

Open Source Software for z/OS and OS/390 UNIX

Freeware source and binaries for the mainframe!

Obtain, build and enhance open source software packages

Detailed descriptions of the most popular programs



Michael MacIsaac
Sándor Bárány
Thorsten Brockmeier
Guillermo Freige
Stefan Koesling

Kiran Madnani, Fulvio Malfatto
Ralf Schandl, Egon Terwedow
James Tison, Jeff Trawick

ibm.com/redbooks

Redbooks



International Technical Support Organization

**Open Source
Software
for z/OS and OS/390 UNIX**

March 2002

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix E, "Special notices" on page 307.

Second Edition (March 2002)

This edition applies to version 2 releases 7, 8 and 9 of OS/390, Program Number 5647-A01.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000, 2002. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	xvii
The team that wrote this redbook	xix
Comments welcome	xxii
 Chapter 1. Introduction	1
1.1 Terminology	1
1.2 A brief history of open source software	2
1.3 The development model of open source software	7
1.4 OS/390 resources	9
1.4.1 News group	10
1.4.2 Web sites	10
1.4.3 An OS/390 TSO resource	11
 Chapter 2. Licensing software	13
2.1 The GNU General Public License (GPL)	13
2.2 Other licenses	14
2.3 The Open Source Definition	14
2.4 The IBM Public License	15
 Chapter 3. What's on the CD-ROM	17
3.1 Package installation ordering	17
3.2 Installing the packages from source	18
3.2.1 Source distribution installation	18
3.2.2 Installing quickly	19
3.2.3 Details on installing	19
3.3 Installing binaries	22
3.4 Installation size	22
3.5 Installation directory	23
3.6 Setting the PATH and MANPATH environment variables	23
 Chapter 4. Open source software packages	25
4.1 What you find in open source packages	26
4.2 Overall source structure	27
4.3 Configuring and compiling	28
4.3.1 The configuration process	28
4.3.2 Modifying the configure script	31
4.4 Troubleshooting, reporting problems, and bugs	31
4.5 Improving package performance on OS/390	32
4.6 Modifying the source code	32
4.7 Sharing your work	32

Chapter 5. Apache	33
5.1 Introduction to Apache	33
5.2 Installation of Apache	33
5.2.1 Step 1: Load the product	34
5.2.2 Step 2: Configure the product	34
5.2.3 Step 3: Compile and link the product	35
5.2.4 Step 4: Install the product	36
5.2.5 Heart of the Apache.install script	37
5.3 Customization and operation of Apache	37
5.3.1 Modifying the server configuration conf/httpd.conf file	37
5.3.2 RACF setup for the Apache server	37
5.3.3 Starting the Apache server	38
5.3.4 Stopping the Apache server	39
5.3.5 Running CGI programs	39
5.4 License	41
5.5 Getting Apache from the Internet	42
5.6 Porting Apache to OS/390	42
Chapter 6. Autoconf	43
6.1 Introduction to autoconf	43
6.2 Installation of autoconf	43
6.3 Customization and operation of autoconf	44
6.4 License	45
6.5 Getting autoconf from the Internet	46
6.6 Porting autoconf to OS/390	46
6.7 Caveats	47
Chapter 7. Automake	49
7.1 Introduction to automake	49
7.2 Installation of automake	49
7.3 Customization and operation of automake	50
7.4 License	51
7.5 Getting automake from the Internet	51
7.6 Porting automake to OS/390	51
Chapter 8. Bash	53
8.1 Introduction to bash	53
8.2 Installation of bash	53
8.3 Customization and operation of bash	54
8.4 License	54
8.5 Getting bash from the Internet	54
8.6 Porting bash to OS/390	54

Chapter 9. CVS	57
9.1 Introduction to CVS	57
9.2 Installation of CVS	57
9.2.1 Core of the cvs.install script	57
9.2.2 Limitations of CVS on OS/390	58
9.3 Customization and operation of CVS	58
9.4 License	58
9.5 Getting CVS from the Internet	59
9.6 Porting CVS to OS/390	59
9.6.1 Future work needed on the OS/390 port of CVS	59
9.7 Other resources	60
Chapter 10. Diffutils	61
10.1 Introduction to diffutils	61
10.2 Installation of diffutils	61
10.3 Customization and operation of diffutils	62
10.4 License	62
10.5 Getting diffutils from the Internet	63
10.6 Porting diffutils to OS/390	63
Chapter 11. Emacs	65
11.1 Introduction to emacs	65
11.2 Installation of emacs	65
11.2.1 Step 1: Load the software	66
11.2.2 Step 2: Configure and build emacs	66
11.2.3 Troubleshooting	66
11.3 Customization and operation of emacs	67
11.4 License	67
11.5 Getting emacs from the Internet	67
11.6 Porting emacs to OS/390	67
Chapter 12. Fgres	69
12.1 Introduction to fgres	69
12.2 Installation of fgres	69
12.3 Example of using fgres	69
12.4 License	69
12.5 Getting fgres from the Internet	70
12.6 Porting fgres to OS/390	70
Chapter 13. Findutils	71
13.1 Introduction to findutils	71
13.2 Installation of findutils	71
13.3 Customization and operation of findutils	72
13.4 License	73

