internet is an amazingly effective way to reach their constituents.

Among other things, these sites contain the e-mail addresses of many congressmen, and usually at least their telephone numbers.

## Library Of Congress

`gopher://marvel.loc.gov/1`

Sooner or later, the entire contents of the Library Of Congress will be online. In the meantime, LOC is gearing up by putting card catalogs and other information into electronic form.

## NASA

<http://mosaic.larc.nasa.gov/nasaonline/nasaonline.html>

NASA has put an incredible amount of information online. You can browse images of the Jupiter comet collision, space shuttle missions, and even look at the moon landing.

## The White House

`gopher://gopher.tamu.edu/11/.dir/president.dir`

Unfortunately, you can't have a chat with the President yet, but you can browse information about the budget, health care, and other White House affairs. You can even look at the President's daily schedule.

## Graphics/Visual Exhibits

Information is clearly the main benefit of being wired, but let's face it—graphics are the coolest part. When a techno-nonbeliever walks up to your machine, show them one of these sites. They may run out in a rash attempt to get wired.

### Dinosaur Exhibit

<http://www.hcc.hawaii.edu/dinos/dinos.1.html>

Browse this award-winning exhibit before or after watching your favorite dinosaur film. You can even get a narrated tour!

**[Figure A.2. The always scary Tyrannosaurus Rex from the Hawaiian exhibit.](#)**

## The Exploratorium

<http://www.exploratorium.edu/>

The Exploratorium is a hands-on museum in San Francisco.

### **[Figure A.3. At the Exploratorium, you can tour an interactive science museum.](#)**

#### **LEGO**

<http://legowww.itek.norut.no/>

LEGO has long been a supporter of new technologies, and it has ties with many educational institutions. The LEGO site has pictures of home-built constructions, instructions on how to make cool things, and even information on LEGO robots. If you have sound support, check out the LEGO theme song.

#### **Silicon Graphics**

<http://www.sgi.com/>

It's a computer company, but this server has great graphics. Of course, that's no surprise from the *de facto* standard in the making of production quality computer graphics.

#### **The Vatican Exhibit**

<http://www.ncsa.uiuc.edu/SDG/Experimental/vatican.exhibit/Vatican.exhibit.html>

This award-winning site was one of the first exhibits available on the Web. The Vatican exhibit presents many of the Vatican Library's books, maps, and manuscripts. Originally, the exhibit was tied to a real exhibit in the Library of Congress in 1993. The online version will remain working indefinitely.

#### **The Virtual Hospital**

<http://indy.radiology.uiowa.edu/VirtualHospital.html>

Ever wanted to try your hand at medicine without the years of medical school? Here's a safe way to do so. They don't have patient simulations just yet, but they're on the way. In the meantime, you can brush up on your knowledge with the Multimedia Textbooks on a variety of medical topics. In reality, the Virtual Hospital is a clearinghouse of information on state-of-the-art medicine for use by medical professionals.

#### **The WebLouvre**

<http://mistral.enst.fr/>

The WebLouvre won the "Best Use of Multiple Media" Award in the "Best of Web'94" contest, and for good reason. While not connected with Le Louvre, WebLouvre attempts to show multimedia exhibits of the quality that would be seen in the "real Louvre."

### **Online Publications**

The Web is well-suited to online publications. You can read at your own pace, skip around between sections, and look at supporting materials with a simple click. Interesting note: The reason reading the screen seems slower and more cumbersome than paper is that the contrast of paper is much better than current monitor technology.

#### **Books Online**

<http://www.thegroup.net/booktitl.htm>

There are numerous books on the Internet, and this site contains pointers to many of them. You can search by title or author, as well as browse. Titles include the *Bible*, *The Adventures of Sherlock Holmes*, and *The Art of War*. Some are even in HTML format. There are also books that are not published anywhere else, such as Bruce Sterling's *Hacker Crackdown*.

## **Dr. Fun**

<http://neuromancer.lib.uchicago.edu/david/drfun.html>

An eclectic cartoon in the spirit of the Far Side, Dr. Fun is updated every day. It's a great way to start a day's work.

## **Encyclopedia Britannica**

<http://www.eb.com/eb.html>

The staple of school report writing, the Encyclopedia Britannica has come online. They do not yet offer public search of their proprietary data, but there is a fair amount of information available in searchable form.

## **Global News Network (GNN)**

<http://gnn.interpath.net/gnn/GNNhome.html>

One of the most professional pages and services on the Web, GNN is an attempt to stretch the magazine metaphor into the Web and domains. Published by O'Reilly and Associates, GNN allows non-subscribers to browse parts of the magazine, or subscribe by filling out a simple form.

## **Mondo 2000**

`gopher://gopher.well.sf.ca.us/11/Publications/MONDO`

While not strictly "on the Web," *Mondo 2000* maintains a Gopher presence that is accessible through the Web.

## **PowerPC News**

<http://power.globalnews.com/>

From the page itself: "*PowerPC News* is a *free* electronic magazine for the Internet Community. It is published every two weeks via e-mail, and is read by over 25,000 readers worldwide. We don't clutter up your mailbox with the whole issue; instead, readers are just sent the contents page, and can retrieve the stories that they want."

## **Wired**

<http://www.wired.com/>

The magazine that has seemingly swept the Inter-nation, *Wired* has had a Web site since nearly the beginning. The full contents for each issue (sans layout and graphics) is available. Some would say it is easier to read from the Web than the brightly colored pages of the magazine.

## **VIBE**

<http://vibe.www.com/>

*VIBE* is a magazine that deals with urban music and American youth culture. As such, the next logical step is a Web presence. You can choose from a high bandwidth image based interface, or a low bandwidth text interface. You can browse articles, music reviews, and sound snippets from upcoming albums.

## Schools and Universities

Some of the first organizations to get onto the Web were colleges and universities. As such, they've had more time to polish their bits and make truly useful servers.

### Carnegie Mellon University

<http://www.cmu.edu/>

Several departments at CMU have put information on the Web. Some servers are: EnviroWeb, a student run server that deals with environmental issues; Human Computer Interaction, which deals with user interfaces and has a list of software tools for constructing them; and EnglishServer, which provides works in all areas of Arts and Humanities.

### MIT

<http://web.mit.edu/> and <http://www.mit.edu/>

MIT has an official server(web) as well as a student run server(www). You can also access many student's home pages from either of these servers.

### Stanford

<http://www.stanford.edu/>

Stanford has put together one of the most complete University Web servers. From registration to restaurants in the area, it's on the Web.

### Honolulu Community College

<http://www.hcc.hawaii.edu/>

You may chuckle, but HCC was one of the first institutions to dive head long into the World Wide Web. You can access school maps, class information, and even a dinosaur exhibit. The dinosaur exhibit is featured in the "Best of the Web '94" collection.

## Sound/Music

Along with graphics, sound is becoming widespread in the Web. With more home machines able to play CD-quality audio, sound will likely be a much more utilized tool.

### Internet Underground Music Archive (IUMA)

<http://www.iuma.com/index.html>

The IUMA has received incredible press coverage. IUMA provides a medium to distribute "commercial-quality" music over the Web.

### RG

<http://rg.media.mit.edu/RG/RG.html>

RG provides music charts, sound samples, and reviews of current dance music. RG also provides information about the equipment used in the making and performance of dance music.

## **Wired Radio Stations**

<http://sccs.swarthmore.edu/~justin/Docs/radio/stationlist.html>

Many radio stations (mostly college) have Web servers that provide charts, song information, membership information, and so on.

## **Artist Information**

Can't remember the name of that song your favorite artist put out four years ago? Need the lyrics to that latest Madonna song to write your next love letter? No problem. There are many fans out there who have taken the time to gather information about artists and make it available electronically.

### **Tori Amos**

<http://www.mit.edu:8001/people/nocturne/tori.html>

### **Anything Box**

<http://web.cps.msu.edu/~johnson/AnythingBox/>

### **Beastie Boys**

<http://www.nando.net/music/gm/BeastieBoys/>

### **Eric Clapton**

<http://bio-3.bsd.uchicago.edu/home/u2/d-hillman/http/welcome.html>

### **Jane's Addiction**

<http://raptor.swarthmore.edu/jahall/dox/Jane'sAddiction.html>

### **Billy Joel**

<http://www-usacs.rutgers.edu/~rotton/billy-joel.html>

### **Madonna**

<http://www.mit.edu:8001/people/jwb/Madonna.html>

### **Nirvana**

<http://www.luth.se/nirvana/>

### **Wilson Phillips**

<http://www.mit.edu:8001/people/jwb/wp.html>

## **Frank Zappa**

<http://carol.fwi.uva.nl/~heederik/zappa/>

## **Sports**

Sports pages contain information about schedules, players, rules, and many other aspects of the game. It's not surprising that soccer seems to be the most popular sport on the Web—it is, after all, the World Wide Web.

### **Sports on the Web**

<http://www.atm.ch.cam.ac.uk/sports/sports.html>

### **Baseball**

`gopher://xmission.com:79/0rob`

### **Basketball**

<http://tns-www.lcs.mit.edu/cgi-bin/sports/nba/highlights>

### **Fantasy Baseball**

[http://www.cm.cf.ac.uk/User/Gwyn.Price/fantasy\\_baseball.html](http://www.cm.cf.ac.uk/User/Gwyn.Price/fantasy_baseball.html)

### **Football**

<http://www.mit.edu:8001/services/sis/NFL/NFL.html>

### **Soccer**

<http://www.atm.ch.cam.ac.uk/sports/webs.html>

### **Surfing (Water-Based)**

<http://sailfish.peregrine.com/surf/surf.html>

## **Stocks/Finance**

Who says business and the Internet don't mix?

### **Security APL and Data Transmission Network**

<http://www.secapl.com/cgi-bin/qs>

### **EDGAR Dissemination Project**

<http://www.secapl.com/cgi-bin/edgarlink>

From the page itself: "The Internet EDGAR Dissemination project will allow you to receive any 1994 filings to the

Securities and Exchange Commission that are available to the public. Nonelectronic filings, filings that are not available to the public, and any data prior to 1994 will not be available here."

## **quote.com**

<http://www.quote.com/>

Quote.com is a commercial service that offers financial market data. With the monthly subscription fee, currently \$9.95, you can get a limited number of stock quotes, get news about stocks, and manage your portfolio (no buying or selling). Quote.com provides a "stock alarm" service that notifies you by e-mail if and when a stock goes above or below a certain price.

## **MIT AI Lab Stock Project**

<http://www.ai.mit.edu/stocks.html>

The MIT AI Lab is experimenting with online historical and timely information about stocks. This site can generate history graphs and current price data for over 300 stocks, mostly computer and media stocks.

[Figure A.4. A graph of Paramount stock from the MIT server.](#)

## **NETWorth**

<http://networth.galt.com/>

"The Internet Resource for Individual Investors." NETWorth stores mutual fund information, performance reports, and prospectuses.

## **Miscellaneous**

These sites didn't seem to fit in any of the other categories, but were worth mentioning nonetheless. When you're in the mood for general surfing, check them out.

### **Boston Computer Society Internet SIG**

<http://isig.mit.edu/>

The Boston Computer Society provides a Special Interest Group (SIG) for those interested in finding out about the Internet and how to use it. Meeting information and minutes can be found on this page.

### **Cardiff's Movie Database**

<http://www.cm.cf.ac.uk/Movies/>

Here you can search a database of over 31,000 movies and find out who directed, starred, or designed the costumes. You can also rate movies and view others' ratings. Also at this site is a "this day in history" service that tells you which celebrities were born or died on a given day.

### **Fax Gateway**

<http://town.hall.org/fax/faxsend-short.html>

Want to send a fax free? The Internet FAX Server is an attempt to link the Facsimile and electronic mail worlds

together "seamlessly." By using a collection of local fax servers, the Internet Fax Server routes your fax to the appropriate local fax machine and avoids long distance charges. If your e-mail system can support MIME, you can send images and other complex data types.

## htMUD

<http://www.elf.com/~phi/>

htMUD is an attempt to link the world of Multi-User Dungeons (muds) to the Web. The concept is that you interact using Telnet, and view your surroundings in full multimedia glory in Mosaic. While it will take some configuration hacking and program installation, htMUD is an excellent attempt at linking two of the most popular services on the Internet.

## Usenet FAQs

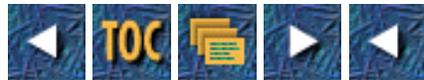
<http://www.cis.ohio-state.edu/hypertext/faq/usenet/top.html>

There are thousands of Usenet news groups. Most have a Frequently Asked Questions (FAQ) document. You can browse and search them through this page. Most are in standard text form, but some have been "hyperized."

## Why the Web Sucks

<http://wiretap.spies.com/ceej/Writing/rant.web.html>

Just as you're all excited about being wired and getting on the Web, someone writes about how bad it is. There are many good points in this very interesting view on the problems of the Web. Just because it's easy to put up something for the whole world to see, does that really mean you have something interesting to say?





- [Appendix B](#)
  - [Signature Files](#)
    - [--by Max Metral](#)
    - [What Are Signature Files?](#)
    - [How to Write Your Own Signature File](#)
      - [The Basics](#)
      - [Enhancements: Humor](#)
      - [Introduction to Graphics](#)
      - [Full-Blown ASCII-Art](#)
      - [The alt.ascii-art and rec.ascii.art Newsgroups](#)
      - [World Wide Web Pages](#)
      - [Gifscii](#)
      - [cmellow@world.std.com](#)
      - [Ed Hammond \(spock@media.mit.edu\)](#)
      - [Nasser Hasian \(nhasian@eis.calstate.edu\)](#)
      - [Josh \(hades@u.washington.edu\)](#)
      - [Christopher King](#)
      - [Richard Lachman](#)
      - [Thomas Langberg](#)
      - [Shawn Lee](#)
      - [Tai Nguyen \(thn1@cornell.edu\)](#)
      - [vampyr@bu.edu: Design for "Jafar"](#)
      - [vandal@netcom.com](#)
      - [Wayne](#)
      - [vampyr@bu.edu for hry@cs.pdx.edu](#)
      - [boba@wwa.com, a.k.a. Scarecrow](#)

## Appendix B

### Signature Files

[--by Max Metral](#)

## What Are Signature Files?

*Signature files* are pieces of text appended to a user's communication, usually e-mail or news postings. Originally, they were intended as a simplification to relieve users of typing the same thing over and over again. Of late, they have become artistic expressions, advertisements, and electronic business cards. Not everyone likes signature files, though. Take Steve Strassmann, coauthor of *The Unix-Haters Handbook* (IDG Books: Indianapolis, 1994), who writes the following:

".sigs were invented because of sendmail and other cretinous unix mail 'tools' that violate all sorts of Internet standards. People had to put their real addresses at the bottom of messages, because the headers were so likely to be worthless. If those systems actually worked as they should (and once did), nobody would ever have needed .sigs.

A UNIX signature isn't a return address; it's the ASCII equivalent of a black velvet clown painting. It's a rectangle of carets surrounding a quote from a literary giant of weenedom like Heinlein or Dr. Who.

.sigs are the perfect source of thoughts for people on junk information diets; myopic bottom-feeders who eat whatever information McNuggets are squirted into their face by passersby. Ultimately, .sigs are for people with snack food where their brains should be."

If that hasn't discouraged you from designing your own signature, read on.

## How to Write Your Own Signature File

To start, please insert tongue in cheek. Also, consider the following issues when designing:

- *Number of lines.* Some mailers require your .sig to be less that a certain number of lines. Check the documentation for your mailer.
- *Angry Sysadmins.* Remember that the longer your sig, the bigger your message. Sysadmins tend to get upset when their disks overflow. These days, however, one full screen .gif in a message would take the same space as about 100 signatures.
- *Angry users.* While the Sysadmins have gigabyte drives to play with, the people you're sending mail to may not. The space problem has merit for individual users.

### The Basics

The functional point of a signature file is to give the reader contact information. Think of what you would put on a business card, and those things should go in the signature. For example:

====<Max>====

General Magic, Inc.

Phone: 415.XXX.XXXX

2465 Latham St.

Fax: 415.XXX.XXXX

Mountain View, CA 94040

Email: memetral@mit.edu

Note the double-column structure, a very popular format. These days, a lot of people are adding their World Wide Web home pages to their signatures as a "Badge of Honor." For example, Tyler Jones:

/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\

Tyler Jones, tjones@willamette.edu

KILL

/-----\

<a href="http://www.willamette.edu/~tjones/">TVJ</a> YOUR < Amethyst >

If you want to impress me, accomplish something. CAR \-----/

### Enhancements: Humor

After you've mastered the informational aspects of the signature, you're ready to move on to the next level: diversionary information. Many people use humorous quotes to add some life to their messages. There are advantages and disadvantages to this approach. While your joke may make someone laugh, will it work for 100 mail messages? Select a joke or quote that expresses a part of you. For example, cynthia@ncd.com, a clear Simpson's fan, has chosen:

~~~~~

Cynthia Virtue cynthia@ncd.com  
Network Computing Devices Tel: xxx-xxx-xxxx  
Mountain View, California, USA Fax: xxx-xxx-xxxx

I am Homer of Borg. Prepare to be --- Oooh! Donuts!

~~~~~

Note the use of the ~ character to frame the signature. We'll see more of that later.

### Introduction to Graphics

Humor, information, blah, blah, blah. ... You want your signatures to look like all those cool ones in the techie magazines. The only way to do that is to whip out the old character-based graphics. Start simple, like will@panix.com:

\_\_\_\_\_  
/ / / / / / \_ \_/  
(\_/\_/ / ( ( \_ \_ \_/

Then add your knowledge from the previous lesson:

||\ /|| || |\\ \ \ \ \ \ \ || ||\ || \*\* Cyber Space Cowboy \*\*  
|| \ / || || || || || || || || \ || \*\* President, Alt.Flame \*\*  
|| || || || || || || || || || \ || \*\* The MoFo Integrator \*\*  
|| || || || || \ || || || || \ || Just the s\*\*t....

or:

~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*  
WinJunkie aka Brad Spry winjunki@vnet.net webmaster@vnet.net  
~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*!^`!~\*-,.\_.,-~\*

Dave Carhart (stilwatr@tiac.net) has incorporated information, humor, and simple but effective graphics. Way to go, Dave!

\|/

(@ @)

-----oOO--( \_)--OOo-----

^ ^ ^            | | |

"Always code as if the guy who ends up maintaining, or testing your code will be a violent psychopath who knows where you live."

-----

Dave Carhart

Voice: 508-XXX-XXXX

Stillwater Consultants Inc.

Internet: stilwatr@tiac.net

-----

### Full-Blown ASCII-Art

Now that you've mastered the basics, time to let loose. Think of images, designs, slogans, and other formats that appeal to you. To help in your design, you can turn to the following sources.

### The *alt.ascii-art* and *rec.ascii.art* Newsgroups

Loads of interesting information about ASCII-art. You will be amazed at what can be done with ASCII. Heck, we can just skip this multimedia business. You can even produce those highly annoying 3-D stereograms with ASCII!!!

### World Wide Web Pages

<http://gagme.wwa.com/~boba> contains numerous resources and informational documents from the ASCII-art guru, Scarecrow. You will feel very strange viewing all ASCII pages in a medium (the Web) that is famous for multimedia.

### Gifscii

Gifscii is a program that can convert GIF images to ASCII. While Steve Strassman would be sick if he knew you were putting screensful of characters at the end of your messages, Gifscii may come in handy for smaller images. You can find Gifscii at ftp.wwa.com in pub/Scarecrow/Gifscii. The file you retrieve will depend on the desired platform:

|            |                  |
|------------|------------------|
| MSDOS      | Gifscii.exe      |
| UNIX (Sun) | Gifscii.ux       |
| Macintosh  | MacGifscii.hqx   |
| Amiga      | Gifscii.lha      |
| Unix/Alpha | ALPHAGifscii.exe |
| VAX        | VAXGifscii.exe   |
| c-source   | Gifscii2.2src    |

With no further ado, we get to some of the best sigs on the Internet:

**cmellow@world.std.com**

.-'~~~~.

vibrate . 'o o000o`. . . . .sQuiShy Designs. . . . .

your :~~~~-.oOo o`.

skull `.\ ~-. o00o.6 1 7 4 9 9 1 9 2 5

`. ; / ~. OO: info-flo

. ' ;-- `o.'

this'll stop , ' ; ~~~-'~ ftp: ftp.std.com /pub/cmellow

them shakes. ; ; ftp: amt.mit.edu /pub/jshiple

\_\_\_\\;\_\\//\_\_\_\_\_WWW: http://bazaar.com/dj/Cmellow/cmellow.html

fingerforPGPpubkey

**Ed Hammond (spock@media.mit.edu)**

-

-|-|-

^/ . ..\^

\_\_\_[=====]\_\_\_

\_\_\_-==++"" . / . . . \ . ""++==-\_\_\_

\_\_-+"" \_\_\ . . . | . . . | . . . /\_\_ ""+-\_\_

/\\_\_+"" `-----=====\\_ <O> \_/=====-----' ""-+\_\_/\

\_/\_/ ""="" \\_\\_

/\_/ \\_\

// \\  
/" ) ("\  
\o\ Live Long and Prosper, /o/  
\\_ ) (\_/

Spock



# Christopher King

Chris

\_\_^\_\_

\_\_^\_\_

( \_\_\_ )-----( \_\_\_ )

| / | Christopher King NCSU Computing Center | \ |

| / | 208 Hillsborough Building (919)XXX-XXXX | \ |

| / | WWW URL>http://www2.ncsu.edu/unity/users/c/chking/ | \ |

|\_\_| "Calm down. It's only ones and zeros." -- Sam Kass |\_\_|

(\_\_\_\_)-----E-mail:chris\_king@ncsu.edu-----(\_\_\_\_)

# Richard Lachman

—

/ \ Sit, Ubu, Sit.

/ ..|\ Good Dog.

(\_\ |\_)

/ \@' \*WOOF\*

/ \

\_ / ` |

\\/ \ | \_\

\ /\_ || \\_\

\\_\_\_\_) |\_) \\_)

— —

' ) ) / /

-- /--' o \_ . /\_ \_ . \_ \_ /

/ \<\_( \_/ /\_( \_/ | \_/ ( \_(\_/ \_

richlach@athena.mit.edu

richlach@media.mit.edu

## Thomas Langberg

```

-----
--      -- Thomas Langberg      -- Email: tll@swl.msd.ray.com      --
--  ///\  -- Raytheon Company    -- Phone: 508-XXX-XXXX (direct)  --
--  c-oo  -- 50 Applehill Dr.   -- Disclaimer: My opinions are my  --
--  \  ^  -- Tewksbury, MA     -- own, any similarity to another  --
--  \_U  --           01876-0901 -- is purely coincidental      --
-----

```

```

--      Q: Why did the chicken cross the Moebius strip?      --
--      A: To get to the other ... er, um ...                --
-----

```

## Shawn Lee

```

      _  _-==_-, -.
/--`' \_@-@.--<
`--'\ \  <__/.
      \ \  " /
      >=\_/'<
_____ / = | \_ | /
_'  '\  _/=== \__ /
= -_ __/=====

```

|  |                                  |  |                  |  |
|--|----------------------------------|--|------------------|--|
|  | Shawn P. Lee (612)XXX-XXXX       |  | Winnie The Pooh, |  |
|  | ChriShawn@tiger.stcloud.msus.edu |  | and              |  |
|  | St. Cloud State University, MSUS |  | TIGGER too!!     |  |

```

=====

```

## Tai Nguyen (*thn1@cornell.edu*)

\_\_\_\_\_

```
(:)===/_/_/\=(:)(:)======:)======:)=
|\|  _\  \  \|\|=|  Tai Nguyen      |\|  official sponsor of |=|
|=|  /_/_/\_  \=|\|\|  resident wise-ass      |=|  SoNiC ToNiC(tm)  |\|
|\|  /  /\  __/\|(:)======:)======:)=
=	/  /  /\  \	=		=	"Nature has been made possible by a grant from	=
\/_/  \_\/	\|\|  the Corporation for Public Broadcasting."	\|				
=\_\/\  \	=		=	- PBS	=	
(:)=\_\/=:(:)(:)======:)======:)=
```

**vampyr@bu.edu. Design for "Jafar"**

```
.od8888bo.      88  .d888888  8888888888  .d888888  888888ba
.d%::::88::888b.      88 d8'      88 88      d8'      88 88      `8b
.d888:::::8:888%.      88 88aaaaa88a 88aaaa 88aaaaa88a 88aaaa8P'
88888:::::88888::%.      88 88      88 88      88      88 88      `8b.
d888888:::88;888888::b 88, ,d88 88      88 88      88      88 88      88
8888888888:888888888888 ..`Y888P'..88.....88..88.....88.....88..88.....88.....
Y88888888:::::8888888P .....
`88888888:::::888888:%'.....
`88888888::888888%'..ii http://192.107.38.20:1999/ iiii
!. `88888888:888888' .!!!!!!! "to be a rock and not to roll" - Led Zeppelin !!!!
|||.. `Y88%8P" ..||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

**vandal@netcom.com**

Y o u r s I n C h a o s

```
— — — —
/\ \/\ \      /\ \      /\_ \
\ \ \ \ \      — — — —  \_ \ \      — \/\ \ \
\ \ \ \ \ /'_ _ ` \ /'_ _ ` \ /'_ _ ` \ /'_ _ ` \ \ \ \
```

\ \ \\_ / \ / \L\.\\_ / \ / \ / \L\ \ / \L\.\\_ \\_ \\_ \\_
\ `\\_ / \\_ /.\\_ \\_ \\_ \\_ \\_ ,\\_ \\_ /.\\_ \\_ \\_ \\_ \\_
`\\_ / \\_ / \\_ / \\_ / \\_ / \\_ / \\_ ,\\_ \\_ / \\_ / \\_ / \\_ / \\_ /

vandal@netcom.com

[DnA/LaW]

Kingdom Of The Blind BBS 714-842-7170

Finger Me For PGP Key

Wayne

wayne's world

\\\\\\\\\\\\\\\\\\\\ \\\\XXXXXXXXXXXX\ \\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\ \\\\XXXXXXXXXXXX\\ \\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\ \\\\XXXXXXXXXXXX\\ \\\\\ \\\\\\\\\\\\\\\\\\\

\*\*\*\*\*

\\\\\\\\ \\\\ ncam@helix.watstar.uwaterloo.ca \\\\ \\\\ \\\\
\\\\\\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\
W A Y N E | W O R L D
\\\\\\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\ \\\\
\\\\\\\\ \\\\ \\\\ rEmiNisCEnCeiSakINdoFabStruSeNesS \\\\ \\\\

\*\*\*\*\*

sig credit: \\\\XXXXXXXXXXXX \\\\XXXXXXXXXXXX
<kerofaye> \\\\XXXXXXXXXXXX \\\\XXXXXXXXXXXX
1994 (C) XXXXXXXXXXXXXXX XXXXXXXXXXXXXXX

vampyr@bu.edu for hry@cs.pdx.edu

88 88 \_
88 " " oo\
88 ( ) \ \_
88 ,dPPYbo, 88 8b,dPPYbo, ,od888bo, 8b d8 ,o\ \bo, .' `
88P' "8a 88 88P' "Y8 d8P' `Y8b `8b d8' d8P'\ `Y8b \







>>> boba@wwa.com <<<