13.5	Getting findutils from the Internet	73
13.6	Porting findutils to OS/390	73
Chapter 14.	Flex and Bison	75
14.1	Introduction to flex and bison	75
14.2	Installation of flex and bison	75
14.2.1	Installation of bison	76
14.2.2	Installation of flex	76
14.3	Customization and operation of flex and bison	76
14.4	License	77
14.5	Getting flex and bison from the Internet	77
14.6	Porting flex and bison to OS/390	78
14.6.1	An issue with bison	78
14.6.2	An issue with flex	79
Chapter 15.	Gd	81
15.1	Introduction to gd	81
15.2	Installation of gd	81
15.3	Customization and operation of gd	81
15.4	License	83
15.5	Getting gd from the Internet	83
15.6	Porting gd to OS/390	84
Chapter 16.	Gmake	85
16.1	Introduction to gmake	85
16.2	Installation of gmake	85
16.3	Customization and operation of gmake	86
16.4	License	86
16.5	Getting gmake from the Internet	86
16.6	Porting gmake to OS/390	87
Chapter 17.	Gnats	89
17.1	Introduction to gnats	89
17.2	Installation of gnats	90
17.3	Customizing and operating gnats	91
17.4	The gnats licence	91
17.5	Getting gnats from the Internet	91
17.6	Porting gnats to OS/390	92
Chapter 18.	GNU grep	93
18.1	Introduction	93
18.2	Installation of GNU grep	93
18.3	Using grep	93
18.3.1	Examples	94

18.4 The GNU grep license	95
18.5 Getting gnu grep from the Internet	95
18.6 Porting gnu grep to OS/390	95
Chapter 19. Groff	97
19.1 Introduction to groff	97
19.2 History	98
19.3 Installation of groff	98
19.4 Customizing and operation of groff.	98
19.5 License	99
19.6 Getting groff from the Internet	99
19.7 Porting groff to OS/390.	99
19.7.1 A configure problem	99
19.7.2 Minor problems.	100
19.7.3 Font troubleshooting.	101
Chapter 20. Gzip	105
20.1 Introduction to gzip	105
20.2 Installation of gzip	105
20.3 Customization and operation of gzip.	106
20.4 License	107
20.5 Getting gzip from the Internet.	107
20.6 Porting gzip to OS/390	108
Chapter 21. Id-utils	109
21.1 Introduction to id-utils	109
21.2 Installation of id-utils.	109
21.3 Customization and operation of id-utils.	110
21.4 License	110
21.5 Getting id-utils from the Internet	110
21.6 Porting id-utils to OS/390	110
Chapter 22. Info-ZIP	113
22.1 Introduction to Info-ZIP.	113
22.2 Installation of Info-ZIP	113
22.3 Customization and operation of Info-ZIP.	113
22.4 License	114
22.5 Getting Info-ZIP from the Internet.	115
22.6 Porting Info-ZIP to OS/390	115
22.6.1 Changes in the zip package	115
22.6.2 Changes in the unzip package	115
Chapter 23. Itools	117
23.1 Introduction to itools	117

23.1.1	Imake versus automake/autoconf	117
23.1.2	Imake and the X Window system	118
23.1.3	Vanilla versus 'extensible architecture' imake	118
23.1.4	Imake versus itools	119
23.1.5	An example of imake	119
23.2	Installing itools	119
23.3	Customization and operation of itools components	121
23.4	License	121
23.5	Getting itools from the Internet	122
23.6	Porting imake to OS/390	122
23.7	Caveats	124
Chapter 24.	Jpeg library	125
24.1	Introduction to the jpeg library	125
24.2	Installation of the jpeg library	125
24.3	Usage of the jpeg library	126
24.4	License	126
24.5	Getting the jpeg library from the Internet	127
24.6	Porting the jpeg library to OS/390	127
Chapter 25.	Korn shell	129
25.1	Introduction to the Korn shell	129
25.2	Installation of the Korn shell	129
25.3	Customization and operation of the Korn shell	130
25.4	License	130
25.5	Getting the Korn shell from the Internet	130
Chapter 26.	Less	131
26.1	Introduction to less	131
26.2	Installation of less	133
26.3	Customizing and operation of less	133
26.4	Licence for less	134
26.5	Getting less from the Internet	135
26.6	Porting less to OS/390	135
Chapter 27.	Libpng library	137
27.1	Introduction to the libpng library	137
27.2	Installation of the libpng library	138
27.3	Usage of the libpng library	138
27.4	License	138
27.5	Getting the libpng library from the Internet	140
27.6	Porting the libpng library to OS/390	140