```

-
/ )
/ /
_( ( _  _  _____
(((\ \> /_> / \
(\\ \_ / / |      This guy here.  He's a troublemaker.  |
\      / |      Always sticking his nose into how things |
\  _/ |      like government, media, ads, PR, etc works.  |
/  / \_____/
/  / | /
/  / | /
```

Remember, don't be a junk food information provider. The sigs featured here, for the most part, have clear artistic value. I think even the most avid UNIX haters could agree with that!





- [C](#)
  - [Standards and Where To Find Them](#)
    - [--by Max Metral](#)
    - [The OSI Reference Model](#)
    - [TCP: Transmission Control Protocol \(Transport Layer\)](#)
    - [SNMP: Simple Network Management Protocol \(Application Layer\)](#)
    - [FTP: File Transfer Protocol \(Application Layer\)](#)
    - [Telnet \(Application Layer\)](#)
    - [SMTP: Simple Mail Transfer Protocol \(Application Layer\)](#)
    - [MIME: Multipurpose Internet Mail Extensions \(Application and Presentation Layers\)](#)
    - [HTTP: HyperText Transfer Protocol \(Application Layer\)](#)
    - [HTML: HyperText Markup Language N/A](#)
    - [POP: Post Office Protocol \(Application Layer\)](#)
    - [IMAP: Interactive Mail Access Protocol \(Application Layer\)](#)

## C

# Standards and Where To Find Them

## [--by Max Metral](#)

"The great thing about standards is that there are so many of them." —unknown

Indeed, there are thousands of standards, and usually at least two for each problem domain. The Internet is no exception. Here's a brief description of the most popular standards. Each standard will be accompanied by its place in the OSI Reference Model as well as any documents that define the standard. Most standards are defined by Request For Comments (RFC) documents. To get an RFC, you can point your Web browser at:

[http://info.cern.ch/hypertext/DataSources/Archives/RFC\\_sites.html](http://info.cern.ch/hypertext/DataSources/Archives/RFC_sites.html)

or FTP to venera.isi.edu and look in the directory in-notes.

## The OSI Reference Model

The Open Systems Interconnection Reference model was designed by the International Organization for Standardization and International Electrotechnical Committee. Its purpose is to provide a model to discuss the different tasks involved in computer networks and communication. The model defines seven layers that account for all the pieces of OSI:

| <i>Layer</i> | <i>Task(s)</i>                                                                                                                                                                                                      |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application  | Manages communications between applications, and between applications and lower levels.                                                                                                                             |
| Presentation | Responsible for mapping the format of communicated data onto that used by the application.                                                                                                                          |
| Session      | Controls data exchange.                                                                                                                                                                                             |
| Transport    | Responsible for communicating data reliably. For example, this layer will handle retries in case of network failures.                                                                                               |
| Network      | Responsible for abstracting the Data Link and Physical Layers in a common format for higher layers. In other words, it provides a common interface regardless of communications media or other lower level factors. |
| Data Link    | Responsible for data communications in a single network. For example, sending bits over one office subnet. When multiple subnets are involved, the Network Layer takes over.                                        |
| Physical     | Responsible for pushing bits over the communications media (fiber, copper, and so on). As the name implies, the physical layer is the actual cable, and not a protocol.                                             |

Most standards listed below have a place in the OSI Reference Model. Noting their place will help you understand exactly where the responsibility of the standard begins and ends.

### TCP: Transmission Control Protocol (Transport Layer)

TCP enables a connection-oriented transport mechanism using the Internet suite of protocols. TCP goes through three stages for each communication session: connection, data transmission, and disconnection. During data transmission, TCP handles error recovery and out-of-order problems. For example, if a packet of information is not acknowledged by the recipient, TCP will attempt to resend that information. It will also reorder packets on the receiver side so it appears that nothing out of the ordinary has occurred. *See RFC 761.*

### SNMP: Simple Network Management Protocol (Application Layer)

SNMP arose from the need of system administrators to debug network problems remotely. When trying to figure out which of 100 network routers has gone bad, it is much more convenient to do it from a desk than to run to all the routers. SNMP provides tools to monitor and change network configurations remotely. Unfortunately, the first version of SNMP did not provide authentication, so in theory anyone could change the network configuration. As a result, most vendors disabled the modification features of

SNMP. There is a successor in the works, SNMPv2, which provides authentication. *See RFC 1157.*

## **FTP: File Transfer Protocol (Application Layer)**

As the name suggests, File Transfer Protocol allows for the exchange of files over the Internet Protocols. FTP aims to shield users from differences in file systems between hosts. For example, when FTPing from a DOS machine, you are able to use the same commands and filename formats as you would from a UNIX machine. *See RFC 959.*

## **Telnet (Application Layer)**

Telnet provides a session-based protocol for communication between hosts. Behind the buzz words, telnet is primarily used for remote login. However, most telnet programs are capable of interfacing to other protocols, such as SMTP and POP (see the RFCs of those protocols for more information). *See RFC 854.*

## **SMTP: Simple Mail Transfer Protocol (Application Layer)**

Almost all mail over the Internet is carried using SMTP at one point or another. In an all-SMTP environment, the path from sender to recipient would be that shown in Figure C.1.

### [Figure C.1. The data path for a typical SMTP message.](#)

SMTP provides protocols and directives for forwarding mail from one host to another on the way to the ultimate destination. Once at the ultimate destination, it is up to the particular implementation of SMTP to make the data available to the message recipient. This can be accomplished in several ways (see POP and IMAP). The format of the content of the mail is described in RFC 822. This includes header and body formatting.

It's interesting to note that the commonly used SMTP is a 7-bit protocol. In other words, all characters sent using SMTP must have an ASCII code less than 127. For text, this is usually no problem. When trying to send graphics, programs, or other binary files, it's a big problem. This is why encoding methods such as Uuencode and BinHex are commonly used. These methods pack 8-bit data into a 7-bit data stream and extract it again on the other side. There is a new specification for 8-bit SMTP, but because most client applications want to be compatible with all mail hosts, few have moved to 8-bit SMTP. *See RFC 821 and RFC 822.*

## **MIME: Multipurpose Internet Mail Extensions (Application and Presentation Layers)**

Often mistaken for Multimedia Internet Mail Extensions, MIME describes formats for many different content types. MIME is on the same level as RFC 822, which describes the content of messages exchanged via SMTP. An interesting feature of MIME is that it accounts for different representations of the same data. For example, an image can be represented in a binary form and a textual form, so that readers who can't handle the image can still display a text alternate. *See RFC 1341.*

## HTTP: HyperText Transfer Protocol (Application Layer)

HyperText Transfer Protocol is associated with the World Wide Web. Mosaic and other browsers use HTTP to retrieve hypertext documents. HTTP is a stateless protocol; an HTTP client connects to the server, gets its data, and leaves. For this reason, truly interactive services on the Web must be hacks that get around this limitation. HTTP can exchange many types of data, including MIME types. *See RFC draft:*

<ftp://ds.internic.net/internet-drafts/draft-ietf-iiir-http-00.txt>

## HTML: HyperText Markup Language N/A

HTML describes a language to represent hypertext and multimedia documents. HTML is not in the OSI Reference Model because it is purely a data format, not related explicitly to communication or networking. HTML is a subset of Standard Generalized Markup Language (SGML). The hypertext documents in the World Wide Web are written in HTML. HTML is composed of text and a set of tags, and can be written with text editors or with a variety of WYSIWYG editors. The following is an example of a bold string in HTML:

This is a `<b>good</b>` book.

*See RFC draft:*

<ftp://ds.internic.net/internet-drafts/draft-ietf-iiir-html-01.txt>

## POP: Post Office Protocol (Application Layer)

Post Office Protocol gained popularity as the client-server model became a standard. In the past, users would read their mail on a UNIX workstation with shared file systems with the mail server. As personal computers began to enter Internet sites, there needed to be a non-file system based solution. There are essentially two reasons for this. First, PCs are not always on. SMTP works best when reliable machines are used as mail hubs. If your machine is off for three days, you don't want your mail to bounce back to its originator. Secondly, many PCs don't have implementations of the heavyweight file sharing protocols used in many UNIX systems. *See RFC 937.*

POP provides a solution. To use the Post Office metaphor, the mail server is the Post Office, and your mail client is the customer. When mail arrives, it goes into local storage in the Post Office. When you want to check your mail, you connect with the Post Office, grab your mail, and leave. After getting your mail, the Post Office can delete its copy of the data.

## IMAP: Interactive Mail Access Protocol (Application Layer)

IMAP is conceptually a post-POP protocol for managing a mailbox from multiple client machines. Travelers will know that it's a real pain to deal with the same mail twice: once while on the road, and once while in the office. IMAP keeps mail on the server machine and enables clients to browse and delete messages remotely. You may also bring down local copies of mail to be read while disconnected. It handles synchronization and the other problems that arise from multilocation access.

IMAP has been around for a while, but hasn't truly caught on yet. The main problem is that IMAP almost goes too far in the client-server model by pushing too much responsibility onto the server. The problem it attempts to solve still exists for many users, but the current trend is to provide access to client software on the road. For example, many Macintosh users can dial into their own machine and handle their mail remotely using AppleTalk Remote Access and a mail client such as Eudora. *See RFC 1203.*

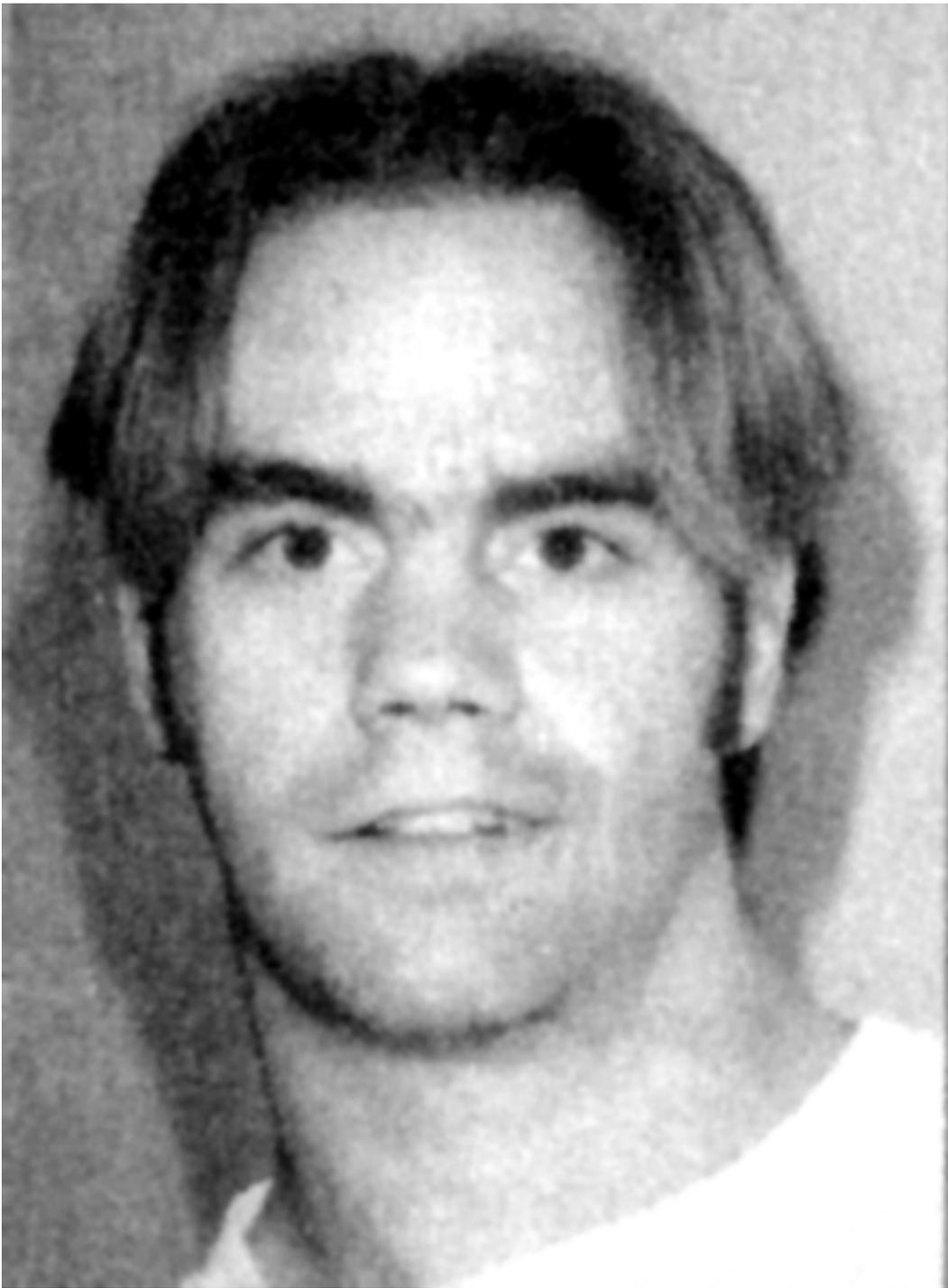


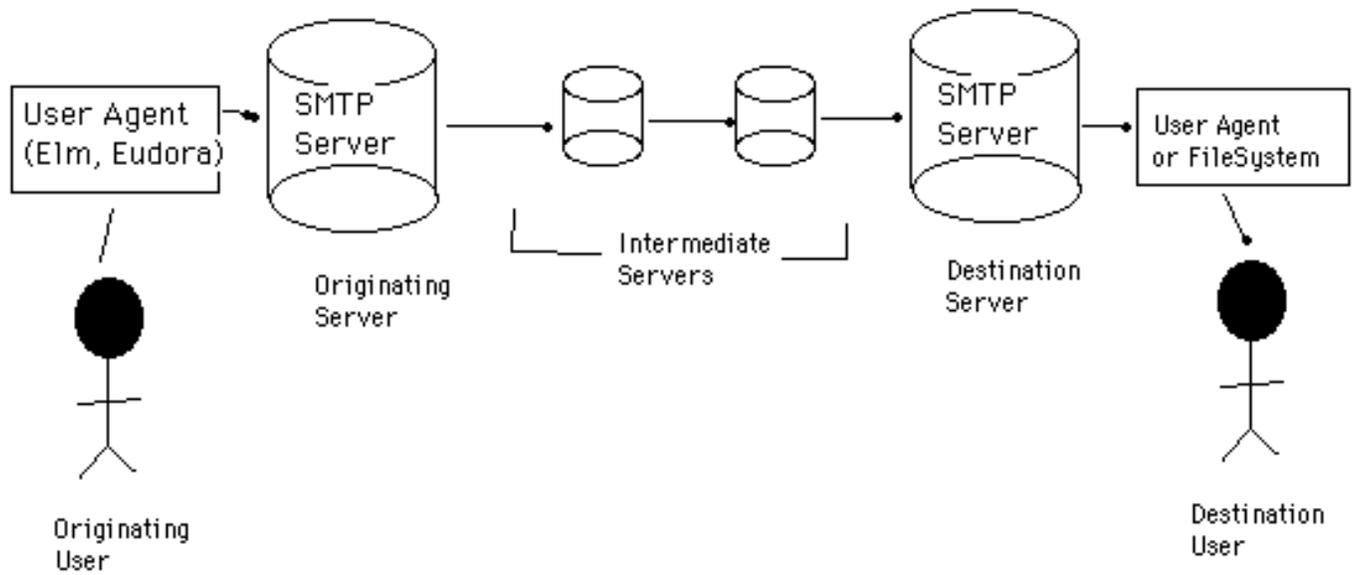


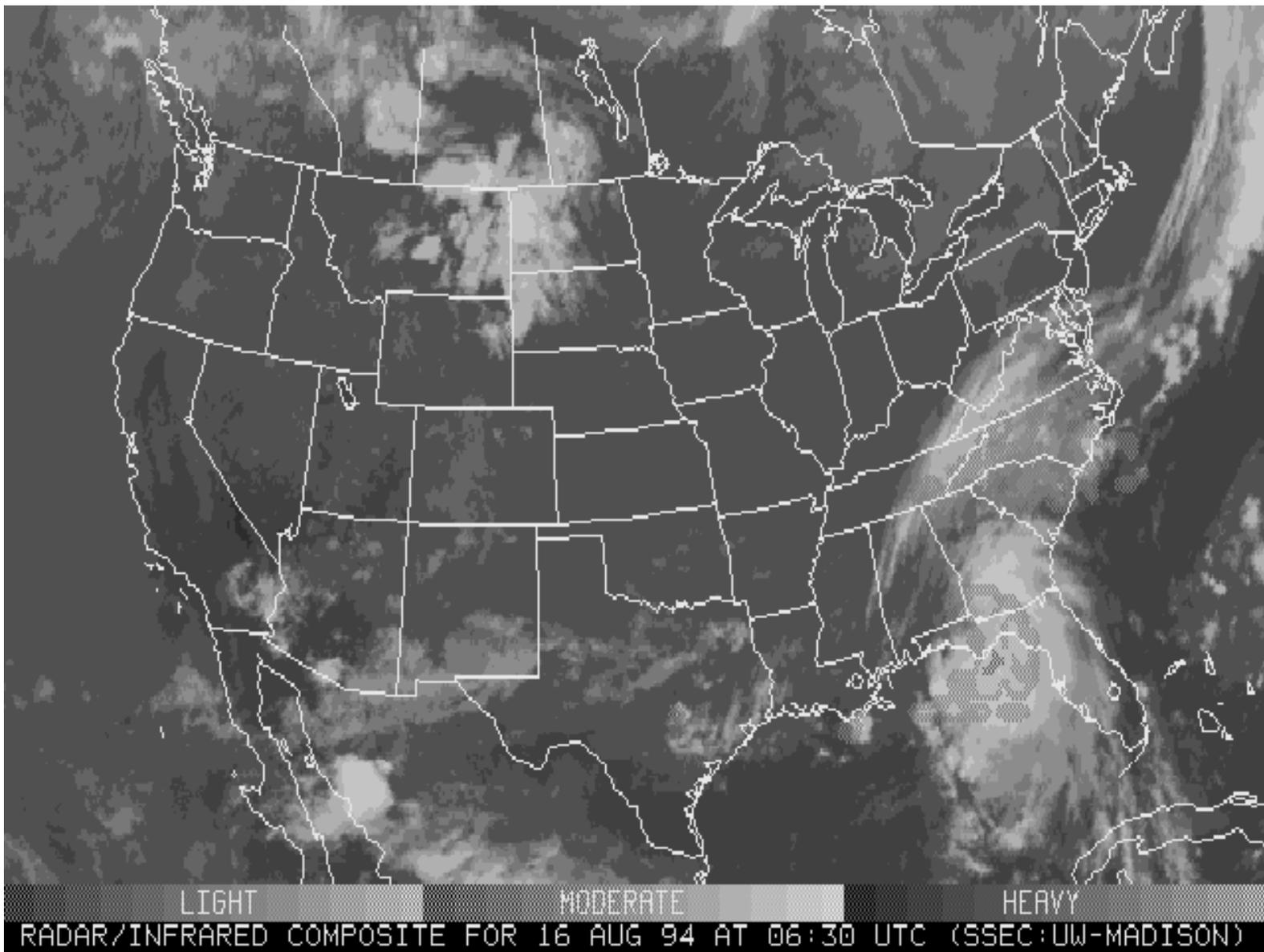
|                      |                                                                 |                |
|----------------------|-----------------------------------------------------------------|----------------|
| <i>0-672-30519-4</i> | <i>Teach Yourself the Internet: Around the World in 21 Days</i> | <i>\$25.00</i> |
| <i>0-672-30520-8</i> | <i>Your Internet Consultant: The FAQs of Online Life</i>        | <i>\$25.00</i> |
| <i>0-672-30466-X</i> | <i>The Internet Unleashed (Book/Disk)</i>                       | <i>\$44.95</i> |
| <i>0-672-30485-6</i> | <i>Navigating the Internet, Deluxe Edition (Book/Disk)</i>      | <i>\$29.95</i> |
| <i>0-672-30507-0</i> | <i>Tricks of the Game Programming Gurus (Book/CD-ROM)</i>       | <i>\$45.00</i> |
| <i>0-672-30402-3</i> | <i>UNIX Unleashed (Book/CD-ROM)</i>                             | <i>\$49.99</i> |
| <i>0-672-30595-X</i> | <i>Education on the Internet</i>                                | <i>\$25.00</i> |
| <i>0-672-30549-X</i> | <i>Teach Yourself TCP/IP in 14 Days</i>                         | <i>\$29.99</i> |
| <i>0-672-30571-2</i> | <i>UNIX Security for the Organization (Book/Disk)</i>           | <i>\$35.00</i> |
| <i>0-672-30617-4</i> | <i>The World Wide Web Unleashed</i>                             | <i>\$39.99</i> |
| <i>0-672-30583-6</i> | <i>Teach Yourself UNIX Shell Programming in 14 Days</i>         | <i>\$29.99</i> |

**Book ISBN 0-672-30599-2**









Document Title: Honolulu Community College Dinosaur Exhibit

Document URL: http://www.hcc.hawaii.edu/dinos/dinos.2.html

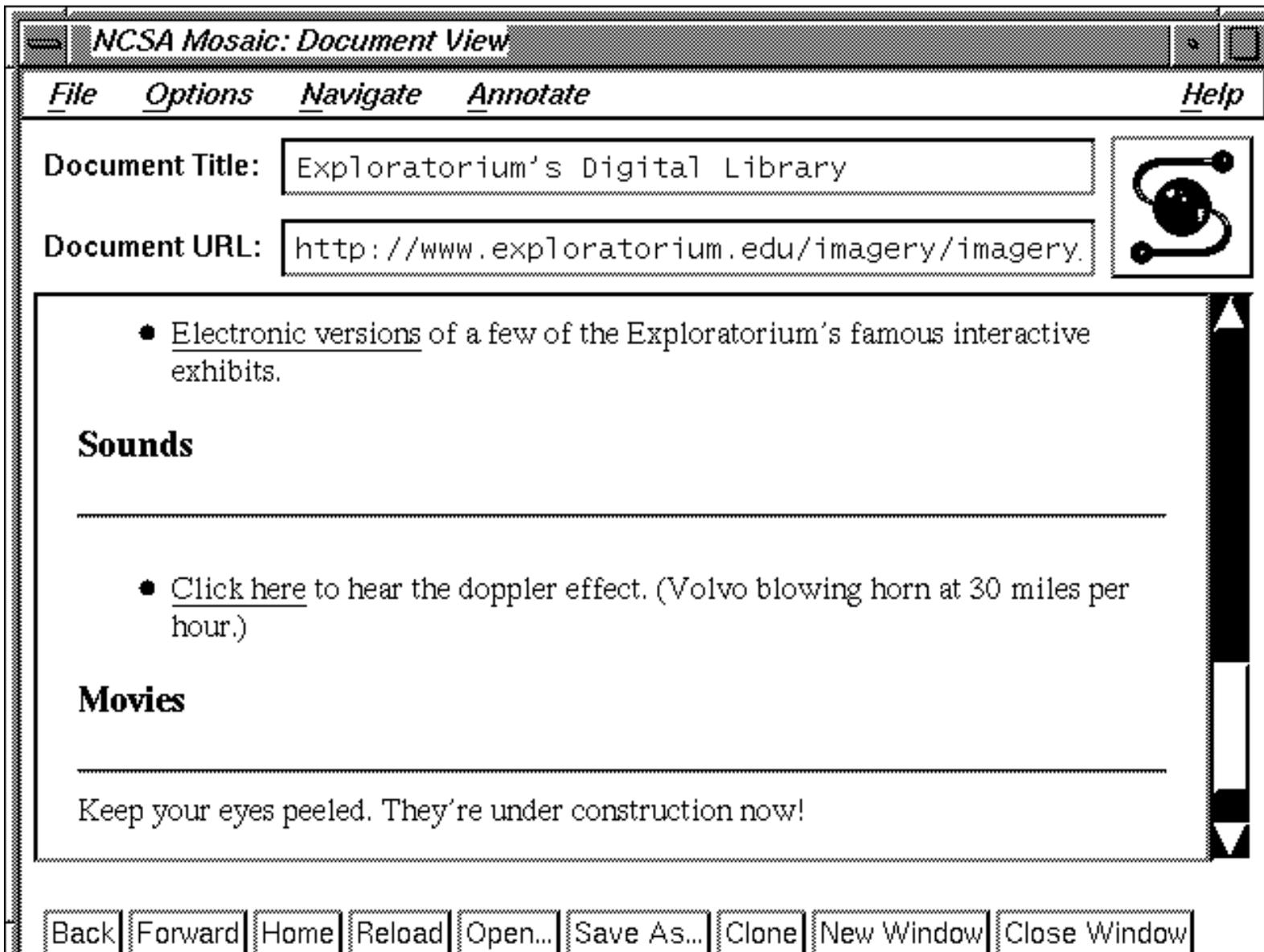


## Tyrannosaurus Rex Head



**Tyrannosaurus rex** lived from about

Back Forward Home Reload Open... Save As... Clone New Window Close Window



File Options Navigate Annotate

Help

Document Title: Graph of Paramount Communications (PCI)

Document URL: http://www.ai.mit.edu/stocks/graph?PCI



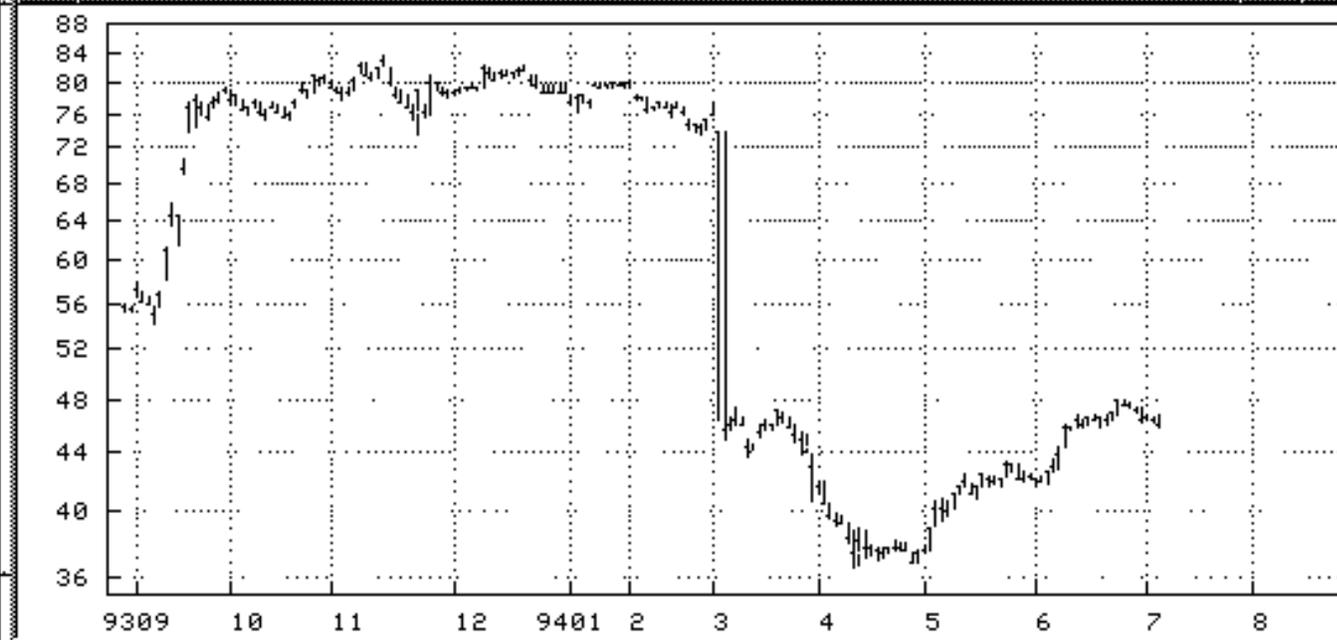
## Graph of Paramount Communications (PCI)

xv 3.00: tmp/ayaa00546.gif

as o

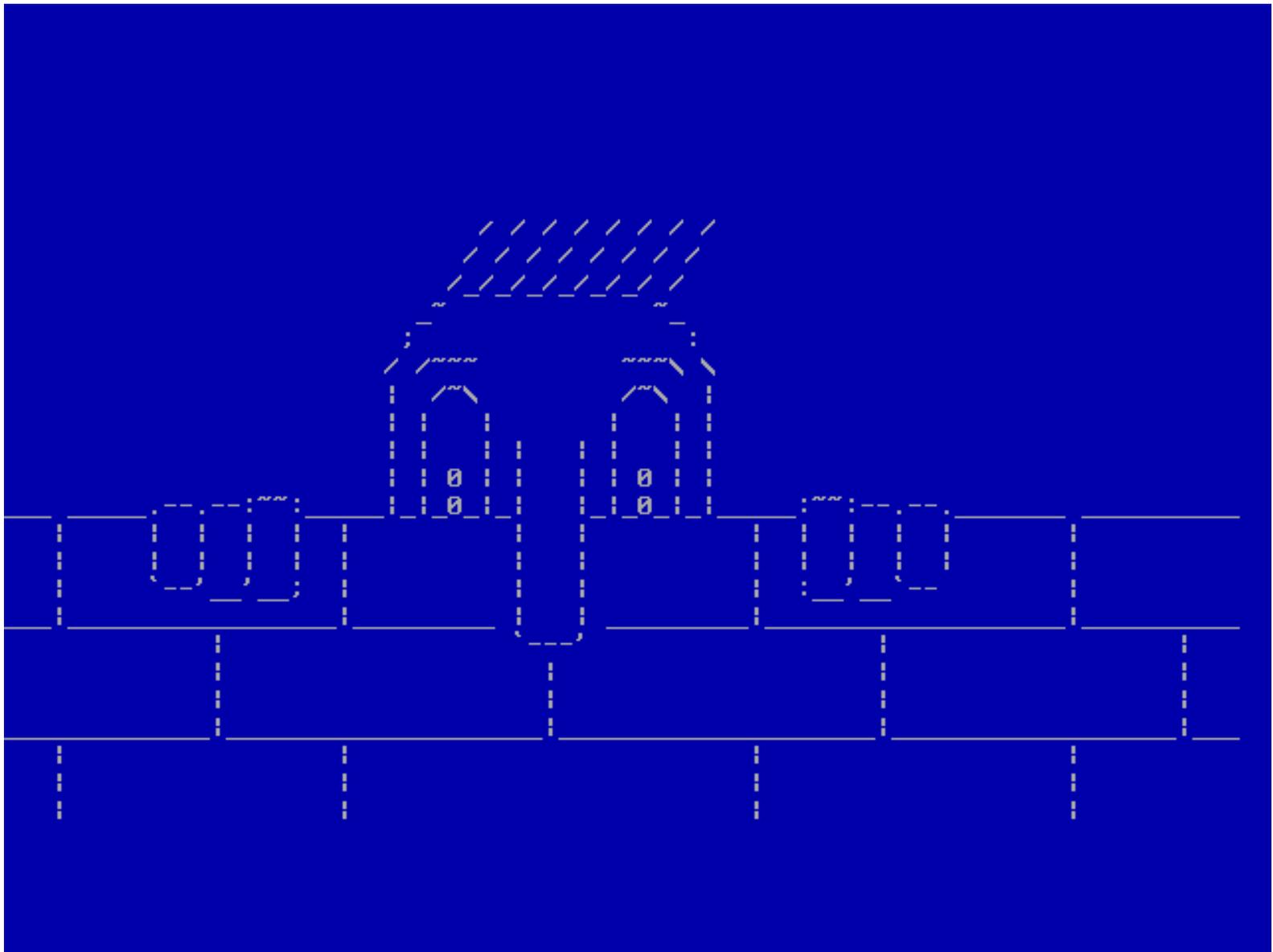
The  
Intel

88  
84  
80  
76  
72  
68  
64  
60  
56  
52

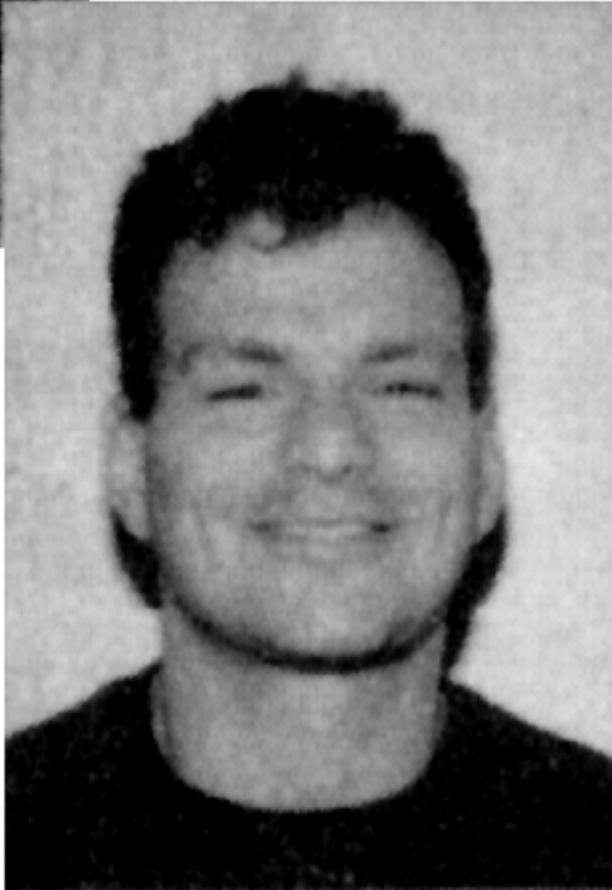


Back

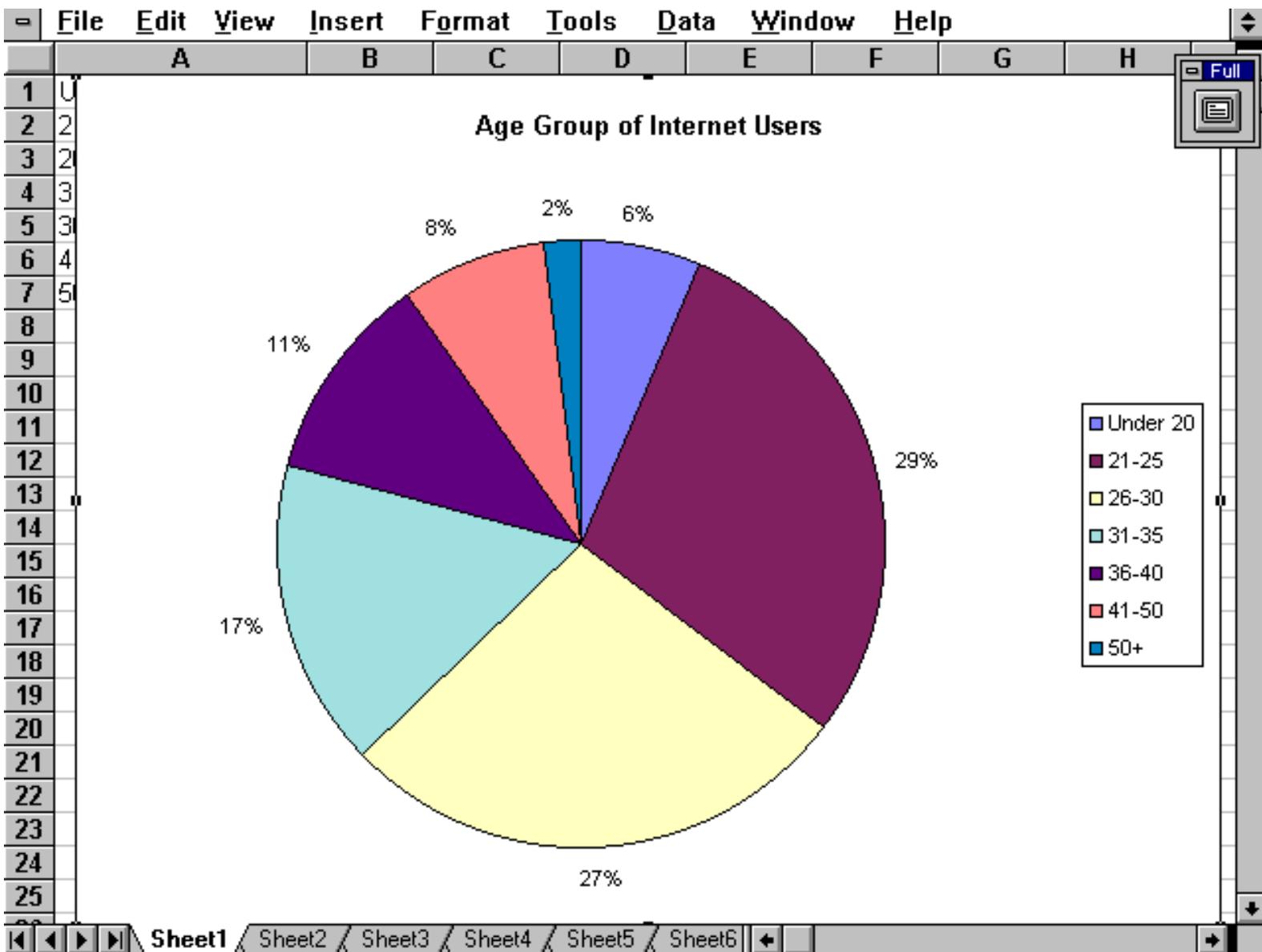






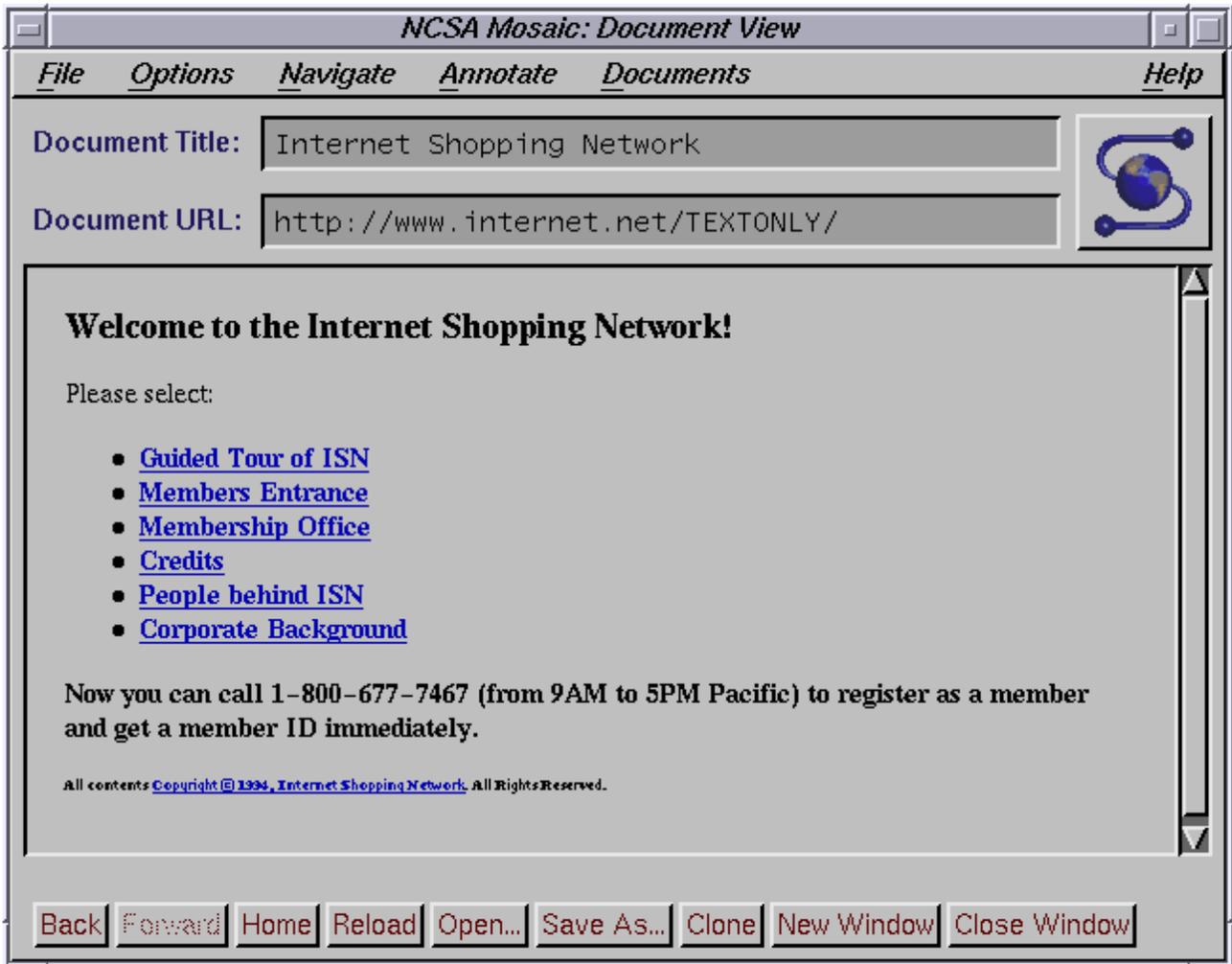


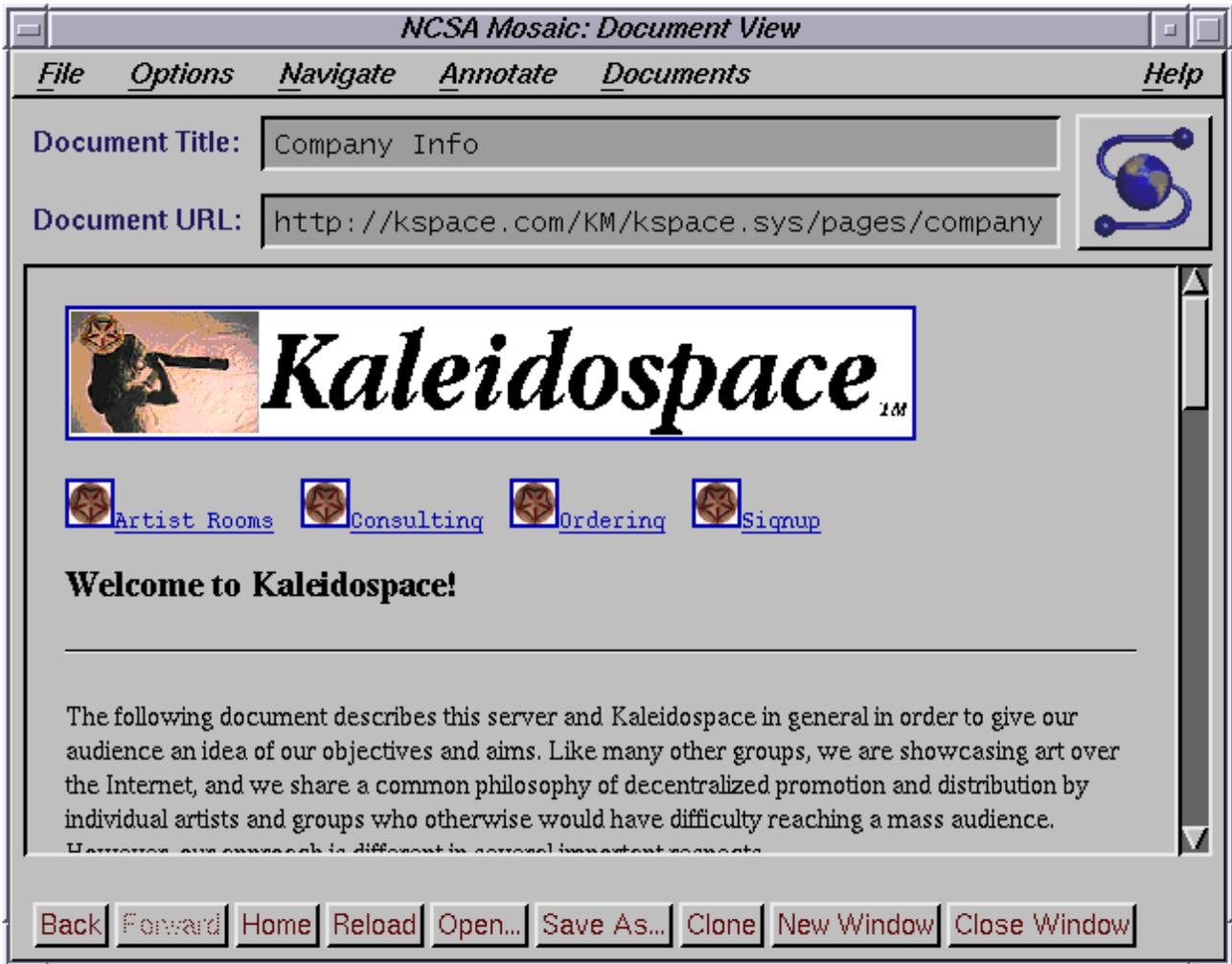


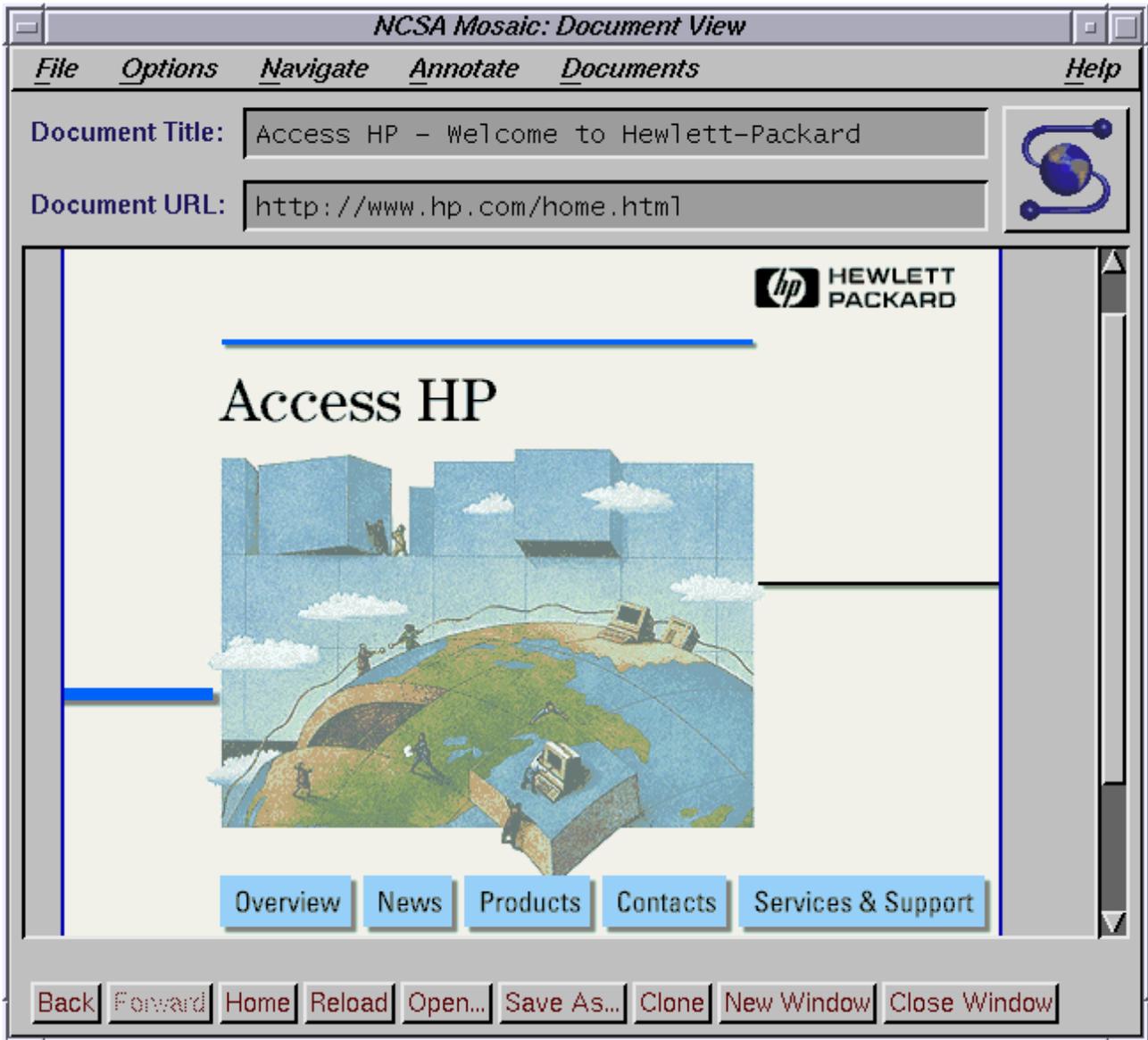


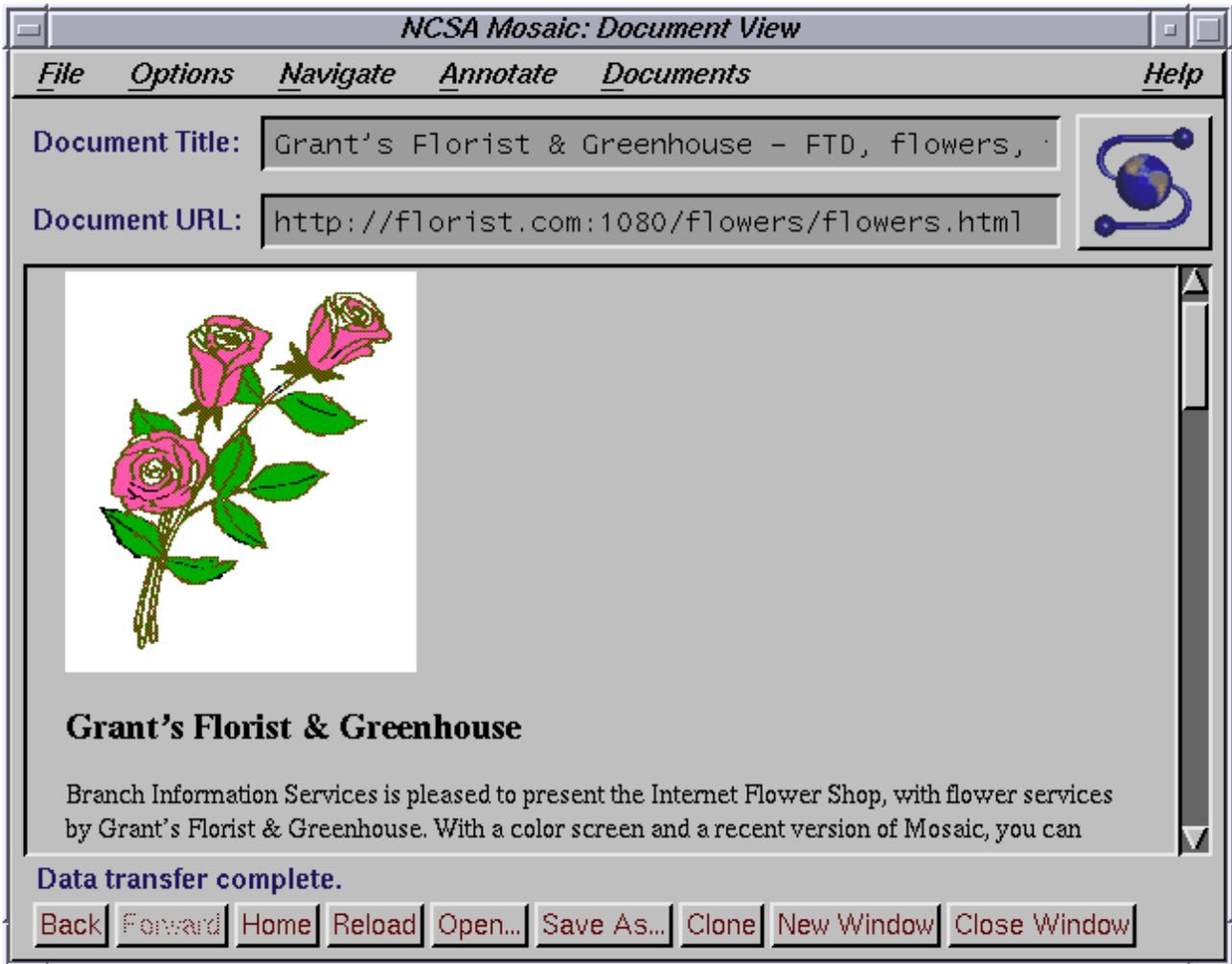


















Jon Callas on Magrathea - Open

Martin calls the meeting to order.  
 Martin selects the topic "Setup the booth."  
 Martin asks, "The first item on the agenda is setup for MacWorld. How are all the logistics going?"  
 Tamzen Cannoy exclaims, "Everybody ready? We got a lot of work to get done in a few days!"  
 Susan Giusto says, "I'm in the booth now. The phone line is in, the machines are delivered, and the backdrop is set up."  
 Tamzen Cannoy asks, "Are there enough munchies to keep everyone going during setup?"  
 Jon Callas says, "I'm going to hit a warehouse store once I fly in. It's only a mile from the airport."  
 Jon Callas whispers to Martin, "Did you manage to get a phone call in to Dorothy?"

Jon Callas sees

-  Tamzen Cannoy
-  Susan Giusto
-  Skippy
-  MacWorld Agenda
-  Martin
-  Jon Callas
-  Demo Minutes
-  Meeting Space™

Good Meetings

Welcome to Meeting Space™

Your Doorway to Business in Cyberspace!



Jon Callas to Martin

Aside To   
 Privately  
 Say Did you manage to get a phone call in to  
 Act Dorothy?

Good Meetings



Top Level Menu

Commands Options Bookmarks Help

Previous Menu ( Top Level Menu )

- <txt> ABOUT CYBERSEX CITY
- <txt> NEW! LIVE HOT SLUTS ONLINE!
- <txt> ABOUT INTERNET SEX NET DATES
- <txt> ABOUT UNCENSORED INTERNATIONAL SEX CALLS
- <txt> HOTTEST SWINGER MAGAZINES IN U.S.
- <txt> SWINGER OF THE WEEK (AD AND PHOTO)
- <txt> ABOUT X-RATED VIDEO CATALOGS
- <txt> AMATEURS X-RATED CATALOG (60 SEC.WAIT)
- <txt> NEW X-RATED CATALOG (60 SEC. WAIT)
- <txt> GAY X-RATED CATALOG (60 SEC. WAIT)
- <txt> GEN'L & S&M CATALOG (60 SEC. WAIT)
- >>> THE SPECIAL DATING LINES

MS-DOS Prompt

SWAIS Search Results Items: 32

| #    | Score  | Source            | Title                                    | Lines |
|------|--------|-------------------|------------------------------------------|-------|
| 001: | [1000] | <UNESCO-DARE-Soc> | The Isaac N. Thut World Education        | 43    |
| 002: | [1000] | <EDPrograms>      | Vocational Education--Business and Educa | 15    |
| 003: | [1000] | <ANU-SocSci-WWW-> | National Trade Data Bank <NTDB> on the I | 122   |
| 004: | [ 900] | <UNESCO-DARE-Soc> | European Centre for Leisure and E        | 50    |
| 005: | [ 900] | <UNESCO-DARE-Soc> | Bureau Africain des Sciences de l        | 62    |
| 006: | [ 879] | <EDPrograms>      | Adult Education--National Adult Educatio | 16    |
| 007: | [ 800] | <UNESCO-DARE-Soc> | The Economics Association UN             | 56    |
| 008: | [ 800] | <UNESCO-DARE-Soc> | International Association of Educ        | 63    |
| 009: | [ 800] | <UNESCO-DARE-Soc> | Social Science Education Consorti        | 66    |
| 010: | [ 800] | <UNESCO-DARE-Soc> | Association for World Education          | 37    |
| 011: | [ 800] | <UNESCO-DARE-Soc> | Laboratory of Education for Peace        | 40    |
| 012: | [ 760] | <EDPrograms>      | Adult Education--State Program Analysis  | 16    |
| 013: | [ 723] | <EDPrograms>      | Vocational Education--Educational Progra | 17    |
| 014: | [ 716] | <EDPrograms>      | Indian Education--Fellowships for Indian | 16    |
| 015: | [ 700] | <UNESCO-DARE-Soc> | National Council for the Social S        | 51    |
| 016: | [ 700] | <UNESCO-DARE-Soc> | International Institutes on Peace        | 37    |
| 017: | [ 700] | <UNESCO-DARE-Soc> | Malmv School of Education, Depart        | 35    |
| 018: | [ 700] | <UNESCO-DARE-Soc> | Sangyo Keiei Kenkyujo UNESCO             | 76    |

<space> selects, arrows move, w for keywords, s for sources, ? for help  
 quake.think.co 10:28

C:\TRUMPET\TELW.EXE

| SWAIS # | Score   | Source            | Search Results Title                     | Items: Line |
|---------|---------|-------------------|------------------------------------------|-------------|
| 001:    | [ 1000] | (ANU-CAUT-Academ) | Use Of Multimedia In Education And Scien | 2           |
| 002:    | [ 875]  | (ANU-CAUT-Academ) | Science For Non-Science Students         | 2           |
| 003:    | [ 875]  | (ANU-CAUT-Academ) | Field Education In Overseas Countries Fo | 2           |
| 004:    | [ 875]  | (ANU-CAUT-Academ) | Developing An Assessment Schedule And Pr | 2           |
| 005:    | [ 750]  | (ANU-CAUT-Academ) | Theory-Practice Links In Teacher Educati | 2           |
| 006:    | [ 750]  | (ANU-CAUT-Academ) | The Development And Implementation Of St | 2           |
| 007:    | [ 750]  | (ANU-CAUT-Academ) | Non-English Language Education           | 2           |
| 008:    | [ 750]  | (ANU-CAUT-Academ) | International Education                  | 2           |
| 009:    | [ 625]  | (ANU-CAUT-Academ) | Teaching/Learning In Environmental Educa | 2           |
| 010:    | [ 625]  | (ANU-CAUT-Academ) | Teaching Pharmacology In Medical, Para-m | 2           |
| 011:    | [ 625]  | (ANU-CAUT-Academ) | Teacher Postgraduate Education In Mathem | 2           |
| 012:    | [ 625]  | (ANU-CAUT-Academ) | Teacher Education                        | 2           |
| 013:    | [ 625]  | (ANU-CAUT-Academ) | Student Learning In Higher Education In  | 2           |
| 014:    | [ 625]  | (ANU-CAUT-Academ) | Social Work Practice In The Area Of Ethn | 2           |
| 015:    | [ 625]  | (ANU-CAUT-Academ) | Secondary Mathematics Teacher Preservice | 2           |
| 016:    | [ 625]  | (ANU-CAUT-Academ) | Science For Students With Limited Backgr | 2           |
| 017:    | [ 625]  | (ANU-CAUT-Academ) | Recognition Of Prior Learning In Adult E | 2           |
| 018:    | [ 625]  | (ANU-CAUT-Academ) | Primary Preservice Teacher Education In  | 2           |

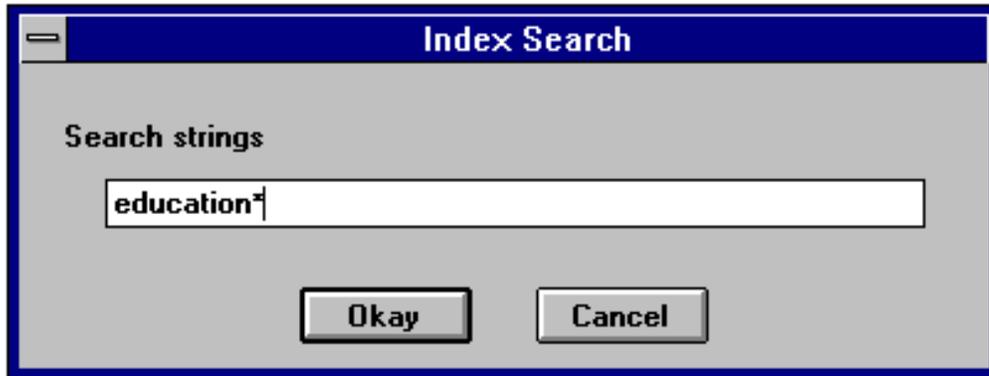
<space> selects, arrows move, w for keywords, s for sources, ? for help

Getting requested item

Commands Options Bookmarks Help

Previous Menu ( Top Level Menu )

- >>> Introduction to the Electronic Newsstand
- <txt> Notice of Copyright and General Disclaimer – Please Read
- >>> Magazines, Periodicals, and Journals (all titles)
- >>> Business Publications and Resources
- >>> Electronic Bookstore
- >>> Music! (8 magazines and 80,000 CD titles)
- >>> Travel/Lufthansa Takes Off...
- >>> The Electronic Car Showroom(tm) (Featuring Lincoln-Mercury)
- >>> HeadsUp: A personalized daily news service from INDIVIDUAL, Inc.
- >>> To get to The Internet Company, please use gopher.internet.com
- <idx> Search All Electronic Newsstand Articles by Keyword



The image shows a dialog box titled "Index Search". It has a title bar with a close button on the left. The main area contains the text "Search strings" above a text input field. The input field contains the word "education" followed by a cursor. Below the input field are two buttons: "Okay" and "Cancel".

NCSA Mosaic for MS Windows

File Edit Options Navigate Annotate Starting Points Personal Help

Document Title: UW Radiology Webserver

Document URL: http://www.rad.washington.edu/



# UW Radiology Webserver

Welcome to the University of Washington Department of Radiology Webserver!

---

[WHAT'S NEW ON OUR SERVER](#)

NUM

NCSA Mosaic for MS Windows

File Edit Options Navigate Annotate Starting Points Personal Help

Document Title: Grand River Elementary Students

Document URL: http://web.cal.msu.edu/JSRI/GR/BradClass.thumb.html

# Grand River Elementary School

Lansing Michigan

Brad Marshall's 5th Grade Class

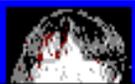
For this directory with no thumbnails click [here](#).



[Brad Marshall](#) *Teacher*



[Gary J LaPointe](#) *Computer Instructor*



NUM

NCSA Mosaic for MS Windows

File Edit Options Navigate Annotate Starting Points Personal Help

Document Title: Claremont High School World-Wide-Web Home Page

Document URL: http://www.cusd.claremont.edu

# Claremont High School

<http://WWW.cusd.claremont.edu/>  
**Claremont Unified School District**  
**Claremont, California USA**

## Welcome to Claremont High School's WWW server!

The primary purpose of this server is to provide information, resources, and services to the members of the [Claremont Unified School District](#) community; more specifically, the faculty, students, staff, and alumni of [Claremont High School](#). In many cases, the services provided to local users are of a general interest and so members of the Internet community at large are welcome to utilize any of the resources found on this server.

Since this server grew extremely rapidly and continues to grow, it is likely that there are some

NUM

NCSA Mosaic for MS Windows

File Edit Options Navigate Annotate Starting Points Personal Help

Document Title: Knowledge One

Document URL: http://KnowOne\_WWW.sonoma.edu



**(800) 505-KNOW** (in CA)

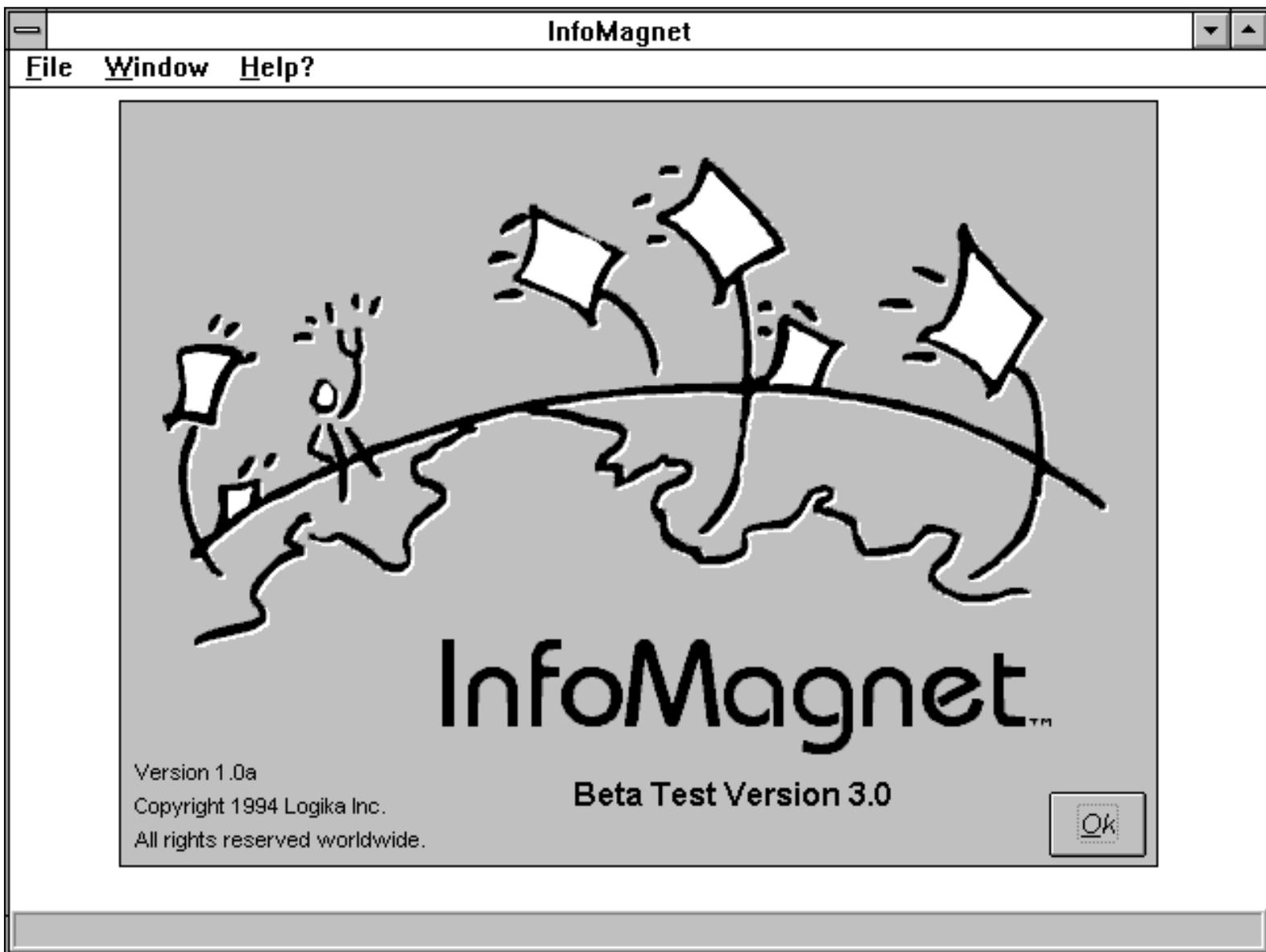
---

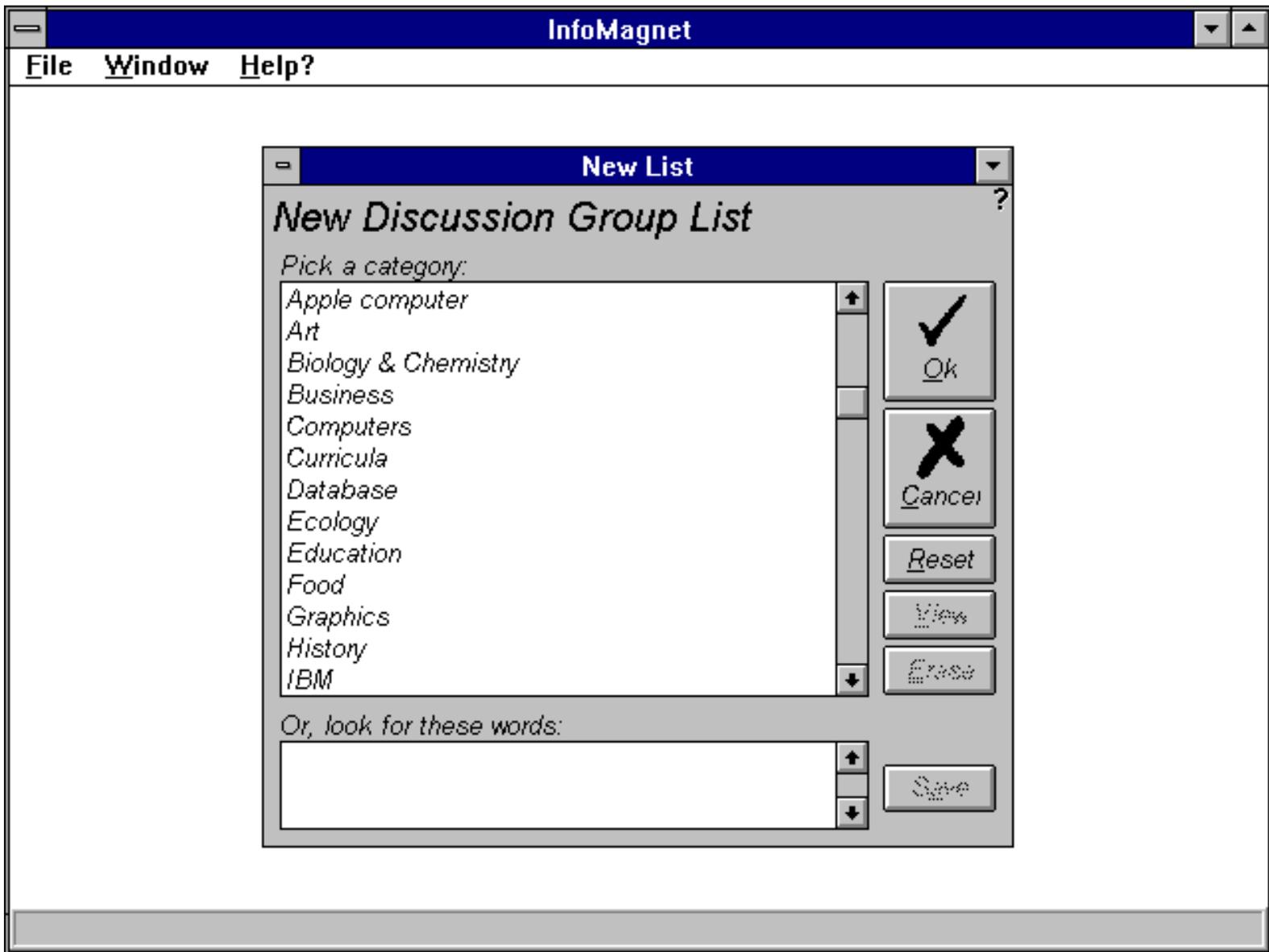
index: [How to reach us...](#) [What we do...](#) [Update Service...](#) [PriceSheet...](#) [Knowledge Navigators...](#)  
[Free Trial Offer...](#) [Join the KO mailing list...](#) [Quote for services](#)

---

Need information fast? Knowledge One.

NUM