Chapter 28. Libtool	143
28.1 Introduction to libtool	143
28.2 Installation of libtool	143
28.3 License	144
28.4 Getting libtool from the Internet	144
Chapter 29. Lynx	145
29.1 Introduction to Lynx	145
29.2 Installation of Lynx	145
29.3 Customization and operation of Lynx	146
29.4 License	146
29.5 Getting Lynx from the Internet	146
29.6 Porting Lynx to OS/390	146
Chapter 30. M4	147
30.1 Introduction to m4	147
30.2 Installation of m4	147
30.3 Customization and operation of m4	147
30.4 License	147
30.5 Getting m4 from the Internet	148
30.6 Porting m4 to OS/390	148
Chapter 31. MySQL	151
31.1 Introduction to MySQL	151
31.2 Installation of MySQL	151
31.3 Customization and operation of MySQL	152
31.4 License	153
31.5 Getting MySQL from the Internet	153
31.6 Porting MySQL to OS/390	153
31.6.1 State of the port	153
31.6.2 Problems faced	154
31.6.3 Necessary code changes	158
Chapter 32. Nedit	163
32.1 Introduction to nedit	163
32.2 Installing nedit	163
32.3 Customizing and operating nedit	165
32.4 The license for nedit	166
32.5 Getting nedit from the Internet	166
32.6 Porting nedit to OS/390	166
Chapter 33. OpenSSL and OpenSSH	167
33.1 Introduction to OpenSSL and OpenSSH	167
33.2 Installation of OpenSSL and OpenSSH	167

33.2.1	Installation of OpenSSL	167
33.2.2	Installation of OpenSSH	168
33.3	Licenses	168
33.3.1	OpenSSL license	169
33.3.2	OpenSSH license	170
33.4	Getting OpenSSL and OpenSSH from the Internet	172
33.5	Porting OpenSSL and OpenSSH to OS/390	172
33.5.1	Status	173
33.5.2	OpenSSH changes	173
33.5.3	OpenSSL changes	182
33.5.4	Zlib	183
33.5.5	Entropy gathering daemon	184
33.5.6	OpenSSH	184
Chapter 34.	Patch	185
34.1	Introduction to patch	185
34.2	Installation of patch	185
34.3	Customization and operation of patch	186
34.4	License	187
34.5	Getting patch from the Internet	187
34.6	Porting patch to OS/390	187
Chapter 35.	Perl 5.005p3	189
35.1	Introduction to Perl	189
35.2	Installation of Perl	189
35.2.1	Step 1: Prepare your system to install Perl	189
35.2.2	Step 2: Load the software	190
35.2.3	Step 3: Apply patches for configuration files (Perl 5.005.03) ..	190
35.2.4	Step 4: Configure Perl	191
35.2.5	Step 5: Build Perl	191
35.2.6	Step 6: Installation verification process (IVP)	192
35.2.7	Step 7: Copy Perl into the runtime directory	192
35.3	Customization and operation of Perl	192
35.3.1	Making Perl accessible from the Internet	193
35.3.2	How to call Perl on OS/390	193
35.3.3	Get an example Perl CGI script	193
35.3.4	Accessing MVS datasets from Perl	193
35.3.5	Useful information and Internet user groups	194
35.3.6	Current limitations on Perl for OS/390	194
35.4	License	195
35.5	Getting Perl from the Internet	195
35.6	Porting Perl to OS/390	196

Chapter 36. Perl 5.6.1	197
36.1 Introduction to Perl 5.6.1	197
36.2 Installation of Perl 5.6.1	197
36.3 License	198
36.4 Getting Perl 5.6 from the Internet	198
36.5 Porting Perl 5.6 to OS/390	198
Chapter 37. PHP	199
37.1 Introduction to PHP	199
37.2 Installation of PHP	200
37.3 Customization and operation of PHP	201
37.4 License	201
37.5 Getting PHP from the Internet	203
37.6 Porting PHP to OS/390	203
Chapter 38. RCS	209
38.1 Introduction to RCS	209
38.2 Installation of RCS	209
38.3 Customization and operation of RCS	209
38.4 License	210
38.5 Getting RCS from the Internet	210
38.6 Porting RCS to OS/390	211
Chapter 39. Rlogin client	213
39.1 Introduction to the rlogin client	213
39.2 Installation of the rlogin client	213
39.3 Customization and operation of rlogin client	213
39.4 License	214
39.5 Getting the rlogin client from the Internet	214
39.6 Porting the rlogin client to OS/390	215
Chapter 40. Rsh	217
40.1 Introduction to rsh	217
40.2 Installation of rsh	217
40.3 Customization and operation of rsh	217
40.4 License	218
40.5 Getting rsh from the Internet	219
40.6 Porting rsh to OS/390	219
Chapter 41. Samba	221
41.1 Introduction to Samba	221
41.2 Installation of Samba	221
41.3 Customization and operation of Samba V2	222
41.4 License	222