**InfoMagnet - [Education]**

**File Window Help?**

*peek* *search* *join* ?

**AAAE-C: American Assoc. for Agricultural Education (central region)**

AAAE-E: American Assoc. for Agricultural Education (eastern region)

AAAE-S: American Assoc. for Agricultural Education (southern region)

AAAE-W: American Assoc. for Agricultural Education (western region)

AAAE: American Assoc. for Agricultural Education (all regions)

ACADEMIA: Academia - Forum on Higher Education in Israel

ACADNEWS: Acadnews - Bulletin on Higher Education in Israel

ACEWEST: Ag Communicators in Education

ADLTED-L: Canadian Adult Education Network

ADVISE-L: User Services List (North Dakota Higher Education Computer Network, Fargo, ND)

AACP-L: Apple Education Consultants Program

AELIST: Association for Experiential Education

AEMA-L: AEMA-L: Arizona Educational Media Association

AERA-A: AERA-A Division A: Educational Administration Forum

AERA-G: AERA-G Division G: Social Context of Education

AERA-I: AERA-I Division I: Education in the Professions

AERA-J: AERA-J Division J: Postsecondary Education

AERA-K: AERA-K Division K: Teaching and Teacher Education

AJCUEDU: AJCU Conference on Teacher Education

ALLIANCE: Alliance for International Educational and Cultural Exchange

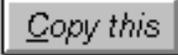
AMIED-L: American Medical Informatics Association Education PSG

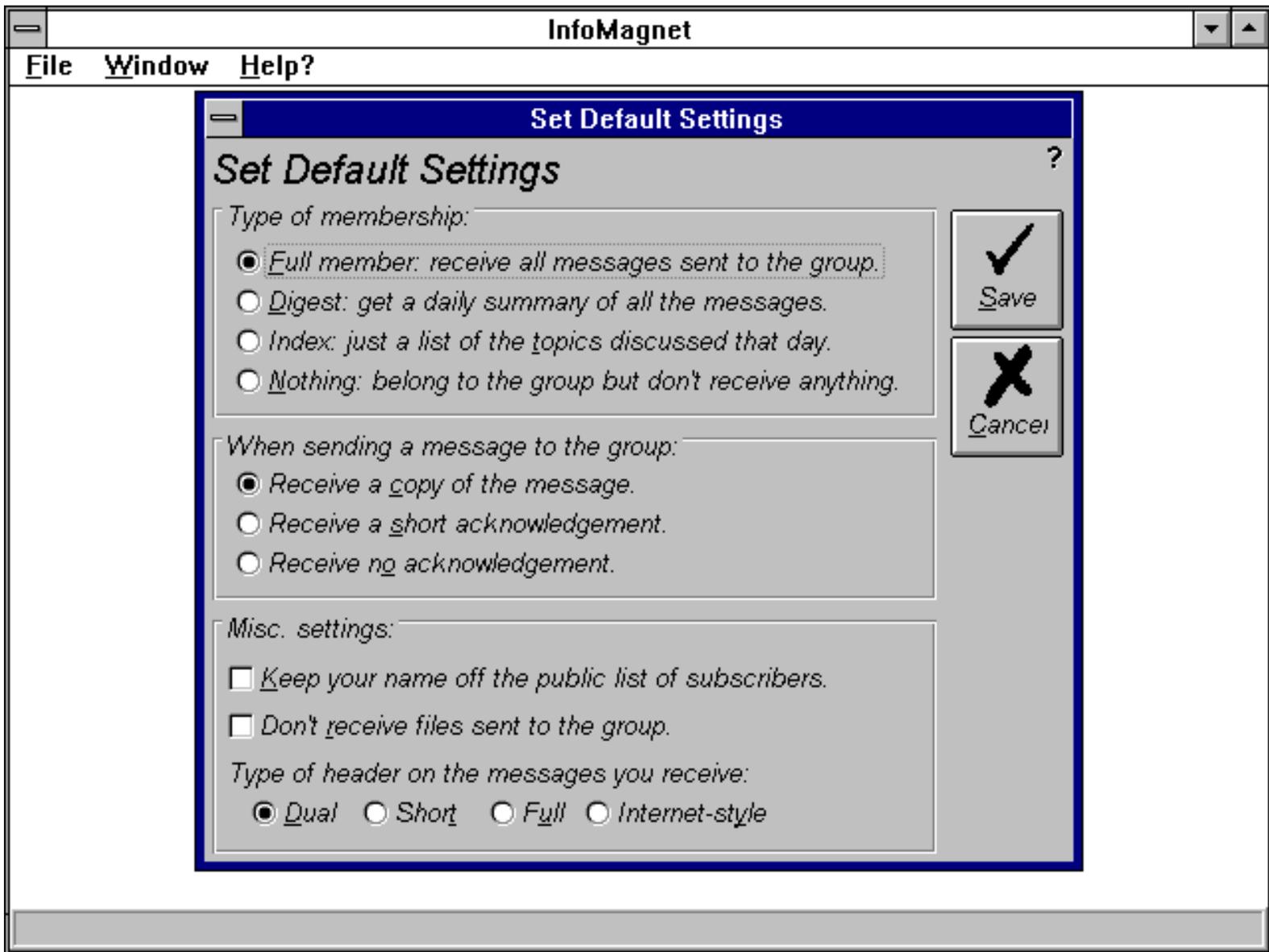
InfoMagnet - [Education]

File Window Help?

peek search join

### Peek at "EDPOLYAN"

|                                       |                                                                   |                                     |                                                                                                                                                                             |
|---------------------------------------|-------------------------------------------------------------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Group name:                           | Description:                                                      |                                     |                                                                                          |
| <input type="text" value="EDPOLYAN"/> | <input type="text" value="Education Policy Analysis Forum"/>      |                                     |                                                                                                                                                                             |
| # of participants:                    | Archives:                                                         | Membership policy:                  |                                                                                          |
| <input type="text" value="754"/>      | <input type="text" value="open to the public"/>                   | <input type="text" value="public"/> |                                                                                                                                                                             |
| Country:                              | Site:                                                             |                                     | Get a list of recent topics.                                                                                                                                                |
| <input type="text" value="USA"/>      | <input type="text" value="Arizona State University, Tempe, AZ."/> |                                     |                                                                                                                                                                             |
| Computerized administrator:           | <input type="text" value="listserv@asuum.inre.asu.edu"/>          |                                     | <br> |
| Human administrator:                  | <input type="text" value="edpolyan-request@asuum.inre.asu.edu"/>  |                                     |                                                                                                                                                                             |
| Send mail to the group at:            | <input type="text" value="edpolyan@asuum.inre.asu.edu"/>          |                                     |                                                                                                                                                                             |
| Other information:                    | <input type="text"/>                                              |                                     |                                                                                                                                                                             |



**InfoMagnet - [Education]**

**File Window Help?**

 peek     search     join    ?

---

** Peek at "EDTECH"** ?

|                                     |                                                                                      |                                                                                                                     |
|-------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>Group name:</i>                  | <i>Description:</i>                                                                  |                                  |
| <input type="text" value="EDTECH"/> | <input type="text" value="EDTECH - Educational Technology"/>                         |                                                                                                                     |
| <i># of participants:</i>           | <i>Archives:</i>                                                                     | <i>Membership policy:</i>                                                                                           |
| <input type="text" value="1742"/>   | <input type="text" value="open to the public"/>                                      | <input type="text" value="public"/>                                                                                 |
| <i>Country:</i>                     | <i>Site:</i>                                                                         | <br>Get a list of recent topics. |
| <input type="text" value="USA"/>    | <input type="text" value="Michigan State University, East Lansing, MI, 48824-1042"/> |                                                                                                                     |
| <i>Computerized administrator:</i>  | <input type="text" value="listserv@msu.edu"/>                                        |                                                                                                                     |
| <i>Human administrator:</i>         | <input type="text" value="edtech-request@msu.edu"/>                                  |                                                                                                                     |
| <i>Send mail to the group at:</i>   | <input type="text" value="edtech@msu.edu"/>                                          |                                                                                                                     |
| <i>Other information:</i>           | <input type="text"/>                                                                 |                                                                                                                     |

**InfoMagnet - [Education]**

**File Window Help?**

 peek     search     join

---

**Peek at "HYPEREDU"**

*Group name:*     *Description:*

*# of participants:*     *Archives:*     *Membership policy:*

*Country:*     *Site:*

*Computerized administrator:*

*Human administrator:*

*Send mail to the group at:*

*Other information:*

 Done

 Topics

Get a list of recent topics.

 Advanced

 Copy this

InfoMagnet - [Education]

File Window Help?

peek search join

### Peek at "MIDDLE-L"

Group name:  Description:

# of participants:  Archives:  Membership policy:

Country:  Site:

Computerized administrator:

Human administrator:

Send mail to the group at:

Other information:

Get a list of recent topics.

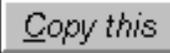
**InfoMagnet - [Education]**

**File Window Help?**

 peek     search     join

---

**Peek at "VOCNET"**

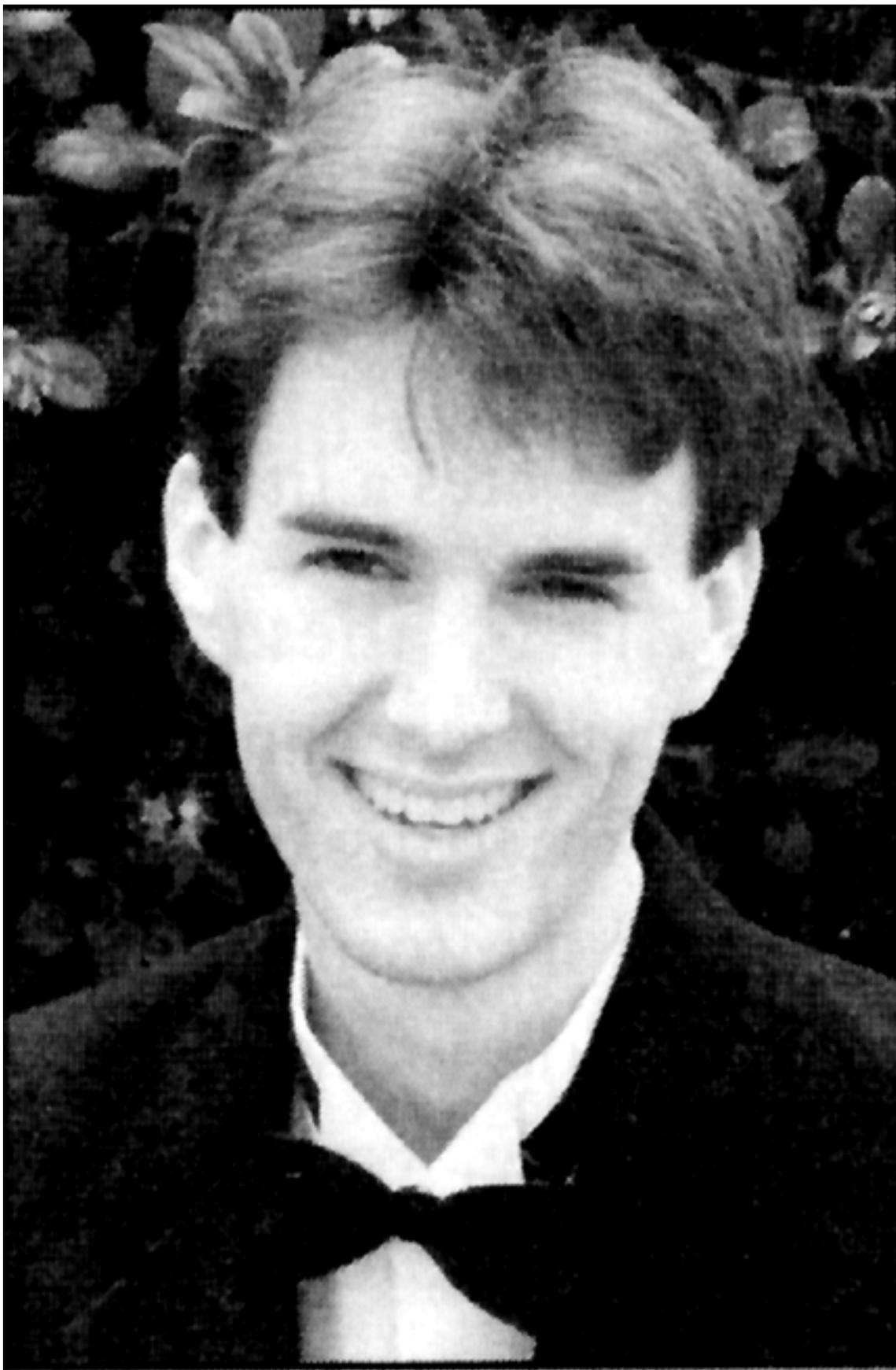
|                                     |                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
|-------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| <b>Group name:</b>                  | <b>Description:</b>                                                                       | <br><br>Get a list of recent topics.<br><br> |                                     |
| <input type="text" value="VOCNET"/> | <input type="text" value="VOCNET - NCRVE UC Berkeley Vocational Education D"/>            |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b># of participants:</b>           | <b>Archives:</b>                                                                          |                                                                                                                                                                                                                                                                                                                                                                                           | <b>Membership policy:</b>           |
| <input type="text" value="564"/>    | <input type="text" value="open to the public"/>                                           |                                                                                                                                                                                                                                                                                                                                                                                           | <input type="text" value="public"/> |
| <b>Country:</b>                     | <b>Site:</b>                                                                              |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <input type="text" value="USA"/>    | <input type="text" value="University of California at Berkeley, Berkeley, California 9"/> |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b>Computerized administrator:</b>  | <input type="text" value="listserv@cmsa.berkeley.edu"/>                                   |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b>Human administrator:</b>         | <input type="text" value="vocnet-request@cmsa.berkeley.edu"/>                             |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b>Send mail to the group at:</b>   | <input type="text" value="vocnet@cmsa.berkeley.edu"/>                                     |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b>Other information:</b>           | <input type="text"/>                                                                      |                                                                                                                                                                                                                                                                                                                                                                                           |                                     |