41.5	Getting Samba V2 from the Internet	223
41.6	Porting Samba V2 to OS/390	223
41.6.1	Porting smbd and smbclient	223
41.6.2	Text character translations	226
41.6.3	Porting SWAT	228
Chapter 42.	Samba 1.9.18p1	231
42.1	Introduction to Samba	231
42.2	Installation of Samba	231
42.3	Customization and operation of Samba	232
42.4	License	232
42.5	Getting Samba from the Internet	232
42.6	Porting Samba to OS/390	232
Chapter 43.	MSNTTP	233
43.1	Introduction to MSNTTP	233
43.2	Installation of sntp	233
43.3	Customization and operation of sntp	233
43.4	License	234
43.5	Getting sntp from the Internet	235
43.6	Porting MSNTTP to OS/390	235
Chapter 44.	Squid	237
44.1	Introduction to Squid	237
44.2	Installation of Squid	237
44.3	Customization and operation of Squid	238
44.4	License	239
44.5	Getting Squid from the Internet	239
44.6	Porting Squid to OS/390	240
Chapter 45.	The Hessling Editor (THE)	243
45.1	Introduction to THE	243
45.2	Installation of THE	244
45.3	Customization and operation of THE	244
45.4	License	246
45.5	Getting THE from the Internet	246
45.6	Porting THE to OS/390	246
Chapter 46.	Vim	249
46.1	Introduction to Vim	249
46.2	Installation of Vim	249
46.2.1	Version 5.6	250
46.3	Customization and operation of Vim	254
46.4	Licence	256

46.5	Getting Vim from the Internet	257
46.6	Porting Vim to OS/390	257
Chapter 47. Webalizer		261
47.1	Introduction to Webalizer	261
47.2	Installation of Webalizer	261
47.3	Customization and operation of Webalizer	262
47.4	License	263
47.5	Getting Webalizer from the Internet	263
47.6	Porting Webalizer to OS/390	263
Chapter 48. Xterm client		265
48.1	Introduction to xterm client	265
48.2	Installation of xterm client	265
48.3	Customization and operation of xterm client	266
48.4	License	266
48.5	Getting the xterm client from the Internet	267
48.6	Porting xterm client to OS/390	267
Chapter 49. Zlib library		269
49.1	Introduction to zlib library	269
49.2	Installation of zlib library	269
49.3	License	269
49.4	Getting zlib library from the Internet	270
49.5	Porting the zlib library	270
Chapter 50. Observations		271
50.1	Hints and tips	271
50.1.1	Reducing executable size	271
50.1.2	Configure and the debugging option	272
50.1.3	Setting aliases for missing commands	272
50.1.4	Setting the DISPLAY variable for X Window applications	273
50.1.5	Login from Linux	273
50.1.6	Downloading an X Window server from the Internet	274
50.1.7	Testing for EBCDIC and OS/390 in configure	274
50.2	Problems encountered	276
50.2.1	Configure problems	276
50.2.2	Using the wrong compiler	276
50.2.3	Order of compiler command line parameters	276
50.2.4	Static constructor test fails	277
50.2.5	Newline denoted by a constant	278
50.2.6	Compilation warning	278
50.2.7	Linker does not find libraries	279
50.2.8	Using make	279

50.2.9 Differences between make and gmake	280
50.2.10 Binder problems in OS/390 V2.9.	280
50.2.11 Codeset for internal strings.	280
50.2.12 Function getenv()	280
50.2.13 Configure fails because of ls alias	281
50.2.14 Problems with setuid(), setgid() or __passwd() functions . . .	281
Chapter 51. Summary.	285
51.1 Summary of packages	285
Appendix A. The GNU General Public License	289
Appendix B. The Open Source Definition	297
B.1 Free redistribution.	297
B.2 Source code	297
B.3 Derived works	297
B.4 Integrity of the author's source code.	297
B.5 No discrimination against persons or groups	298
B.6 No discrimination against fields of endeavor	298
B.7 Distribution of license	298
B.8 License must not be specific to a product.	298
B.9 License must not contaminate other software	298
B.10 Sample licenses	298
Appendix C. The IBM Public License	299
Appendix D. Using the additional material	305
D.1 Using the CD-ROM.	305
D.1.1 System requirements for using the CD-ROM.	305
D.1.2 How to use the CD-ROM	305
D.2 Locating the additional material on the Internet	306
Appendix E. Special notices	307
Appendix F. Related publications	311
F.1 IBM Redbooks publications.	311
F.2 IBM Redbooks collections.	311
F.3 Other resources	311
F.4 Referenced Web sites.	312
How to get IBM Redbooks	315
IBM Redbooks fax order form	316

Index	317
IBM Redbooks review	323

Preface

This IBM Redbook is a survey of open source software that is available for z/OS and OS/390 UNIX. It presents a brief history of open source software, addresses licensing issues and describes a general model of obtaining, building and enhancing open source software packages. Many open source software packages are then addressed, each in its own chapter.