```
#  
Type=1  
Name=Electronic Serials  
Path=1/e-serials  
Host=gopher.cic.net  
Port=70  
URL: gopher://gopher.cic.net:70/11/e-serials
```

```
Press <RETURN> to continue,  
  <m> to mail, <D> to download, <s> to save, or <p> to print:
```

```
Esc-chr: ^] help: ^l? port:2 speed:19200 parity:spc echo:rem VT102 ....
```







**Terminal - FREENET.TRM**

File Edit Settings Phone Transfers Help

<<< CLEVELAND FREE-NET DIRECTORY >>>

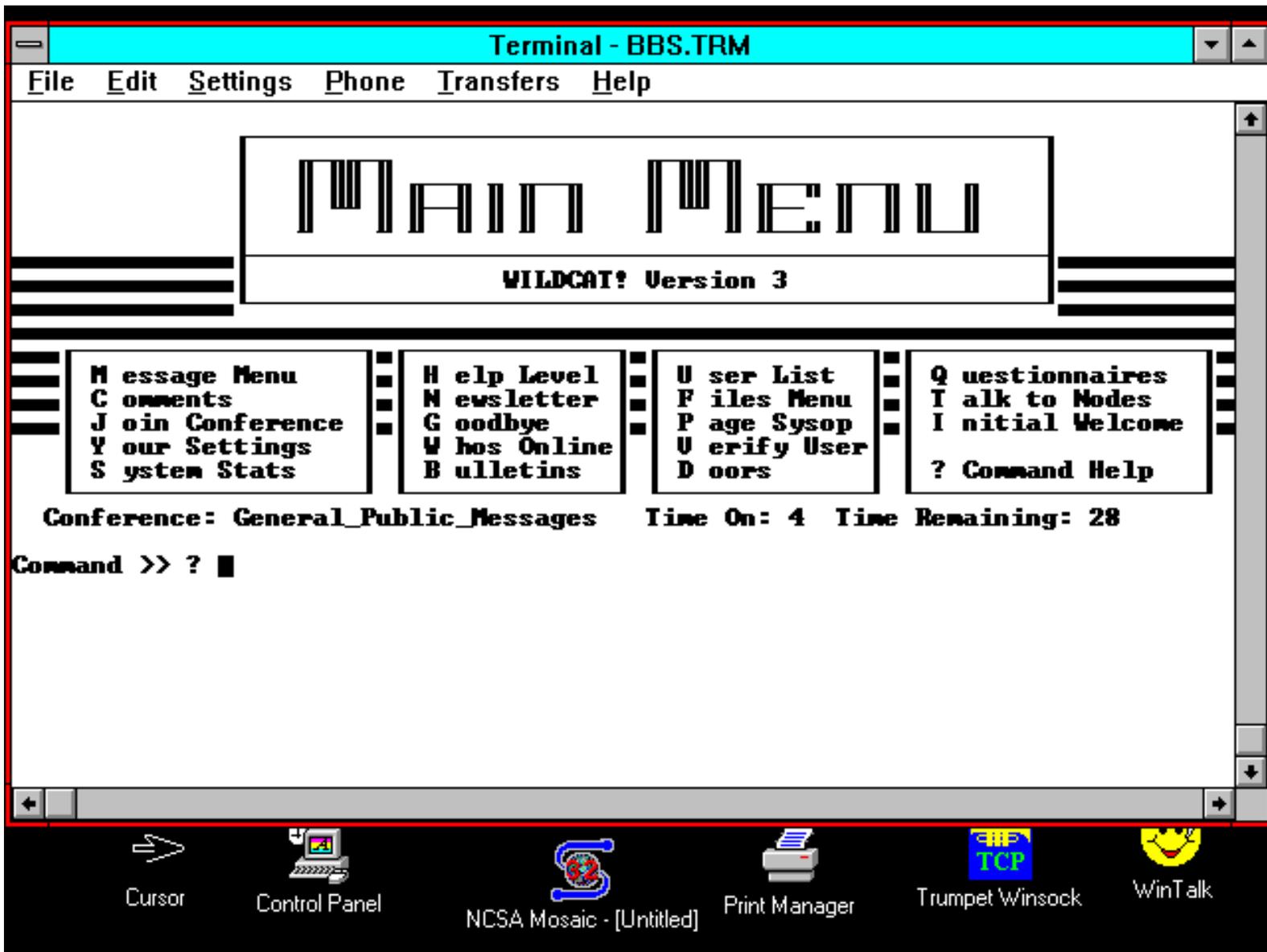
- 1 The Administration Building
- 2 The Post Office
- 3 Public Square
- 4 The Courthouse & Government Center
- 5 The Arts Building
- 6 Science and Technology Center
- 7 The Medical Arts Building
- 8 The Schoolhouse (Academy One)
- 9 The Community Center & Recreation Area
- 10 The Business and Industrial Park
- 11 The Library
- 12 University Circle
- 13 The Teleport
- 14 The Communications Center
- 15 NPTN/USA TODAY HEADLINE NEWS

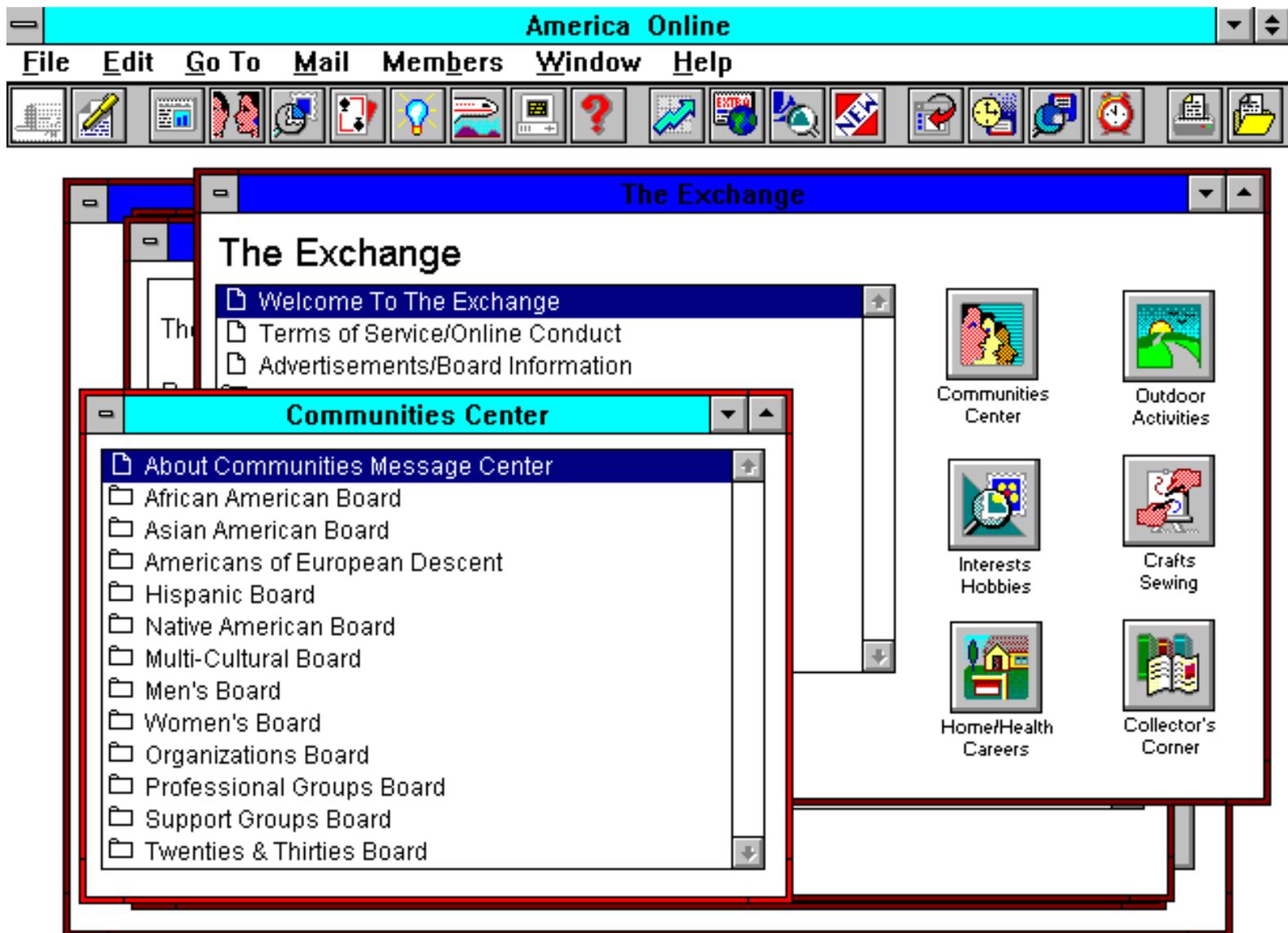
-----

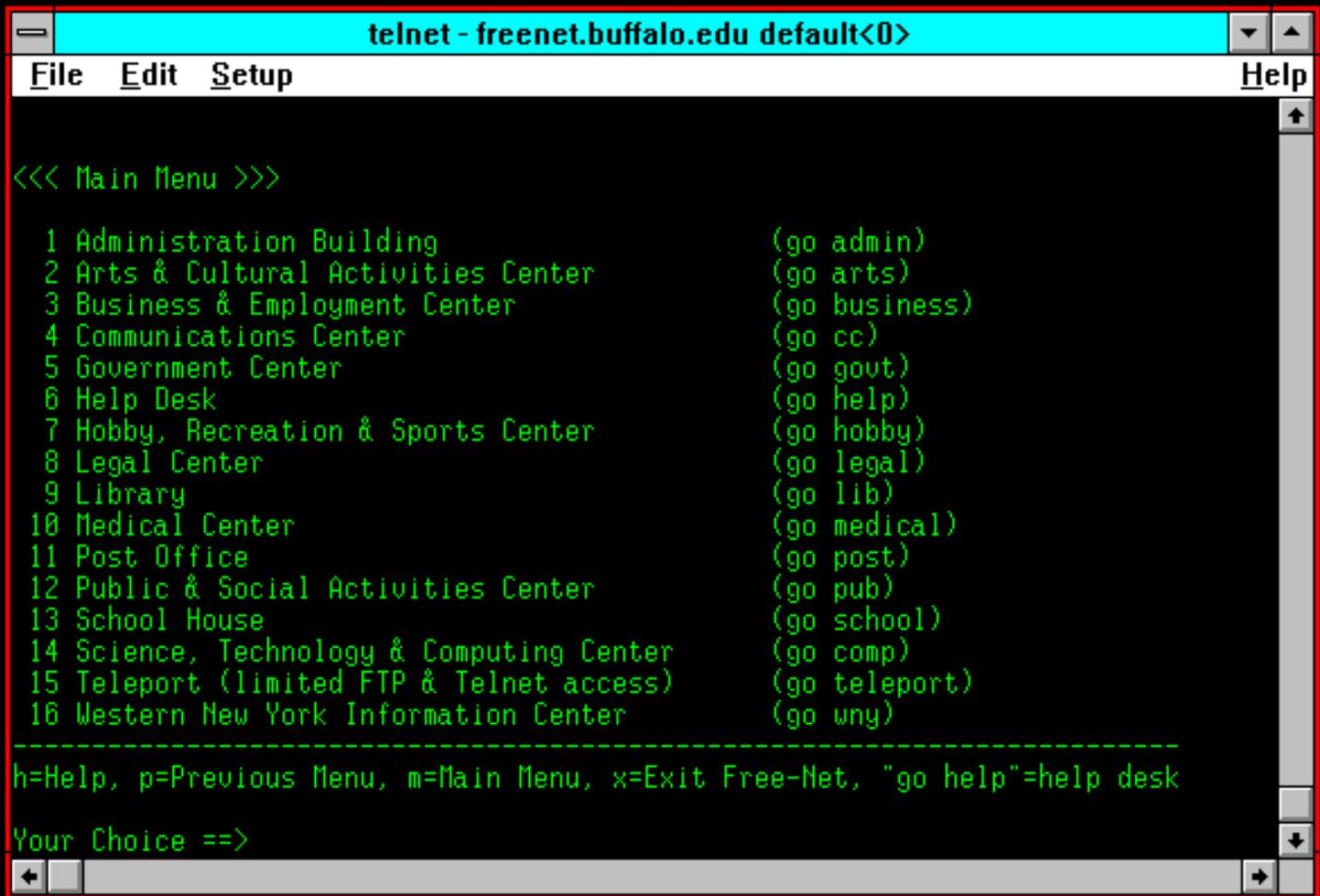
h=Help, x=Exit Free-Net, "go help"=extended help

Your Choice ==>

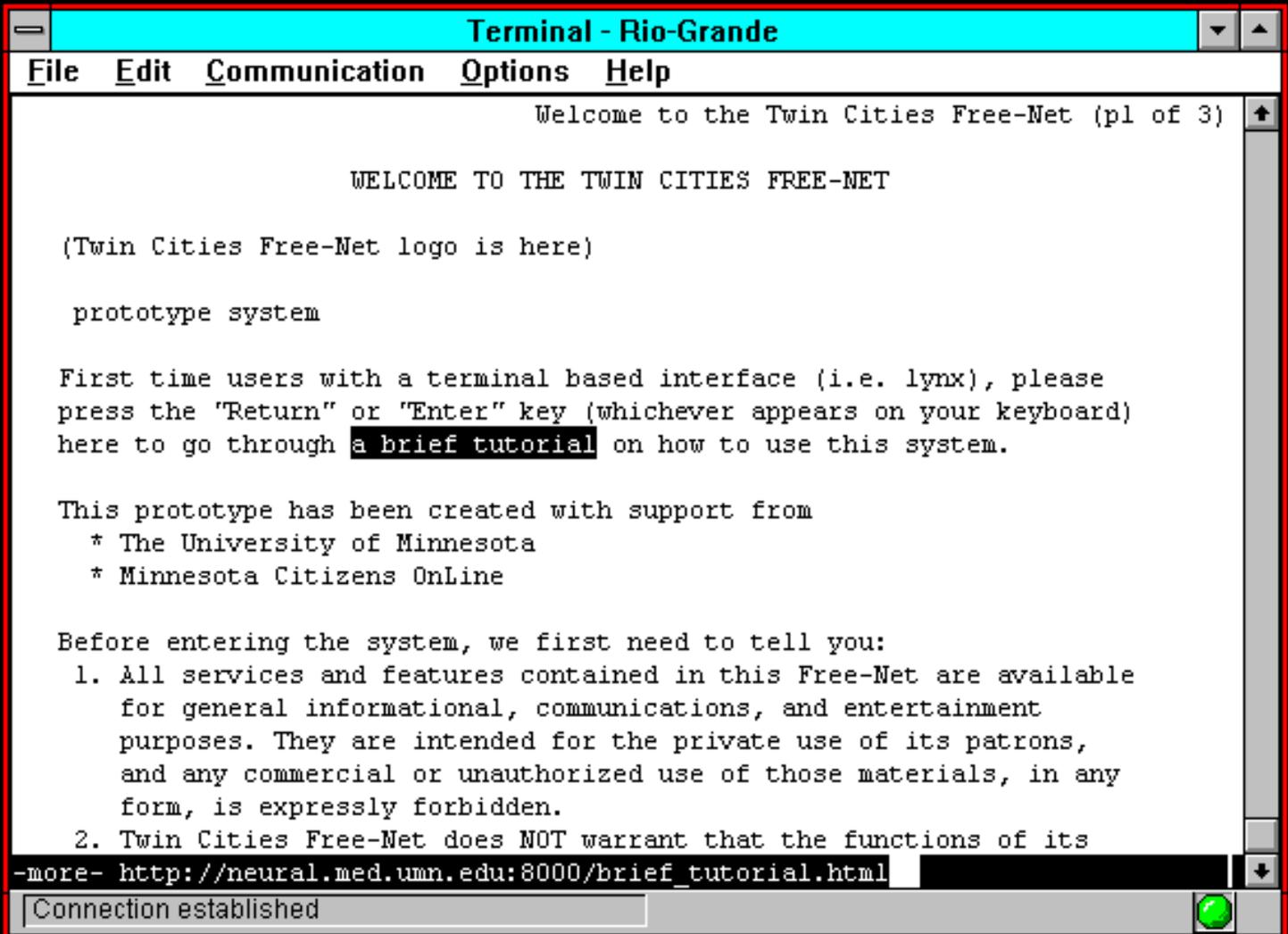


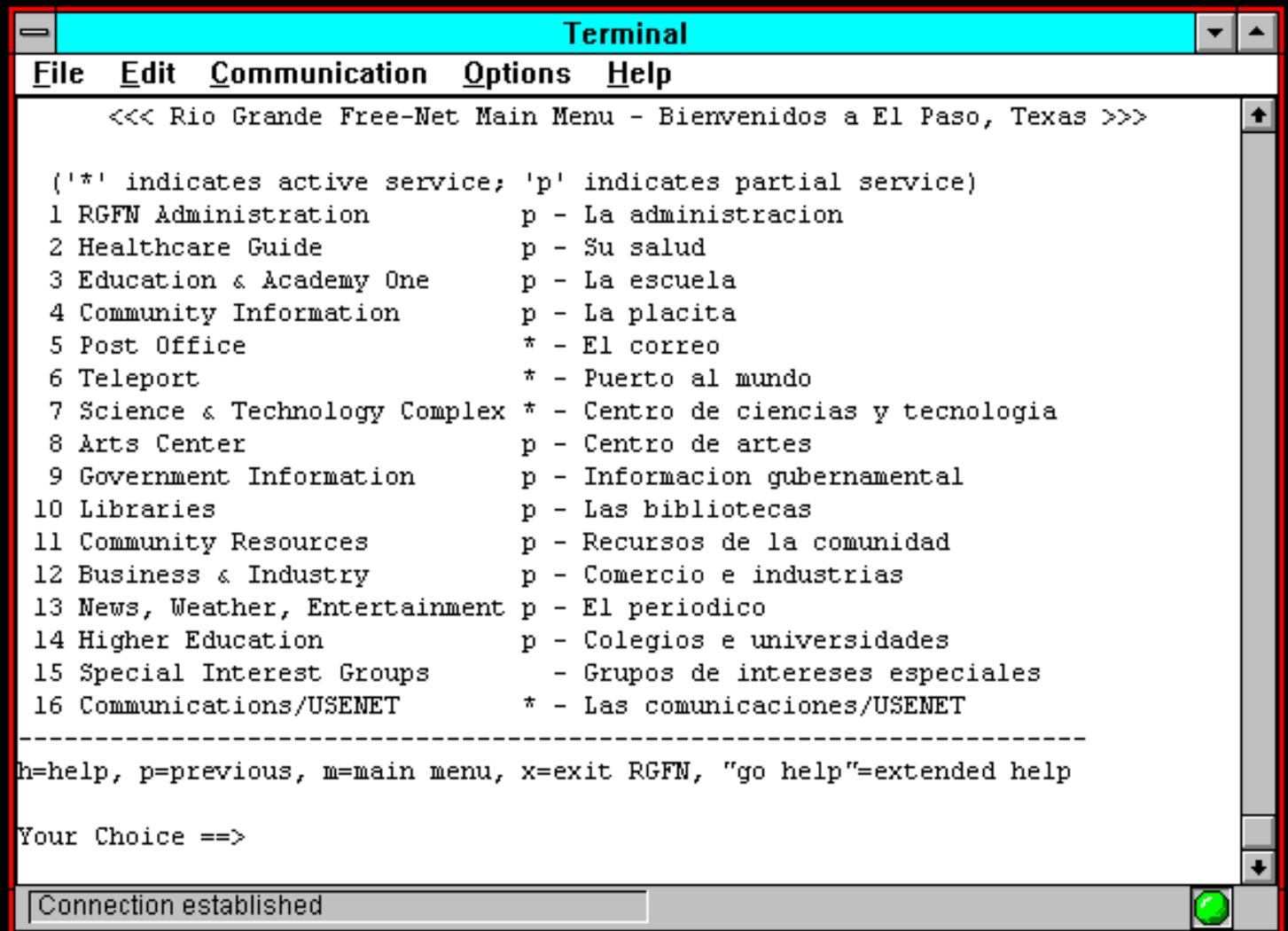














URL:



# LIBERTYNET

Linking People and Information in the Delaware Valley

Welcome to the LibertyNet Electronic City, a program of the University City Science Center. Here you will find valuable information about Philadelphia and Delaware Valley businesses, communities, and education and recreation opportunities.

If this is your first visit to LibertyNet, we'd appreciate it if you'd [click here](#) to learn more about us.

*LibertyNet wishes to thank our [many sponsors](#).*

## [What's New on LibertyNet](#)

LibertyNet grows daily, so be sure to check here for all the latest additions.