The hardcopy version of this redbook includes a CD that is not an IBM product and not formally supported. The CD contains many open source software packages ported to OS/390. Both the source code and the compiled executables are included. Typically, the software is installed in /usr/local which should be a separate HFS. Each package is put in its own directory under /usr/local. For every command there is a man page. All commands and man pages are copied to /usr/local/bin and usr/local/man, respectively.

These packages are included:

Apache	A highly popular Web server
Atools	Small scripts for working with ASCII files
Autoconf	Produces shell scripts which automatically configure source code packages
Automake	A tool for generating Makefile.in files for use with autoconf
Bash	The Bourne Again SHell, which is compatible with the UNIX sh and offers many extensions found in csh and ksh
Bison	GNU's upwardly compatible replacement for the parser generator yacc
CVS	A version control system that allows you to keep old versions of files, a log of who, when, and why changes occurred
Diffutils	A set of utilities for comparing files, which includes GNU diff, which shows line-by-line changes in several flexible formats
Emacs	A popular UNIX editor; it supports normal telnet sessions and the X Window system
Fgres	Tool to search and substitute patterns in text and binary files
Findutils	Tools for finding files, directories and libraries
Flex	GNU's fast lexical analyzer—similar to lex
Gd	A C library for dynamically creating images

Gmake	GNU's tool to determine how to efficiently build programs based on a configuration "makefile"
Gnats	GNU's bug tracking system
GNU grep	GNU's tool to search files for patterns
Groff	A document formatting system based on a device-independent version of troff
Gzip	GNU's program for compressing and decompressing files
Id-utils	A package of simple, fast, language-independent tools that index identifiers, literals, or words of human-readable text
Info-ZIP	The definitive open source zip/unzip tool
Ittools	Tools for creating and maintaining Makefiles
Jpeg library	A library that can create and decompress JPEG images
Korn shell	A powerful replacement for the standard shell
Less	A pager similar to the <code>more</code> command
Libpng	Tools to create and manipulate PNG image files
Libtool	Tool to simplify the use of shared libraries
Lynx	A text-based Web browser
M4	A macro processor
MSNTP	A Simple Network Time Protocol tool
MySQL	A relational database management system
Nedit	An X Window-based editor, with a GUI interface much like you find on Macintosh and Windows systems
OpenSSH	A free version of the secure shell (SSH) suite of network connectivity tools
OpenSSL	An implementation of the Secure Sockets Layer and Transport Layer Security protocols
Patch	GNU's version of the <code>patch</code> command used for incorporating changes into a number of files
Perl	A programming language that combines the features and capabilities of C, sed, awk, and sh
PHP	A server-side scripting language to create dynamic Web pages
RCS	A source code control system
Rlogin client	A terminal emulation program

Rsh client	A remote shell program
Samba	A UNIX SMB server (Windows file and print serving protocol)
Squid	A hierarchical HTTP/FTP proxy/cache with redirect options
THE	The Hessling Editor - an XEDIT-like editor
Vim	An improved vi editor
Webalizer	A tool to analyze Web server log files and displays the results as HTML pages
Xterm client	An X Window terminal emulation program
Zlib	A general compression library

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Michael MacIsaac is a team leader for S/390 redbooks and workshops at the ITSO Poughkeepsie Center. He writes about and teaches IBM classes on OS/390, focusing on UNIX System Services. Michael has worked at IBM for 13 years, mainly as a UNIX programmer. He has written Redbooks on Windows NT porting to OS/390, S/390 file serving capabilities, S/390 server consolidation, and was the main author of the redbook *Networked Applications on OS/390 UNIX*.

Thorsten Brockmeier is an engineering student on a vocational program with IBM Germany. His areas of expertise include relational databases, SQL and procedural/object-oriented programming. He has ported and written on MySQL.

Sándor Bárány is Systems Architect for M&M Investitionsberatungs GmbH, a consulting firm in Austria, providing professional services. He has 25 years of experience in the IT field. He holds an MS degree in Mathematics from the ELTE University of Budapest, Hungary. He applies his extensive field experience in portability, connectivity, system architectures and security as an architect and project leader in the Austrian banking industry.