## [Search LibertyNet for Specific Information](#)

If you know what you're looking for, use this search feature to save time.

### Armadillo's WWW Server

Navigation icons: Home, Back, Forward, Stop, Reload, and a dropdown menu labeled "Armadillo's WWW".

URL: <http://riceinfo.rice.edu/armadillo/>



## Welcome to Armadillo's World Wide Web Server

Providing resources and a view of the future to the K-12 educational community.

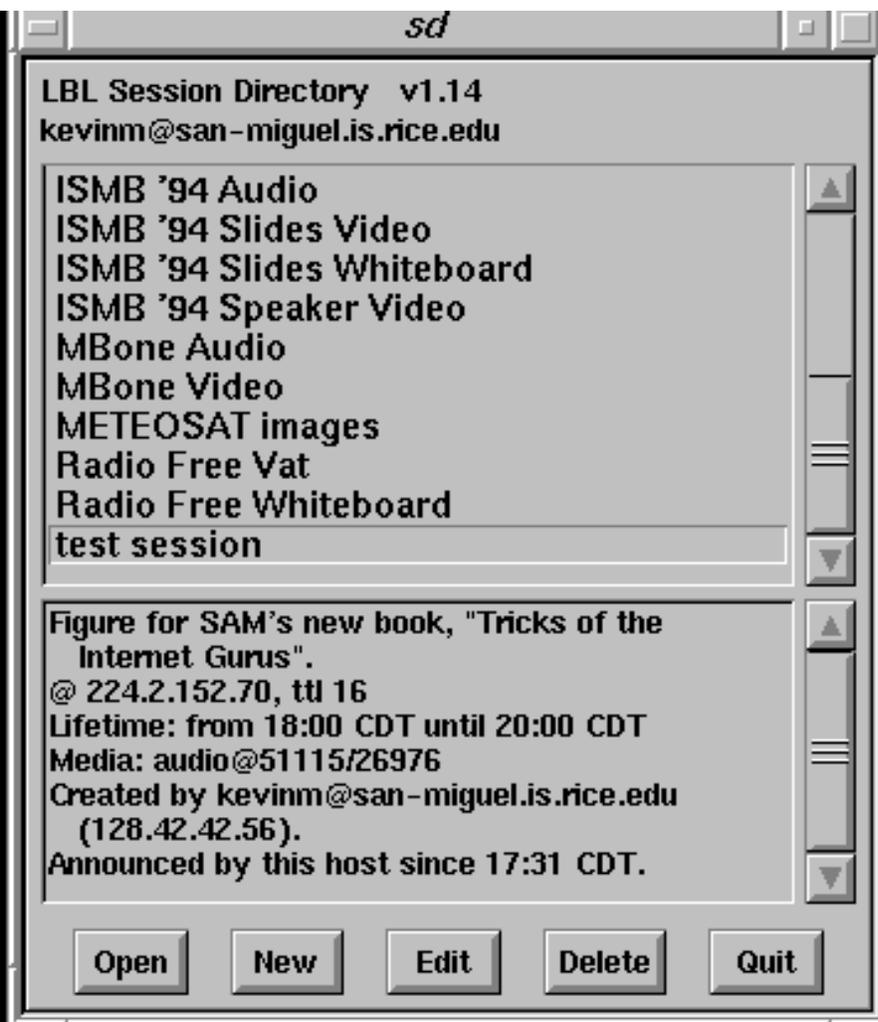


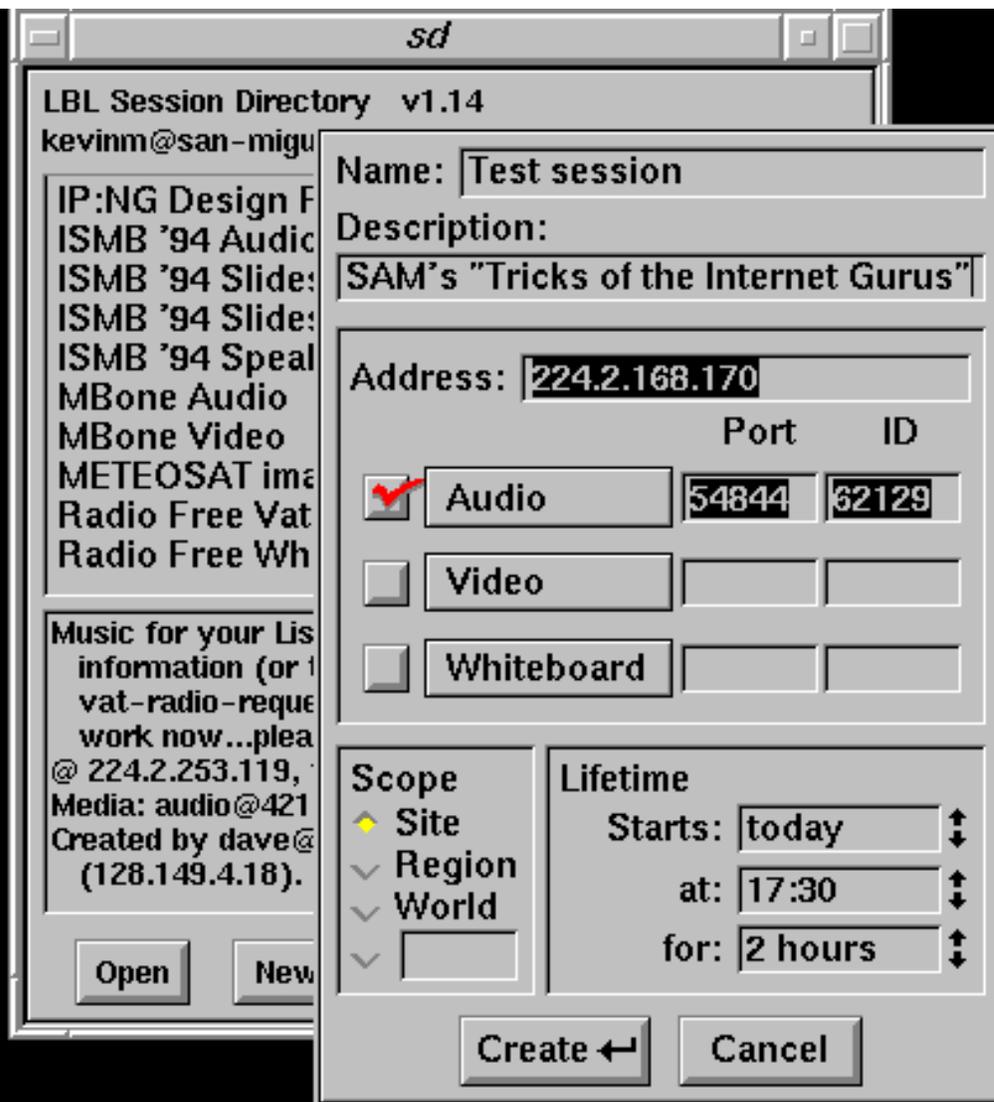
ginger

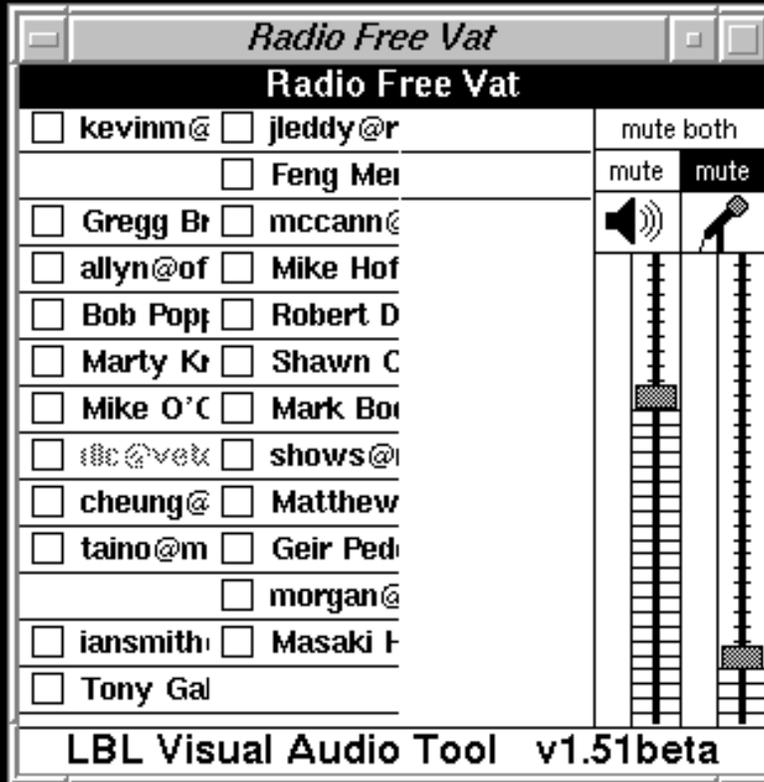
Mosaic Global History

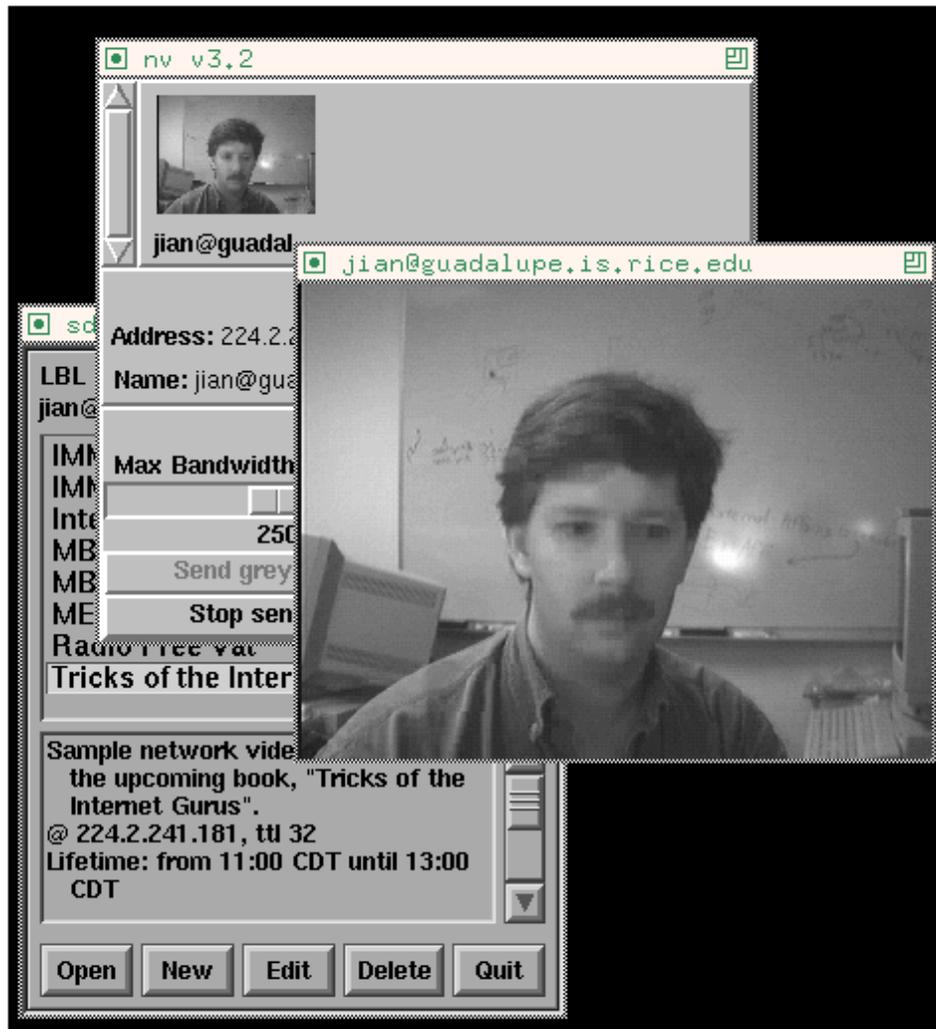
Personal Annotations

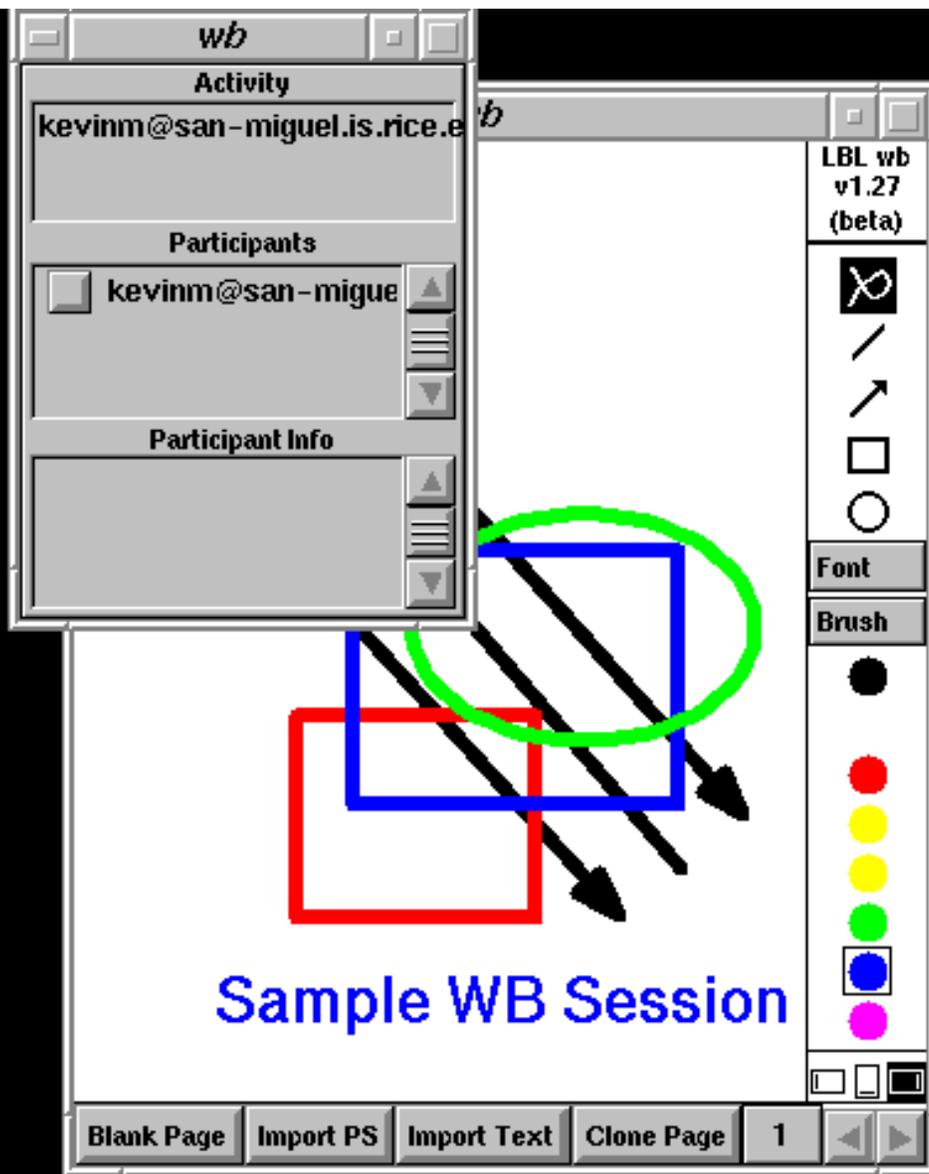
Trash











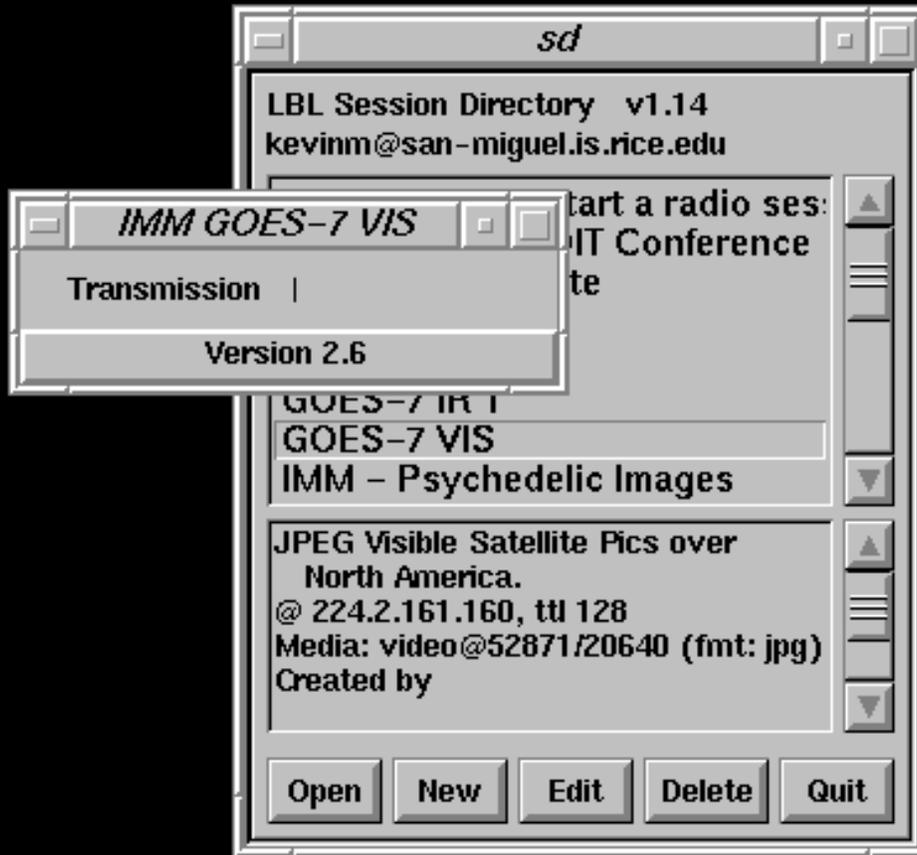
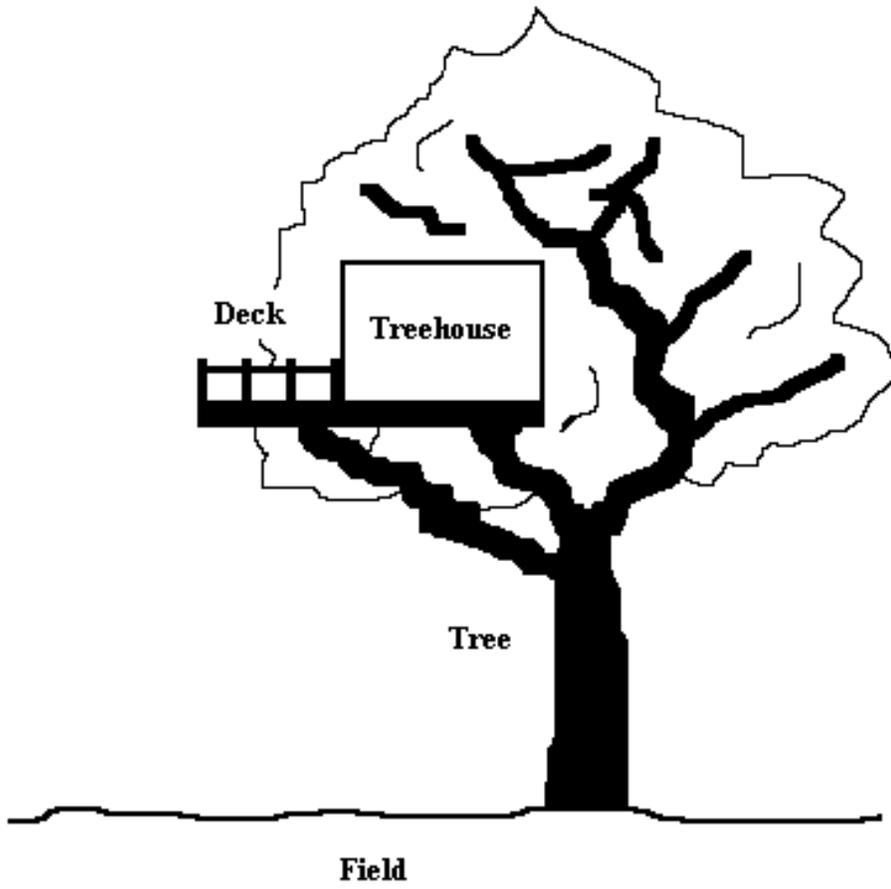
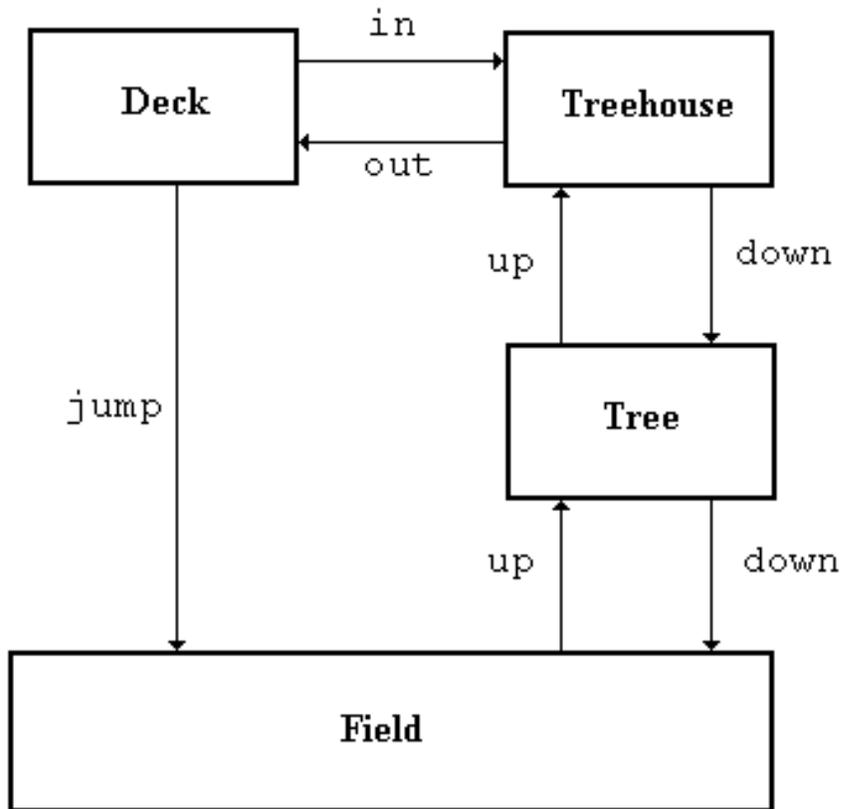




Photo courtesy of Jeanine Sih













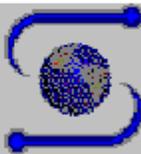


NCSA Mosaic for MS Windows

File Edit Options Navigate Annotate Starting Points Personal Help

Document Title: NCSA Home Page

Document URL: http://www.ncsa.uiuc.edu/



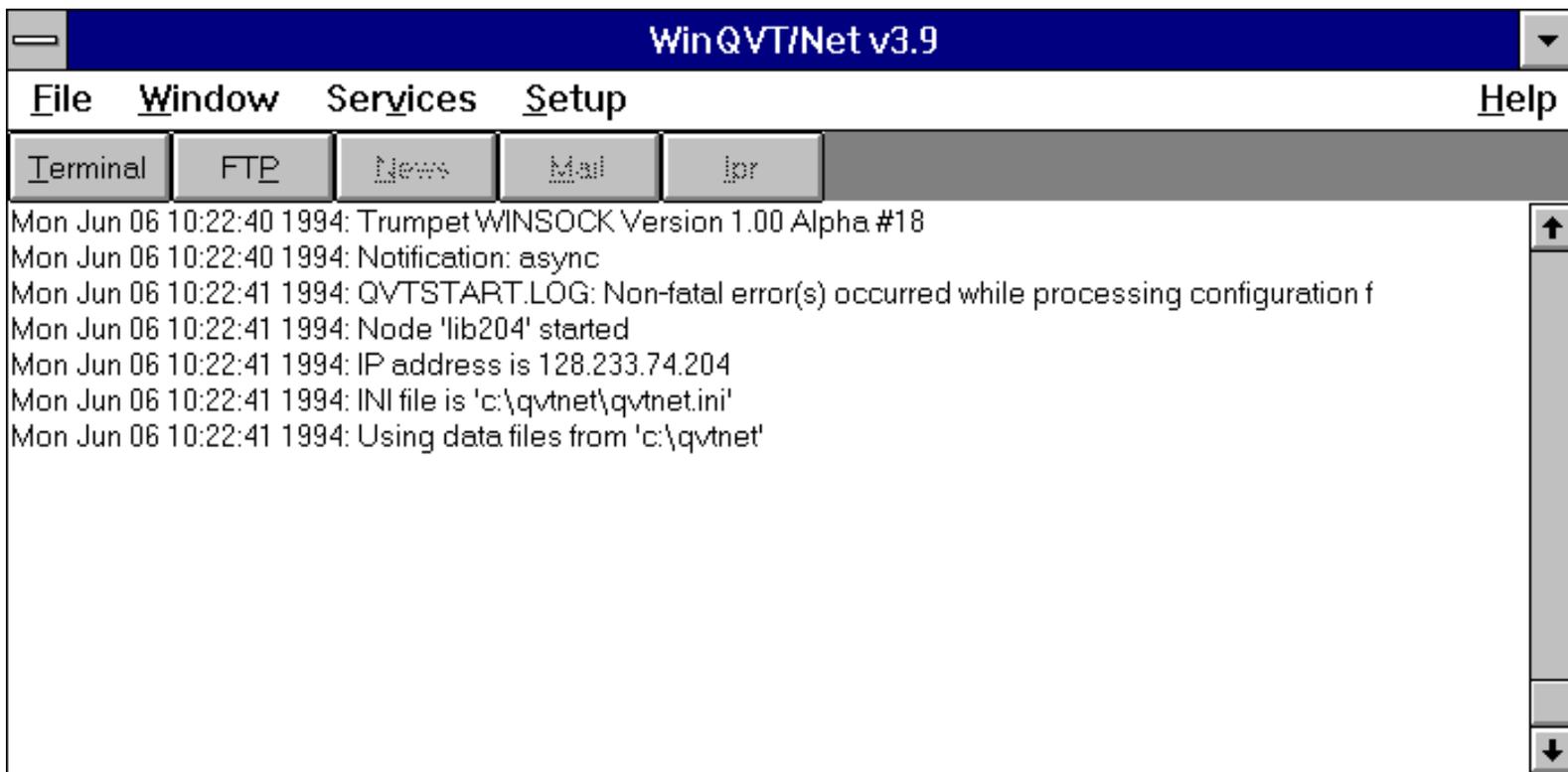
# National Center for Supercomputing Applications

University of Illinois at Urbana-Champaign

-  [News and Announcements](#)
-  [NCSA General Information](#)
-  [Multimedia Exhibits](#)
-  [NCSA Software Tools](#)
- 

http://www.ncsa.uiuc.edu/General/NCSAIntro.html

NUM



## Start Terminal Session

Host Name or IP Address:

localhost  
pc\_100.psychol.utas.edu.au  
pc\_100  
ftp.psychol.utas.edu.au  
ftp

Configuration:

|         |   |
|---------|---|
| default | ↑ |
| sundog  |   |
| sklib   |   |
| willow  |   |
| herald  |   |
| jester  |   |
| duke    |   |
| skyfox  |   |
| admin1  |   |