Guillermo Freige is a Systems Programmer for La Plata National University, Argentina. He has 13 years of experience in VM/CMS technical support, with 8 years of specialized experience in UNIX and Linux. He administers Linux for S/390, VM/ESA and OS/390 MVS and UNIX System Services. He is also the Senior Network Administrator of the Buenos Aires State Government.

state-wide intranet, a Intel/Linux-based TCP/IP network with 150+ servers, and does freelance network consulting.

Stefan Koesling is a vocational student in Germany, studying Information Technology Management. His interest in open source software arose from exposure to the Linux operating system.

Kiran Madnani is a graduate student at Rensselaer Polytechnic Institute, NY, and has worked as a Research Assistant with the Computer Science department at RPI. His areas of interest include computer networks, distributed and parallel computing, and open source software development. He holds a Bachelors degree in Computer Engineering from University of Mumbai, India.

Fulvio Malfatto is an independent consultant working in Italy. He has about 20 years of experience in the IT field, 10 of them spent as an IBM System Engineer on VM in the Technical/Scientific area, and later working mainly with UNIX (AIX and Linux) and open source software. His interests include the integration between Web-enabled applications and legacy-system applications. He holds an MS degree in Industrial Chemistry from the University of Torino.

Ralf Schandl is an Advisory IT Specialist for IBM Global Service in Germany. He has five years of experience as a software developer of network-related software. He works mainly on AIX, but also on OS/390 UNIX. He relies heavily on open source software tools in his job.

Egon Terwedow is an OS/390 Technical Specialist working for the Ford Motor Company in Germany. He has more than 12 years of experience in the MVS arena, with expertise in OS/390 systems programming, configuration, and performance management. Recently, he set up a new interface to mainframe data using the OS/390 Internet Technology and the UNIX System Services in the Ford data centers in the US and Europe. He also participated in writing the redbook *Enterprise Web Serving with the Lotus Domino Go Webserver for OS/390*.

James Tison is a Senior Software Engineer with IBM in Poughkeepsie, NY. He handles architectural issues for the TPF operating system and most recently has been working on exploiting OS/390 UNIX as a cross-development platform for TPF. He has 22 years of experience in the IT field. His areas of expertise include TPF, OS/390, VM/CMS, and Linux.

Jeff Trawick is an Advisory Software Engineer with IBM in Research Triangle Park, NC. He currently works on the Apache Web server. He has ten years of

experience working on SNA and TCP/IP networking products for S/390 at IBM. He holds an MS degree in Computer Science from the University of Alabama. His areas of interest include TCP/IP, socket programming, UNIX, and open software development.

Thanks to the following people for their contributions to this project (in no particular order):

Rich Conway, Paul de Graaff
International Technical Support Organization, Poughkeepsie Center

David Fiander, Cameron Bates, and others at MKS - for the ports on the OS/390 GNU Utilities Web page
Mortice Kern Systems

Greg Smith - for the OS/390 ports of xterm and bash
Lockheed Martin

Paul von Berhen - for the OS/390 ports of nedit and Info-ZIP
Storage Technologies

Preston Bannister - for tree tools
NSA

Garth Godfrey, Bill Schoen, John Czukermann, William Kinnaman, John Kapernick, Tim Kane
IBM Poughkeepsie

Peter Prymmer - for help with Perl
Forte Software Division of Sun Microsystems

Al Dunsmuir
IBM Toronto

Phil Sidler - for the port of Nedit 5.1
Airborne Express

Manfred Porst - for help with porting OpenSSH
csg Computer Service GmbH

Dieter Kessler - for notes on porting CVS
Software AG

Wayne Johnson - for help with porting CVS
Digital Communications Consultants

Paul Gilmartin - for quick answers on the mvs-oe list server
StorageTek

Franck Gerodou - for Samba fixes
AGF

Werner Lemberg - for getting EBCDIC support into groff
GNU

Paul Anderson; IBM UK, Tibor Lietz; IBM Germany and Michael Mai;
Deutsche Bank - for continuing work on OpenSSH

Krishnaswami, Natarajan - for ports of GNU grep
IBM UK and IBM Germany

Josh Knight - for forum answers on porting questions
IBM US

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 323 to the fax number shown on the form.
- Use the online evaluation form found at ibm.com/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Introduction

“...source code is fundamental to the furthering of computer science and freely available source code is truly necessary for innovation to continue.”

Chris DiBona, Sam Ockman, Mark Stone, *OpenSources: Voices from the Open Source Revolution*

There is a major cultural change happening in the computer industry today. The act of programming is changing from being an industrial or enterprise-centric exercise to becoming a community exercise. Rather than program development being done with a team in a single location working for a single company with tight security on the source code, it is being done in an open, distributed fashion by people from many different companies and institutions with the source code available for all to copy and use. Besides cost, this change is being fueled by inexpensive, fast, global communications, with the Internet as the main conduit. The product of this programming is called *freeware* or *open source software*.

This cultural change of open source software is beginning to create an economic change. Business models for selling software that were formerly viable are slowly being made less effective by the availability of free software.

In some respects, the cultural change that open source software is creating is a direct contradiction to the OS/390 (MVS) culture. The OS/390 culture is to use only mission-critical, well-tested tools and applications. In contrast, the open source software culture seems to promote grabbing packages off the Internet and simply trying them out. A balance between these approaches must be achieved.

On one hand, the software associated with this redbook is literally “grabbed off the Internet”, tried out, and put on the CD. On the other hand, many of the software packages have been used and tested by thousands or sometimes millions of users.

The user of this software must understand this balance. To reiterate our premise, the software associated with this redbook is not an IBM product and not formally supported. But many of the packages may be very solid, very useful tools, especially to software and Web developers.

1.1 Terminology

The terms *free software* and *open source software* can be used interchangeably, as stated by Bruce Perens, the main author of the Open

Source Definition. The Free Software Foundation claims that these terms are almost synonymous, but that there is a difference. This redbook uses the term *open source software* which now seems to be more commonly used.

If you would like more detail on open source software, start with the book *OpenSources: Voices from the Open Source Revolution* published by O'Reilly & Associates. It is a collection of papers and it is freely available on the Internet at:

<http://www.oreilly.com/catalog/opensources/book/toc.html>

It includes papers from Richard Stallman, Linus Torvalds, Eric Raymond, Bruce Perens and many other visible open source software advocates.

1.2 A brief history of open source software

The increase in the creation and use of open source software may appear to be a new phenomena, but in reality it has been around for a long time. Its roots may be traced to the birth of both UNIX and TCP/IP. In the 1960s, the concept of open source software didn't have to exist; people assumed the code would be made available (in fact, IBM only started charging for software in 1970). Access to the hardware was the bigger issue because it was so much more expensive than it is today.

Freely available software has touched all aspects of computing. Without it, computing would not be nearly as advanced and global as it is today. The following technologies are some of the major aspects of computing affected by open source software; we discuss them in further detail in the following sections:

- UNIX
- The Internet and TCP/IP
- e-mail
- News groups
- The Free Software Foundation
- The World Wide Web
- Linux

UNIX

UNIX was largely started by Ken Thompson and Dennis Ritchie of AT&T's Bell Labs in 1969. Because AT&T saw no commercial value for UNIX at the time, the first copy of UNIX left AT&T's Bell Labs for the University of California at Berkeley in 1974. Many people saw the need for a free operating system, especially for universities. To fill this need, the first Berkeley Software Distribution (BSD) of UNIX was created by Bill Joy early in 1977. Of course,

the distribution came with the source code. There was a \$1000 fee from Berkeley for the tape, but there were no restrictions on redistributing it.

Later, AT&T tried to earn money with System V UNIX. UNIX Systems Laboratory (USL) was spun off. At the June 1979 USENIX meeting in Toronto, AT&T announced a price hike for UNIX (per CPU) from \$1000 to \$7500 for educational institutions, and from \$5000 to \$40000 for commercial institutions. They were successful for a while earning money and tried to sue BSD which had forked into some commercial ventures, but the “genie had been let out of the bottle”. Today FreeBSD and other variants of the original Berkeley Software Distribution thrive, but UNIX is no longer a product - it is a brand name.

The Internet and TCP/IP

The first node-to-node message over the ARPANET was sent between UCLA and the Stanford Research Institute (SRI) in October 1969. This was the start of the Internet. In 1974, Vint Cerf and Bob Kahn published *A Protocol for Packet Network Intercommunication*, which specified in detail the design of a Transmission Control Program (TCP). Today, the Internet is growing faster than any other technology in history.

In January 1986, a group of researchers funded by the government got together to discuss Internet standards; thus the Internet Engineering Task Force (IETF) was born. The IETF has always been an open organization in that all documents are public and freely available over the Internet. Anyone can attend an IETF meeting. The motto of the IETF is to have a “rough consensus and running code”. It could be viewed as the most important open source organization in existence today since the Internet is so crucial to today’s computing models.

Following is a pertinent quote from an IETF member:

The IETF supported the concept of open sources long before the Open Source movement was formed. Up until recently, it was the normal case that “reference implementations” of IETF technologies were done as part of the multiple implementations requirement for advancement on the standards track. This has never been a formal part of the IETF process, but it was generally a very useful by-product. Unfortunately this has slowed down somewhat in this age of more complex standards and higher economic implications for standards. The practice has never stopped, but it would be very good if the Open Source movement were to reinvigorate this unofficial part of the IETF standards process.