| admin2  |   |
| admin3  |   |
| james   | ↓ |

telnet

rlogin

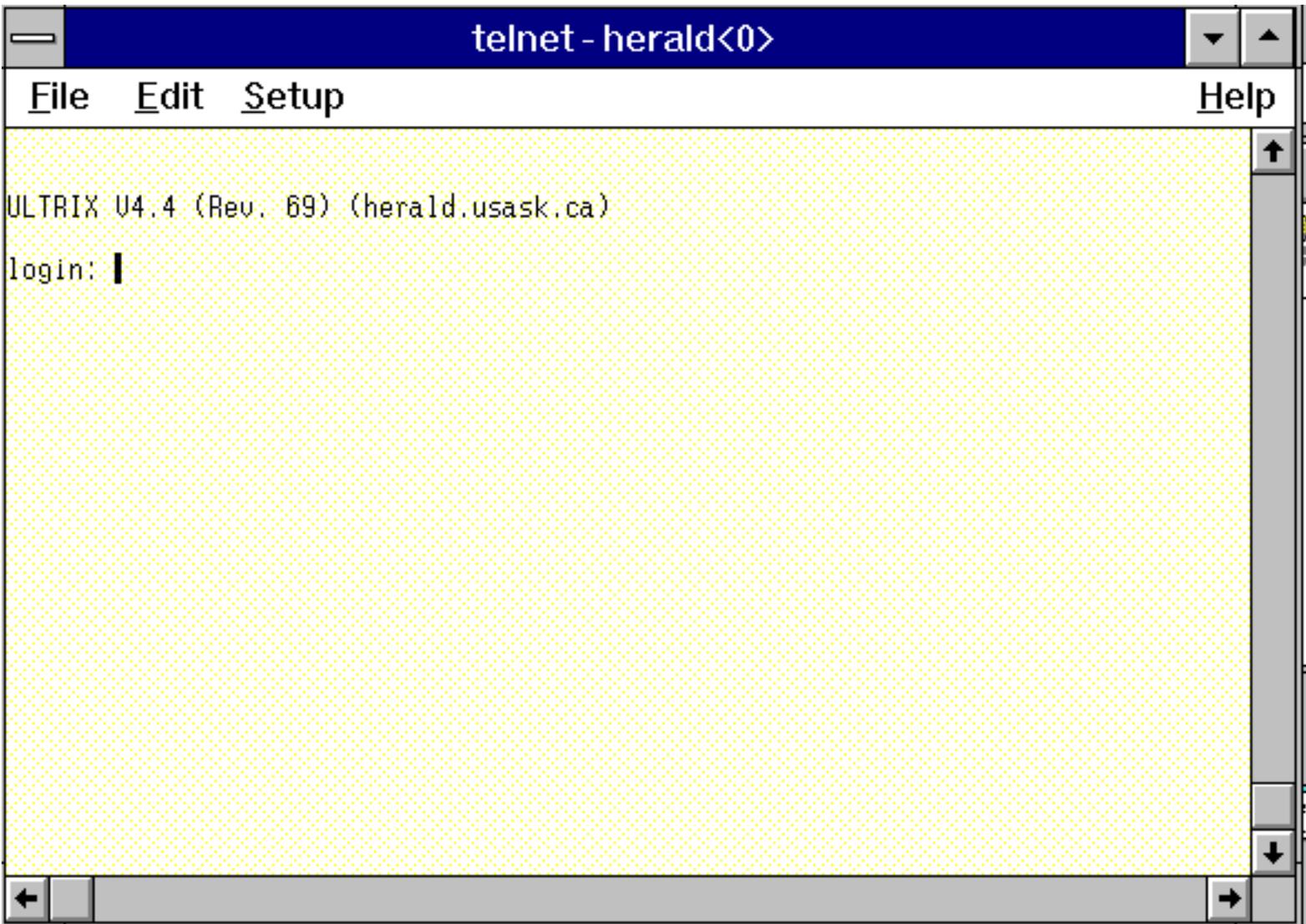
IP Port:

23

Username:

Ok

Cancel









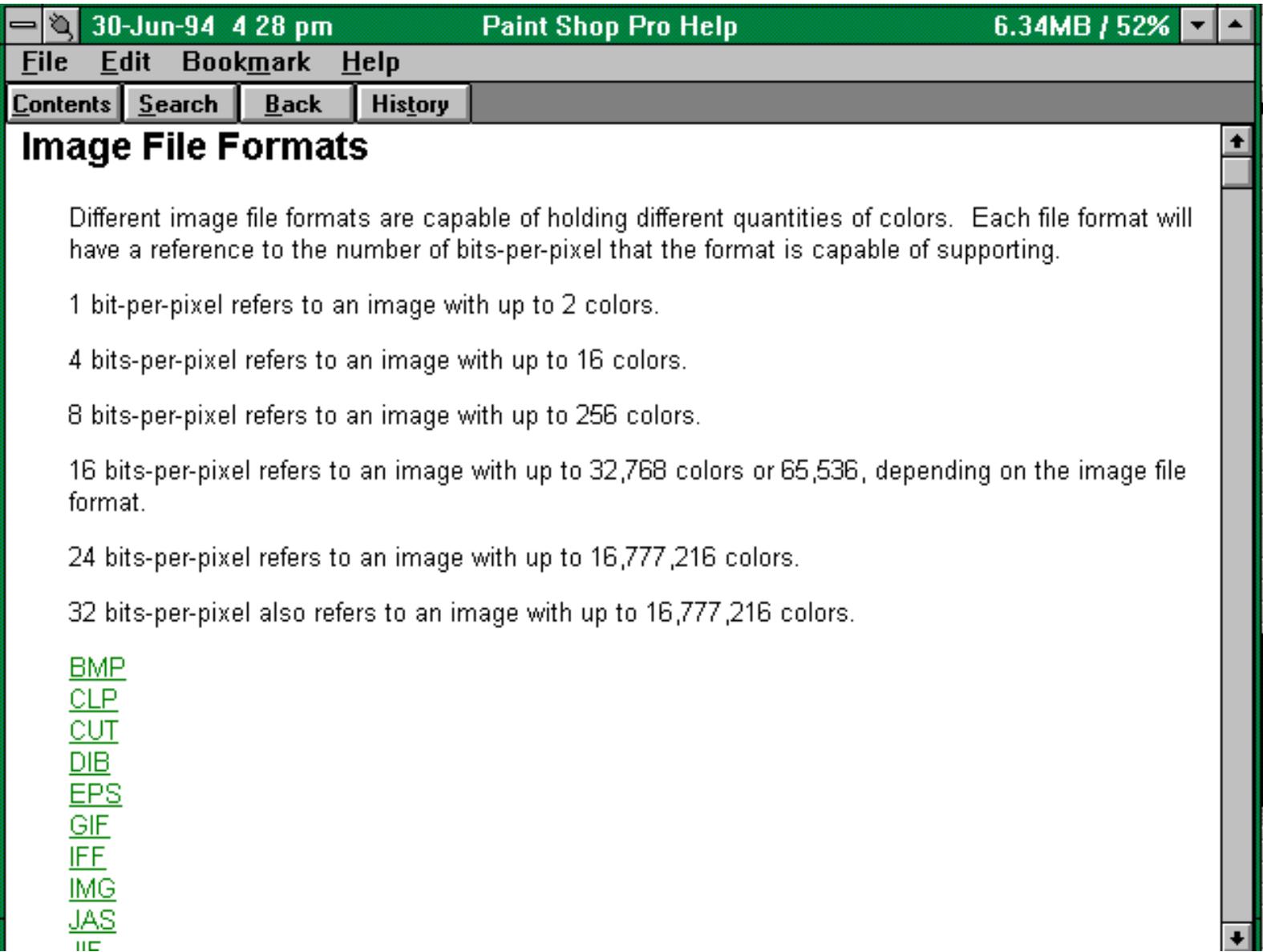


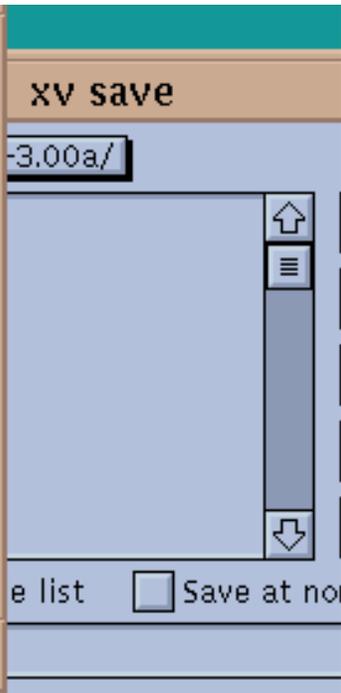
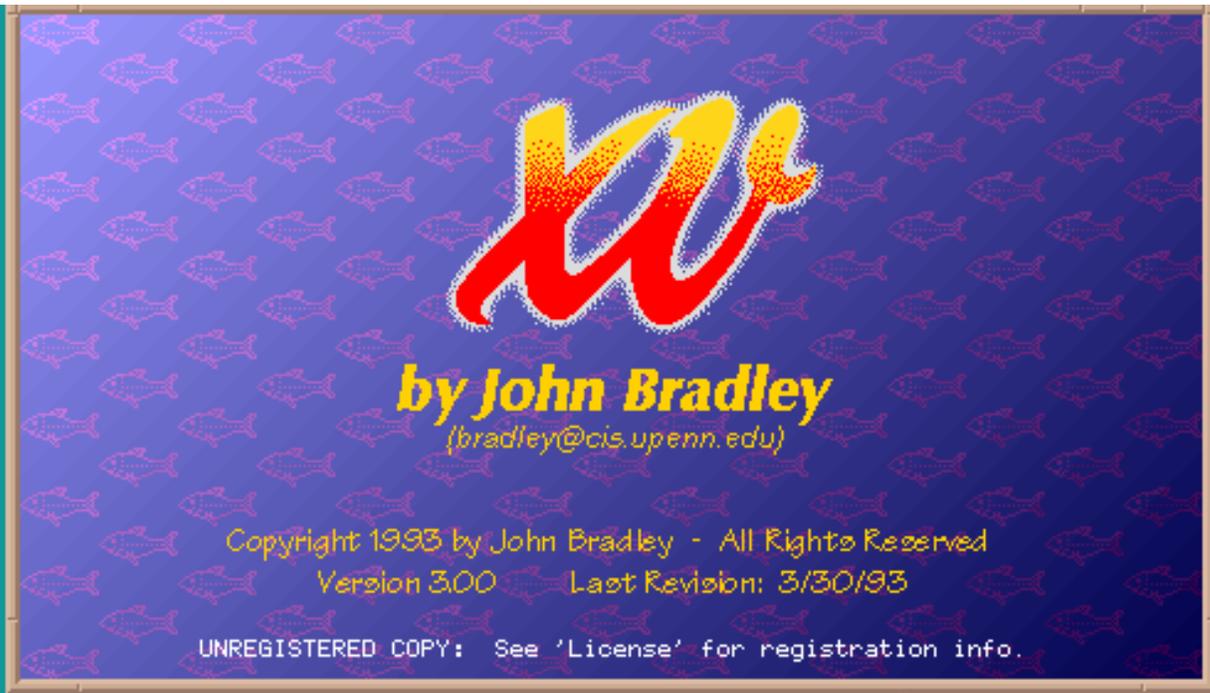


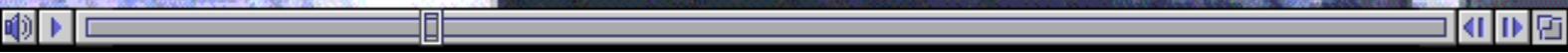
..  
EXAMPLE1.IMG  
EXAMPLE2.GIF  
EXAMPLE3.GIF  
GWS.EXE  
GWSINSTL.EXE  
JULIE.GIF  
JULIEC.GIF

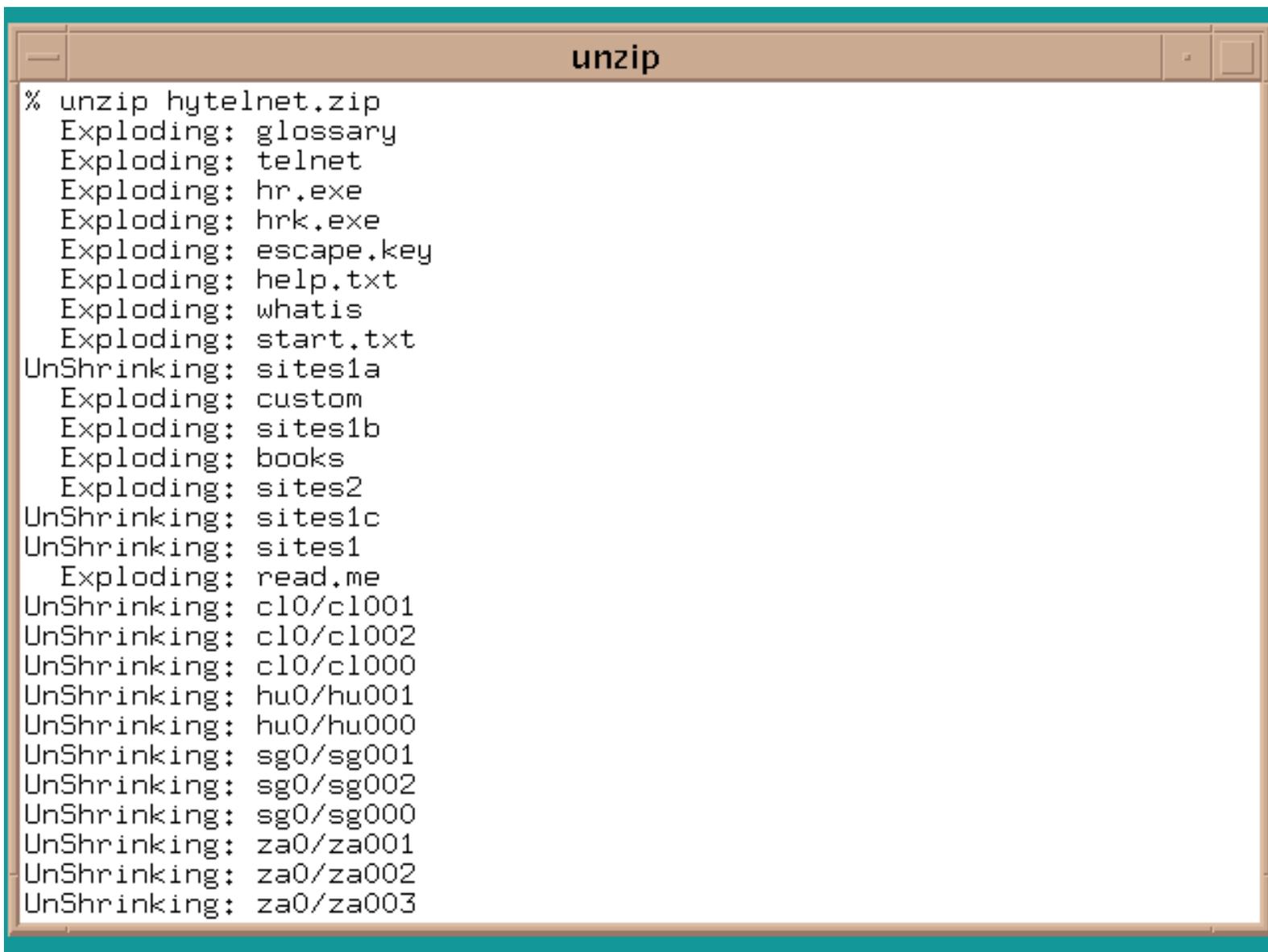
[ Write to ]

ART (PFS:1st Publisher)  
BMP (Windows 3)  
CUT (Halo)  
EPS (PostScript)  
EXE (Self displaying)  
GEM/IMG (Ventura)  
GIF (CompuServe)  
HRZ (SSTU) ♦  
JPG (JFIF JPEG)  
IFF/LBM (Amiga)  
MAC (MacPaint)  
MSP (Microsoft Paint)  
PIC (PC Paint/Pictor)  
PCX (PC Paintbrush)  
RAS (Sun Raster)  
RLE (Windows 3)  
TGA (Truevision Targa)  
TIFF  
TXT (Text files)  
WPG (WordPerfect)









MIME

PINE 3.88 MESSAGE TEXT Folder: INBOX Message 25 of 25 100%

Date: Fri, 01 Jul 1994 11:57:57 -0600 (CST)

From: Earl Fogel <fogel@herald.usask.ca>

To: Earl.Fogel@sask.usask.ca

Subject: Here's that picture you asked for

Parts/attachments:

|   |       |         |                          |
|---|-------|---------|--------------------------|
| 1 | Shown | 3 lines | Text                     |
| 2 | OK    | 16 KB   | Image, "Picture of Earl" |

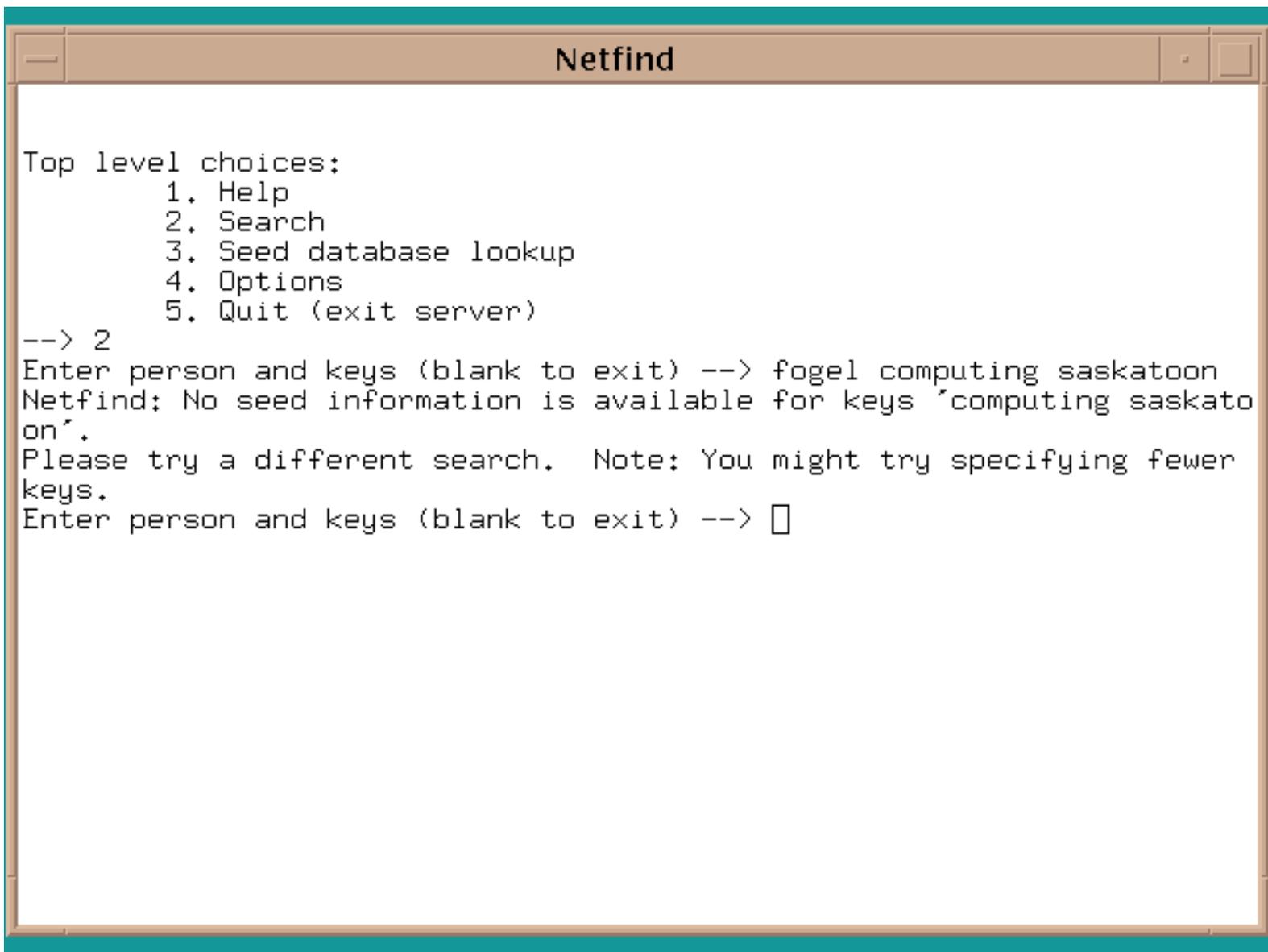
-----  
It's a GIF file. I hope your mail program can handle MIME attachments.

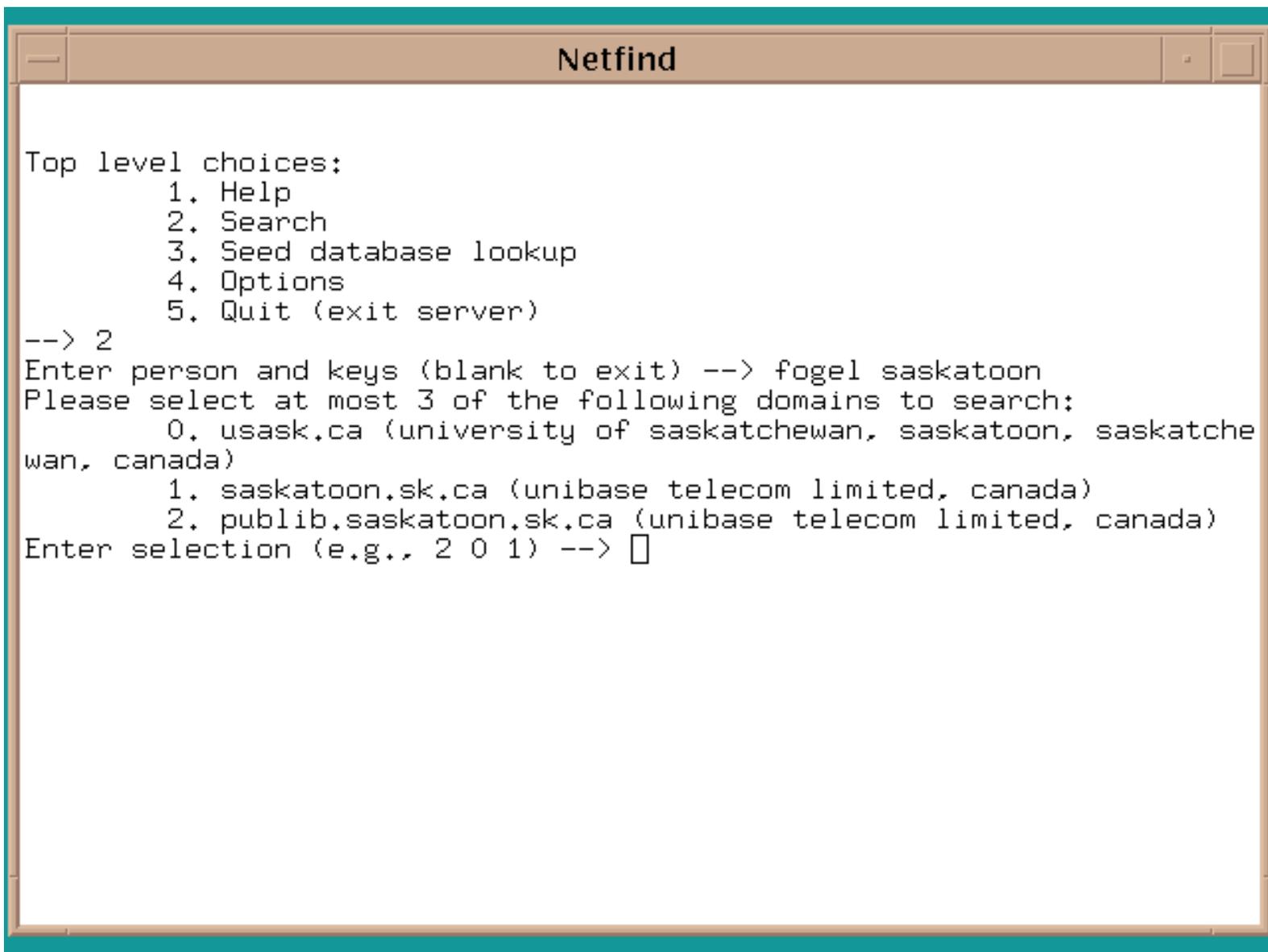
Earl  
[Part 2, "Picture of Earl" Image/GIF 16KB]  
[Not Shown. Use the "V" command to view or save this part]

Enter attachment number to view or save (1 - 2) : 2

^G Help            ^C Cancel

RETURN Enter





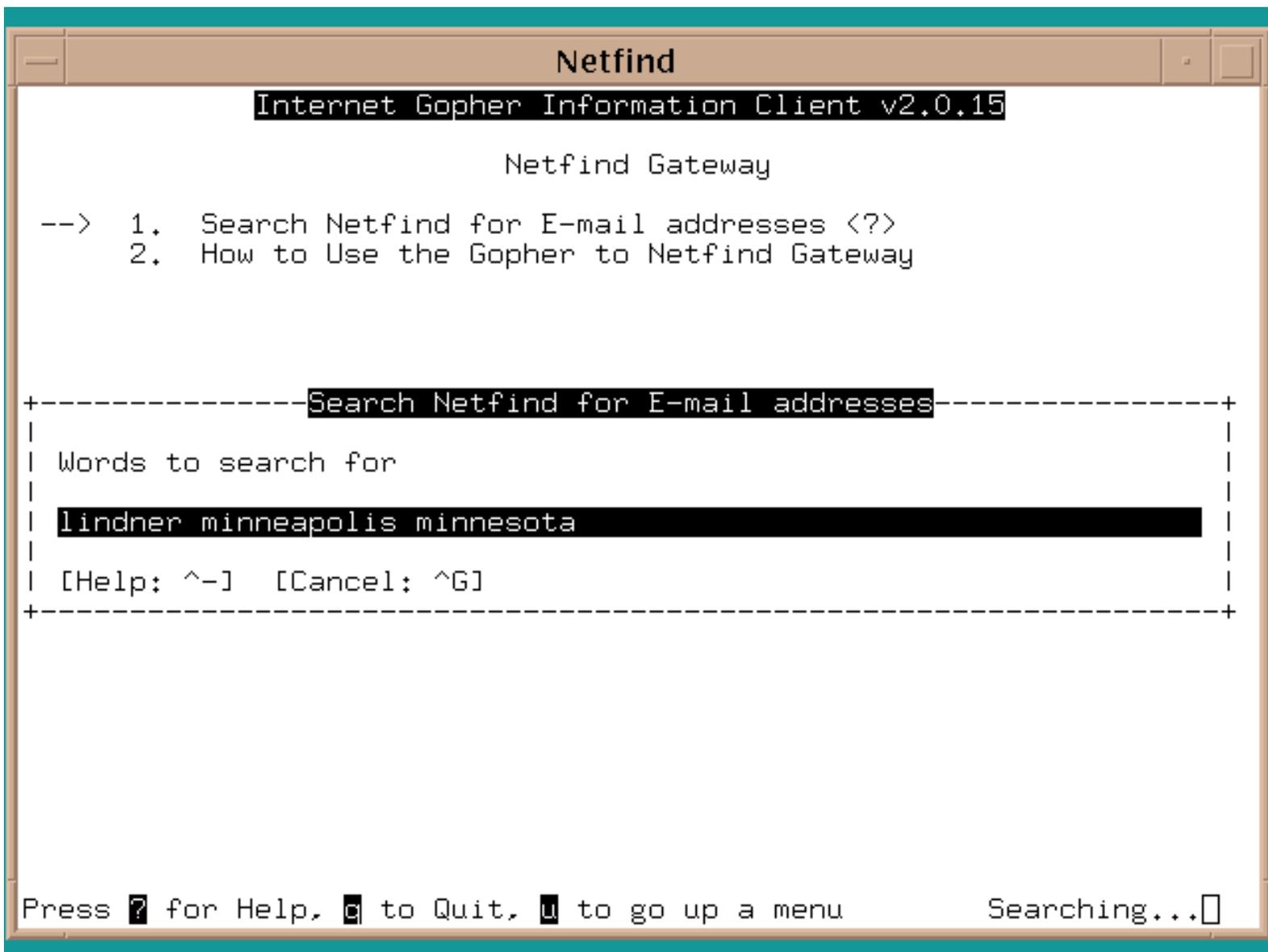
## Netfind

```
Enter person and keys (blank to exit) -->fogel saskatoon saskatchewan
Searching usask.ca
( 1) check_name: checking domain usask.ca. Level = 0
( 1) get_domain_addr: Got nameserver access.usask.ca
( 1) get_domain_addr: Got nameserver dvinci.usask.ca
( 2) check_name: checking nameserver access.usask.ca. Level = 2
( 3) check_name: checking nameserver dvinci.usask.ca. Level = 2

MAIL FOR Earl Fogel IS FORWARDED TO fogel@herald.usask.ca
NOTE: this is a domain mail forwarding arrangement - so mail
      intended for "fogel" should be addressed to "fogel@usask.ca"
      rather than "fogel@herald.usask.ca".

( 3) check_name: checking host herald.usask.ca. Level = 2
( 2) read timed out
( 3) read timed out
SYSTEM: dvinci.usask.ca
      Login name: fogel           In real life: Earl Fogel
      Directory: /home/fogel     Shell: /usr/local/bin/local_sh
      Last login Mon Jun 27 14:29 on ttytc from decent.usask.ca
```

...



Netfind

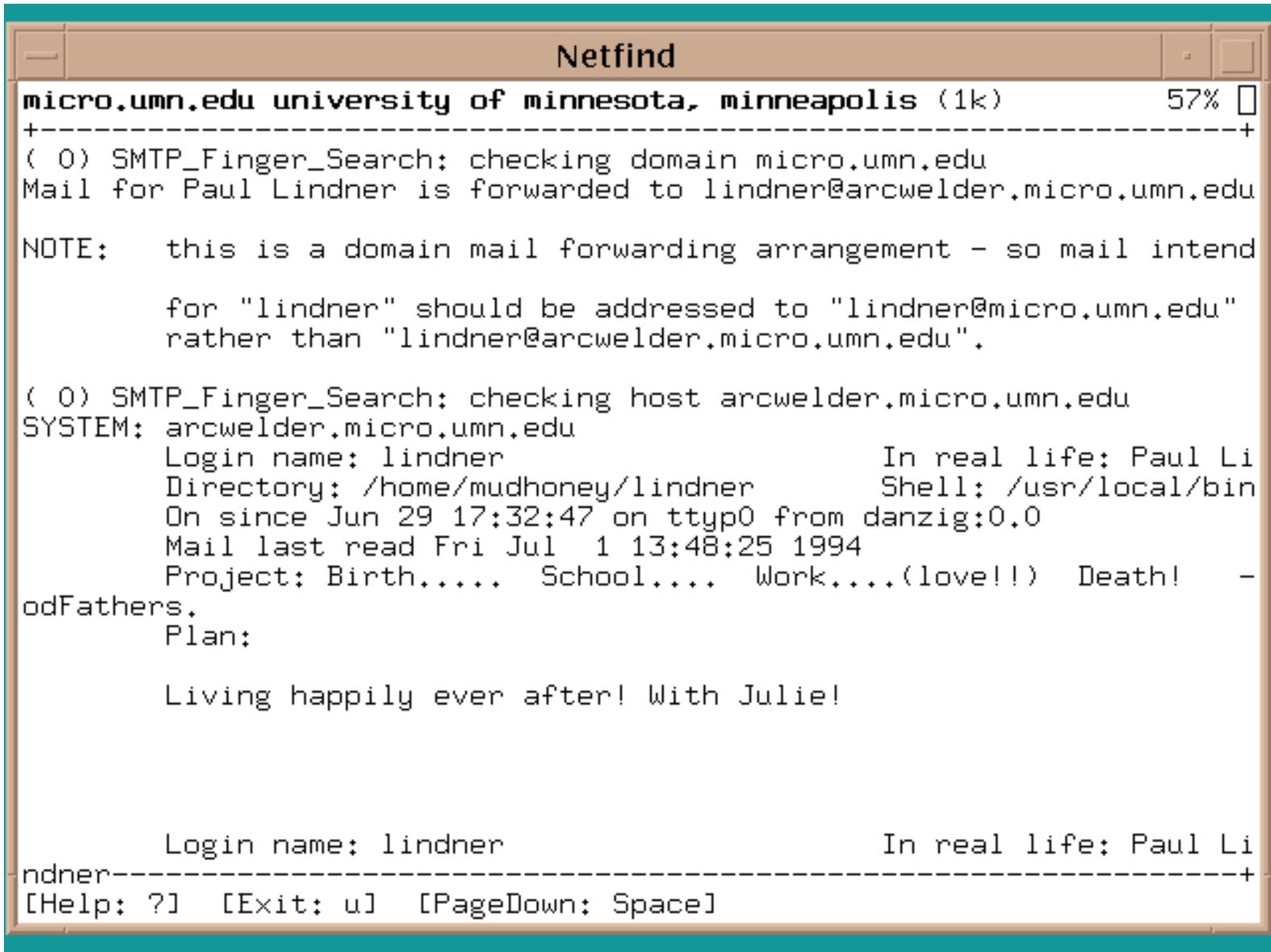
Internet Gopher Information Client v2.0.15

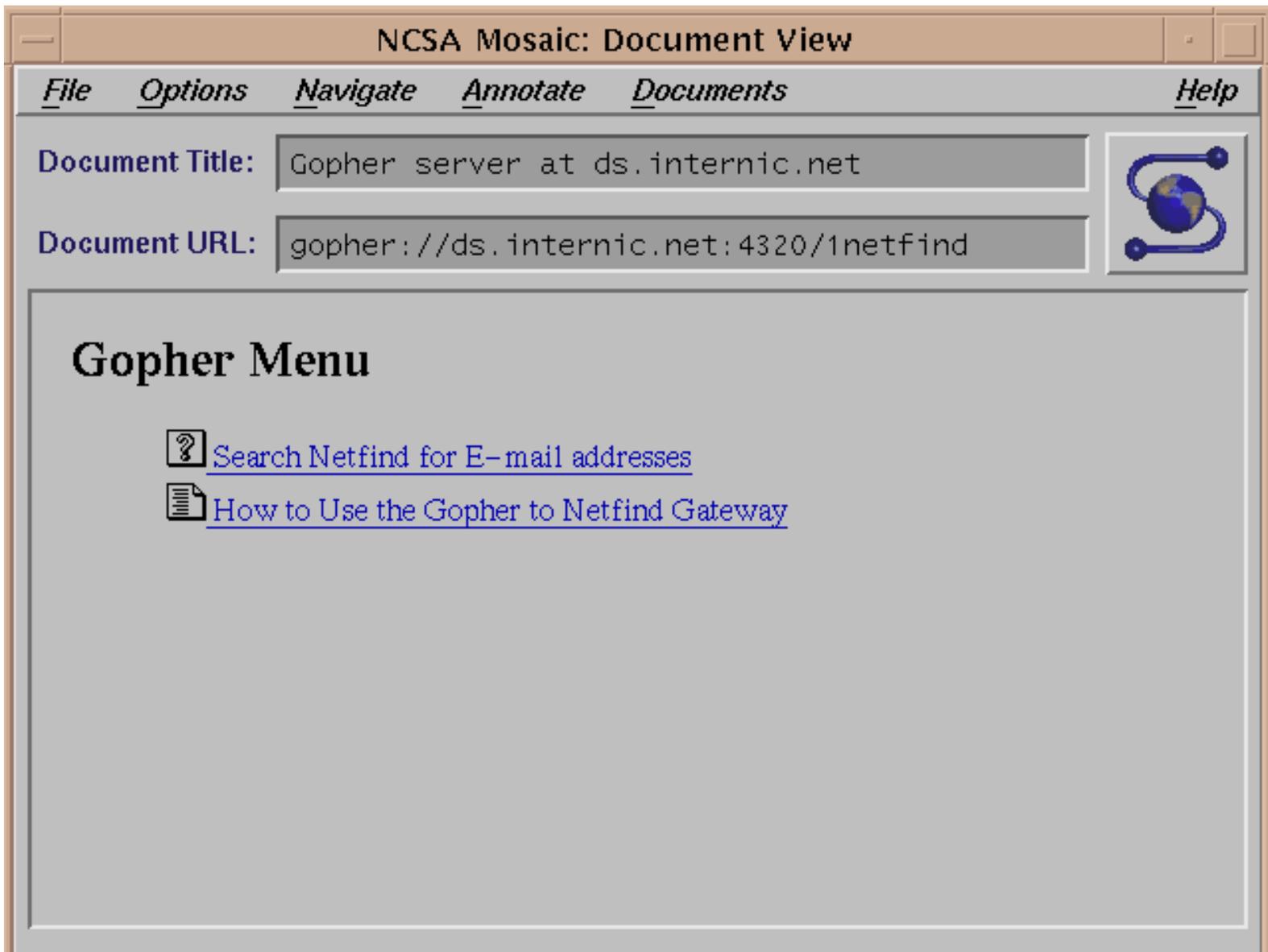
Search Netfind for E-mail addresses: lindner minneapolis minnesota

- 133. mamtc.org minnesota advanced manufacturing technology ce..
- 134. manna-machine.com manna-machine, minneapolis, minnesota
- 135. mapp.org mappcor, minneapolis, minnesota
- 136. math.umn.edu mathematics department, university of minne..
- 137. mcad.edu minneapolis college of art and design, minnesot..
- 138. mdh.umn.edu university of minnesota, minneapolis
- 139. me.umn.edu mechanical engineering department, university..
- 140. med.openpn.umn.edu medical school, university of minneso..
- 141. med.umn.edu medical school, university of minnesota, min..
- 142. medc.umn.edu university of minnesota, minneapolis
- 143. mes.umn.edu university of minnesota, minneapolis
- 144. mhec.umn.edu university of minnesota, minneapolis
- 145. mia.pres.umn.edu university of minnesota, minneapolis
- > 146. micro.umn.edu university of minnesota, minneapolis
- 147. minnetech.mn.org open systems architects, inc, minneapol..
- 148. mishima.mn.org a.n.n. systems, minneapolis, minnesota
- 149. mmf.umn.edu university of minnesota, minneapolis
- 150. mndly.umn.edu university of minnesota, minneapolis
- 151. mnfound.umn.edu university of minnesota, minneapolis
- 152. mntech.org minnesota technology, inc, minneapolis, minne..
- 153. mpl.mcd.mot.com microcomputer division, motorola, inc, m..
- 154. mpis.pres.umn.edu university of minnesota, minneapolis

Press **?** for Help, **Q** to Quit, **U** to go up a menu

Page: 7/12





## NCSA Mosaic: Document View

File Options Navigate Annotate Documents Help

Document Title: Gopher server at ds.internic.net

Document URL: gopher://ds.internic.net:4320/7netfind%20db1



### Gopher Menu

-  [cern.ch.kfki.hu](http://cern.ch.kfki.hu) central research institute for physics, budapest, hungary
-  [cern.ch](http://cern.ch) european laboratory for particle physics, geneva, switzerland
-  [cernvm.cern.ch](http://cernvm.cern.ch) via nazario sauro, cagliari, italy
-  [decnet.cern.ch](http://decnet.cern.ch) organisation (formerly conseil) europeenne pour la recherche nucleaires (european center for nuclear research), geneva, switzerland
-  [dxcoms.cern.ch](http://dxcoms.cern.ch) cern, european laboratory for particle physics, geneva switzerland, cn division, geneva, switzerland
-  [dxmint.cern.ch](http://dxmint.cern.ch) organisation (formerly conseil) europeenne pour la recherche nucleaires (european center for nuclear research), geneva, switzerland

# NCSA Mosaic: Document View

File   Options   Navigate   Annotate   Documents   Help

**Document Title:** cern.ch european laboratory for particle phy



**Document URL:** gopher://ds.internic.net:4320/0netfind%20net

```
( 1) SMTP_Finger_Search: checking host apol3mit4.cern.ch
( 4) connect timed out
( 4) SMTP_Finger_Search: checking host wsde97.cern.ch
( 4) get_host_addr: gethostbyname for wsde97.cern.ch failed
( 4) SMTP_Finger_Search: checking host dxal02.cern.ch
( 2) connect timed out
( 2) SMTP_Finger_Search: checking host vsoph8.cern.ch
( 3) connect timed out
( 3) SMTP_Finger_Search: checking host online.cern.ch
( 1) connect timed out
( 1) SMTP_Finger_Search: checking host oahu.cern.ch
( 0) connect timed out
( 2) connect timed out
( 3) connect timed out
( 4) connect timed out
( 1) connect timed out
```

NOTE: Received no responses, and some host or network failures occurred. Maybe try this search again later.

## Netfind

Top level choices:

1. Help
2. Search
3. Seed database lookup
4. Options
5. Quit (exit server)

--> 2

Enter person and keys (blank to exit) --> anyone zimbabwe

Searching uz.zw

( 1) check\_name: checking domain uz.zw. Level = 0

The domain 'uz.zw' does not run its own name servers,  
and there is no aliased domain IP address/CNAME/MX record for  
this domain -> Skipping domain search phase for this domain.

-----  
Search of domains completed. Proceeding to search of hosts.  
-----

( 1) check\_name: checking host zimbix.uz.zw. Level = 0

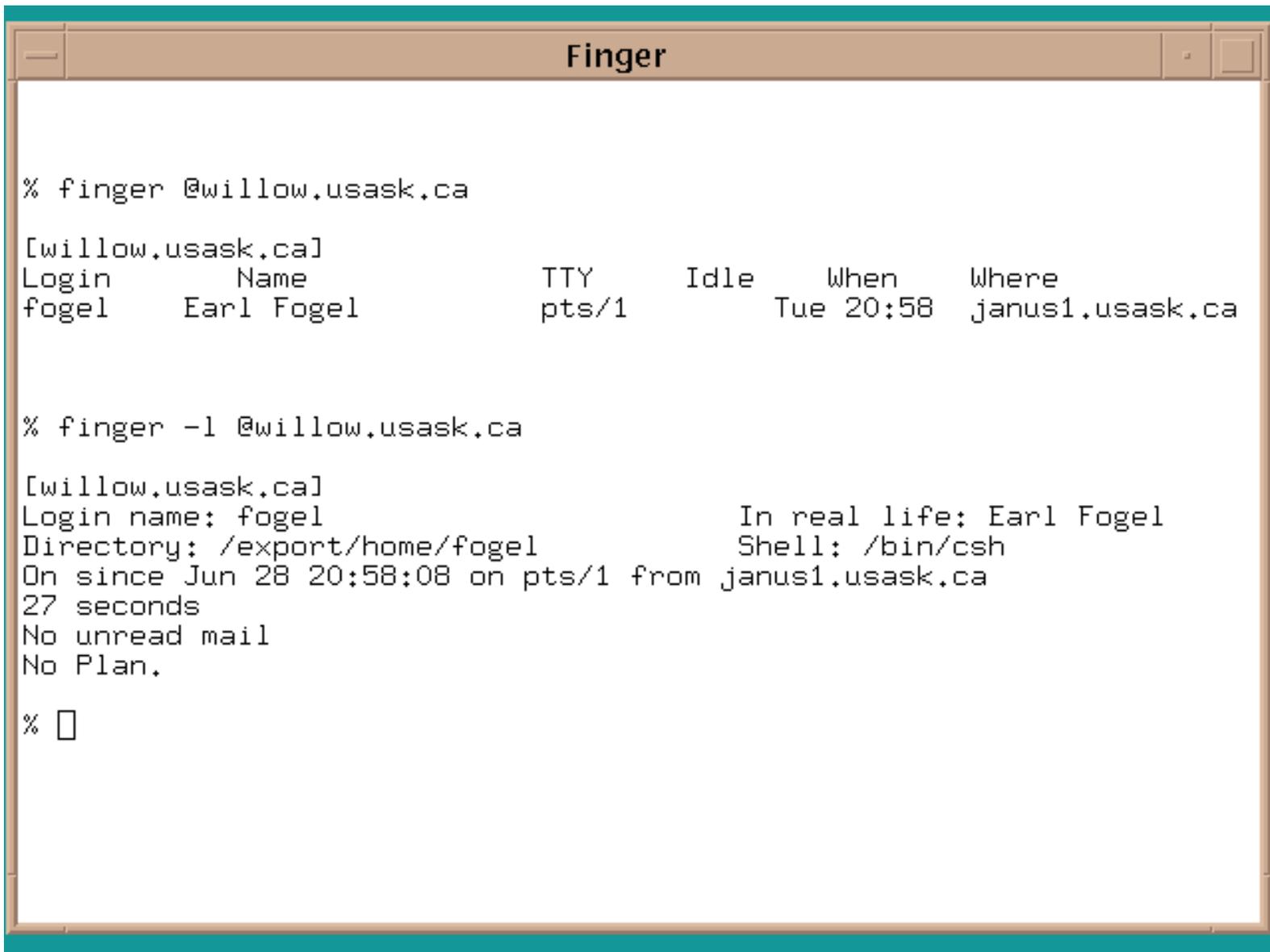
( 1) get\_host\_addr: gethostbyname for zimbix.uz.zw failed

SUMMARY:

- Found no address records for the domain 'uz.zw',  
indicating it is probably not directly connected to the Internet.  
Netfind can only locate users at directly connected sites.

Enter person and keys (blank to exit) -->





## Finger

```
% telnet willow.usask.ca 79
Trying 128.233.74.5...
Connected to willow.usask.ca.
Escape character is '^]'.
fogel
```

| Login | Name       | TTY   | Idle | When      | Where           |
|-------|------------|-------|------|-----------|-----------------|
| fogel | Earl Fogel | pts/1 | 1    | Tue 20:58 | janus1.usask.ca |

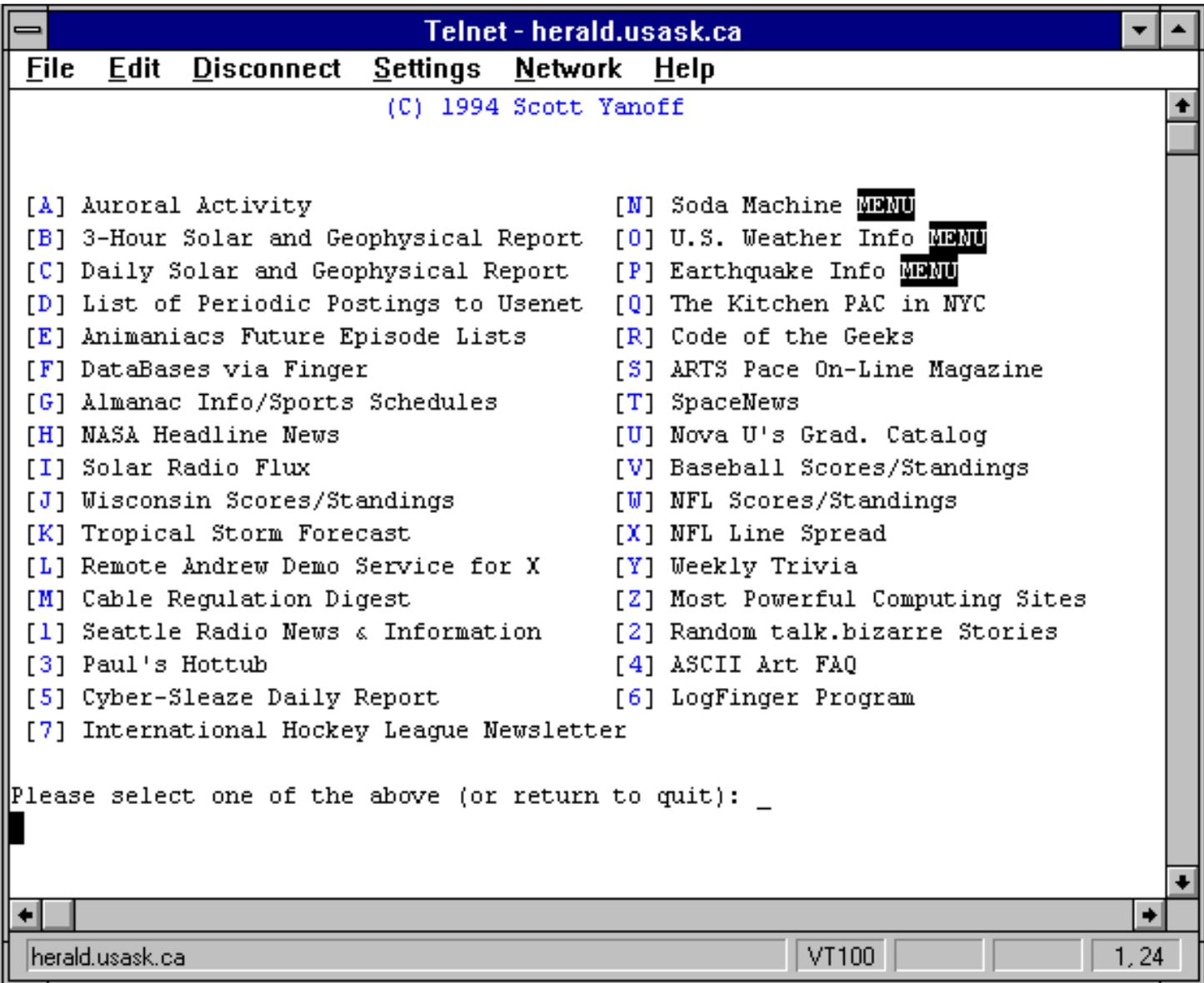
```
Connection closed by foreign host.
```

```
% telnet willow.usask.ca 79
Trying 128.233.74.5...
Connected to willow.usask.ca.
Escape character is '^]'.
/wfogel
```

```
Login name: fogel                In real life: Earl Fogel
Directory: /export/home/fogel    Shell: /bin/csh
Last login Fri Jul  1 09:43 on pts/1 from herald.usask.ca
No unread mail
No Plan.
```

```
Connection closed by foreign host.
```

```
% □
```



NCSA Mosaic: Document View

File Options Navigate Annotate Documents Help

Document Title: Special Internet Connections: Last update 6/

Document URL: ftp://csd4.csd.uwm.edu/pub/inet.services.htm



---

## GEOPHYSICAL/GEOGRAPHICAL/GEOLOGICAL

---

### Earthquake Info.

Recent quake info (location, time, magnitude, etc.).

- [finger\\_quake@gldfs.cr.usgs.gov](mailto:finger_quake@gldfs.cr.usgs.gov)
- Washington, Oregon: [finger\\_quake@geophys.washington.edu](mailto:finger_quake@geophys.washington.edu)  
[128.95.16.50]
- Alaska: [finger\\_quake@fm.gi.alaska.edu](mailto:finger_quake@fm.gi.alaska.edu)
- Southern California: [finger\\_quake@scec2.gps.caltech.edu](mailto:finger_quake@scec2.gps.caltech.edu)
- Southern Illinois: [finger\\_quake@slueas.slu.edu](mailto:finger_quake@slueas.slu.edu)
- Nevada, California: [finger\\_quake@seismo.unr.edu](mailto:finger_quake@seismo.unr.edu)
- Northern California: [finger\\_quake@andreas.wr.usgs.gov](mailto:finger_quake@andreas.wr.usgs.gov)
- Utah, Wyoming, Montana: [finger\\_quake@eqinfo.seis.utah.edu](mailto:finger_quake@eqinfo.seis.utah.edu)
- [talbot\\_geophys.washington.edu](mailto:talbot_geophys.washington.edu) (Login/Password: user:ls)

## Finger

```
% finger fogel
Login name: fogel                      In real life: Earl Fogel
Directory: /export/home/fogel         Shell: /bin/csh
On since Jul  1 13:41:10 on pts/1 from duke.usask.ca
No unread mail
Plan:
1 person has fingered me since 01:41:43 PM on Friday July 01, 1994

% finger fogel
Login name: fogel                      In real life: Earl Fogel
Directory: /export/home/fogel         Shell: /bin/csh
On since Jul  1 13:41:10 on pts/1 from duke.usask.ca
No unread mail
Plan:
2 people have fingered me since 01:41:43 PM on Friday July 01, 1994

% finger fogel
Login name: fogel                      In real life: Earl Fogel
Directory: /export/home/fogel         Shell: /bin/csh
On since Jul  1 13:41:10 on pts/1 from duke.usask.ca
No unread mail
Plan:
3 people have fingered me since 01:41:43 PM on Friday July 01, 1994
□
```

NCSA Mosaic: Document View

File Options Navigate Annotate Documents Help

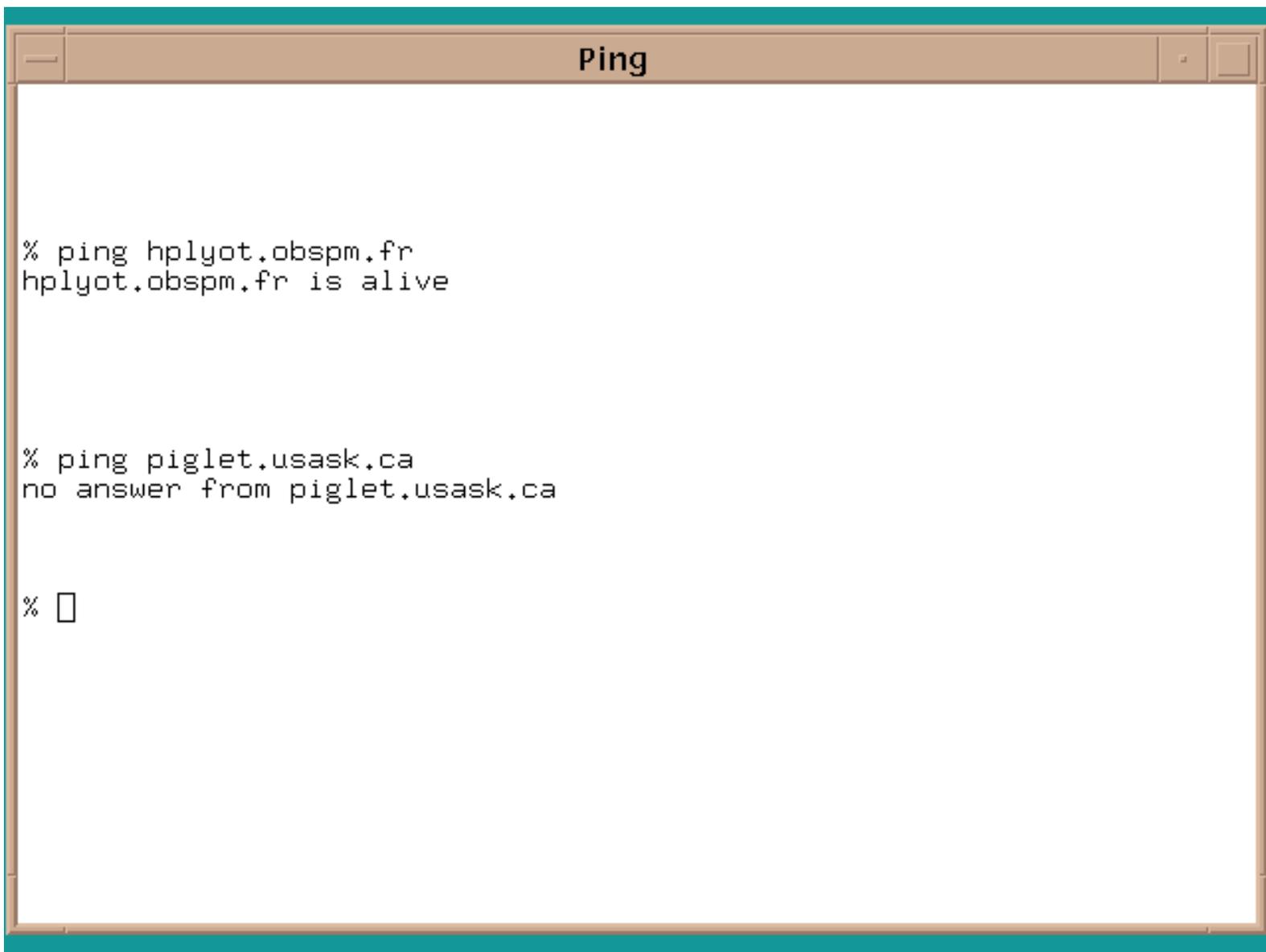
Document Title: Finger @herald.usask.ca

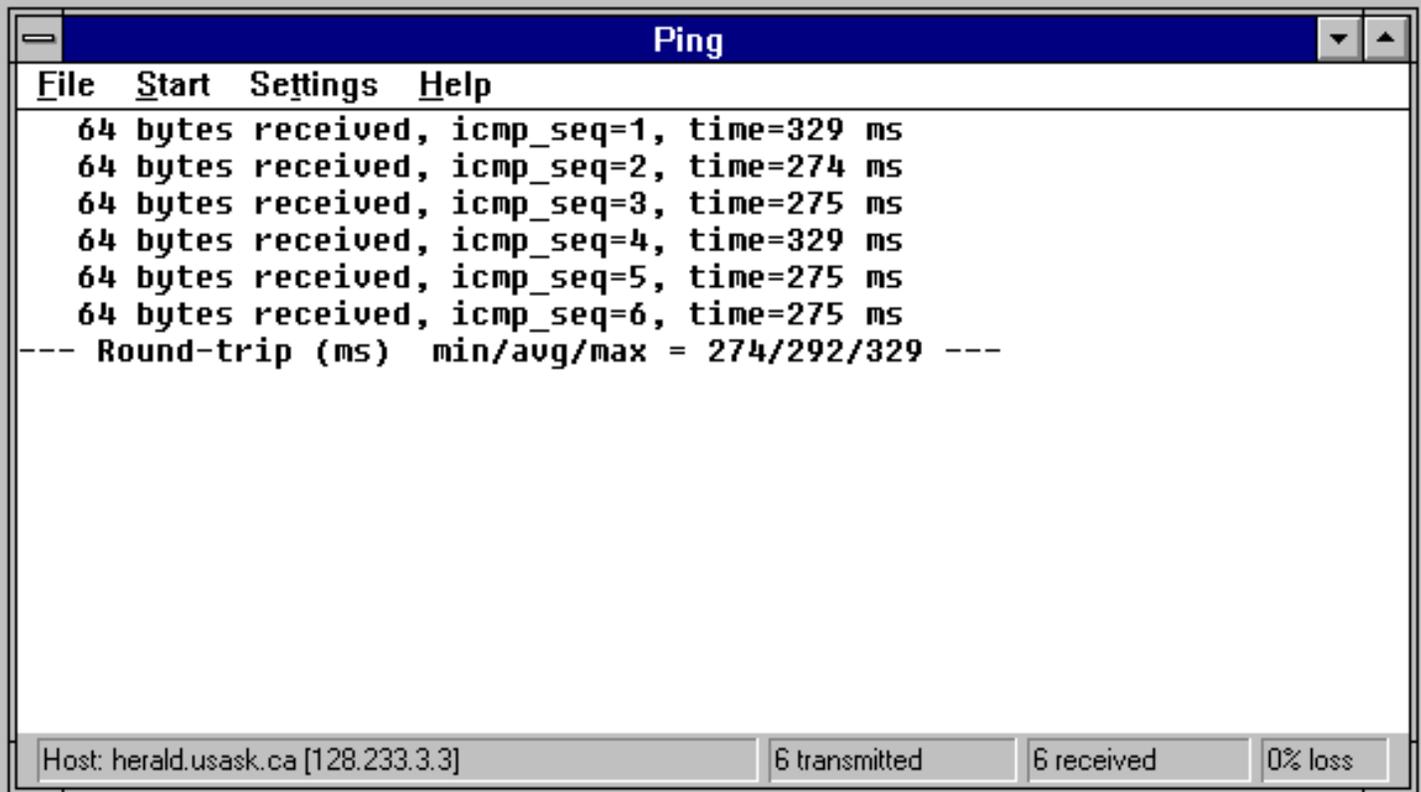
Document URL: http://www.mit.edu:8001/finger



# Finger @herald.usask.ca

| <i>Login</i>             | <i>Name</i>          | <i>TTY</i> | <i>Idle</i> | <i>When</i> | <i>Offic</i>  |
|--------------------------|----------------------|------------|-------------|-------------|---------------|
| <a href="#">dot130</a>   | Don Tse              | 03         |             | Fri 09:43   |               |
| <a href="#">stevens</a>  | Debra Stevens        | 04         | 19:         | Wed 14:14   |               |
| <a href="#">prabhuv</a>  | Vikram Prabhu        | 05         |             | Fri 12:54   |               |
| <a href="#">hlady</a>    | Jason Hlady          | 06         |             | Fri 12:12   |               |
| <a href="#">gravlinp</a> | Peter Gravlin        | 07         |             | Fri 12:21   |               |
| <a href="#">rac128</a>   | Raymond Chin         | 09         |             | Fri 12:32   |               |
| <a href="#">pmb124</a>   | Pamela Marie Burnham | 10         |             | Fri 12:41   |               |
| <a href="#">scottp</a>   | Peter Scott          | 12         |             | Fri 12:46   | Library-Sys x |
| <a href="#">bidwellp</a> | Paul Bidwell         | p0         | 1:48        | Fri 11:10   |               |
| <a href="#">fogel</a>    | Earl Fogel           | p1         |             | Fri 12:54   | 61 Physics x  |
| <a href="#">goldie</a>   | Hugh Goldie          | p2         |             | Fri 12:42   |               |
| <a href="#">mcleod</a>   | Stephen McLeod       | p5         | 1d          | Wed 13:23   | Arts Comput 9 |





Program  
Manager



NEWT