Scott Bradner, *The Internet Engineering Task Force, OpenSources: Voices from the Open Source Revolution*

Domain Name System (DNS)

The Domain Name System (DNS) is an Internet service that translates human-readable domain names into machine-readable IP addresses. DNS has the useful capability of passing on questions. If one DNS server doesn't know how to translate a particular domain name, it continues to ask others, until the correct IP address is returned.

BIND

The package BIND (from "Berkeley Internet Name Domain") is an implementation of DNS which provides a reference implementation of the major components of DNS:

- named - a DNS server
- A DNS resolver library
- Tools for verifying the operation of the DNS server

The BIND DNS Server is used on the majority of name serving machines on the Internet, providing a stable name serving architecture. It was originally written at UC Berkeley as a graduate student project under a grant from DARPA. Versions of BIND through 4.8.3 were maintained by CSRG at UC Berkeley. Douglas Terry, Mark Painter, David Riggle and Songnian Zhou made up the initial BIND project team. This tool is a critical piece of the globally successful Internet, and it is open source software. At the time of writing, the current level is 8.2.2, patch level 5, and can be found on the Web at:

<http://www.isc.org/products/BIND/bind8.html>

e-mail

In 1971, Ray Tomlinson of BBN invented an electronic mail (e-mail) program to send messages across a distributed network. The original program was derived from an intramachine e-mail program called SNDMSG and an experimental file transfer program called CPYNET. The first node-to-node message was sent between UCLA and SRI in October of that year. The next year, Tomlinson wrote basic e-mail message send and read programs.

Sendmail is an open source e-mail server that is widely used. It performs a large amount of mail routing on the Internet.

News groups

Usenet was conceived in 1969 by three graduate students: Tom Truscott, Jim Ellis and Steve Bellovin, of Duke University. Usenet is like an electronic news magazine or a "world". It has various news groups organized by topic areas, and in a number of languages. Users can post articles to any of the

newsgroups. Today it links millions of people and computers to over 10,000 different news groups with millions of appends.

In 1987, UUNET was founded to provide commercial UUCP and Usenet access. Rick Adams wrote “B” news, for which the source code was openly available.

Like e-mail, news groups were founded upon open source software.

GNU and the Free Software Foundation

In 1984, Richard Stallman quit his job at MIT and began writing GNU (a recursive acronym that stands for GNU's Not UNIX) software. He felt he had to leave his job in order to distribute free software, lest M.I.T. claim to own the work. The first big project was the creation of the GNU compiler, gcc.

He then began work on GNU Emacs in September 1984. The head of the MIT lab allowed Richard to continue using the lab's facilities, so when Emacs became usable, he put it on the FTP server named prep.ai.mit.edu.

In 1985 the Free Software Foundation was started as a tax-exempt charity. Others became interested in this group and started donating software to the organization. In order to make money, they charged for tape distributions of free software and manuals.

X Window System

With his software, Stallman wanted to avoid what had happened with the X Window System. It was developed at MIT and released as free software with a permissive software license. This system was adopted by various companies (IBM being one), and then re-released with their UNIX operating systems without the source code. Stallman describes the situation the following way:

The developers of the X Window System did not consider this a problem--they expected and intended this to happen. Their goal was not freedom, just “success”, defined as “having many users”. They did not care whether these users had freedom, only that they should be numerous.

This led to a paradoxical situation where two different ways of counting the amount of freedom gave different answers to the question, “Is this program free?” If you judged based on the freedom provided by the distribution terms of the MIT release, you would say that X was free software. But if you measured the freedom of the average user of X, you would have to say it was proprietary software. Most X users were running

the proprietary versions that came with Unix systems, not the free version.”

Richard Stallman, *The GNU Operating System and the Free Software Movement*, *OpenSources: Voices from the Open Source Revolution*

World Wide Web

The Web exploded as fast and as successfully as it did in part because it started as an open source software project. The Mosaic browser was made available from the National Center for Supercomputing Applications at the University of Illinois in April 1993. The source code for Mosaic is still available on the Web at:

<ftp://ftp.ncsa.uiuc.edu/Mosaic/>

After the developers of Mosaic left to found Netscape in 1994, the source code was no longer open source, as selling the browser became their business model. Fearing the loss of control of this nascent market, Microsoft made its browser free, which severely impacted the Netscape business model. In 1998 Netscape decided to create the Mozilla organization and again open up the source code. That code is available today at:

<http://www.mozilla.org/source.html>

Apache Web server

On the server side, the Apache Web server is an open source package. In 1995, Apache became the leading Web server and has continued to gain market share. Figure 1 on page 6 shows the market share for Web servers.

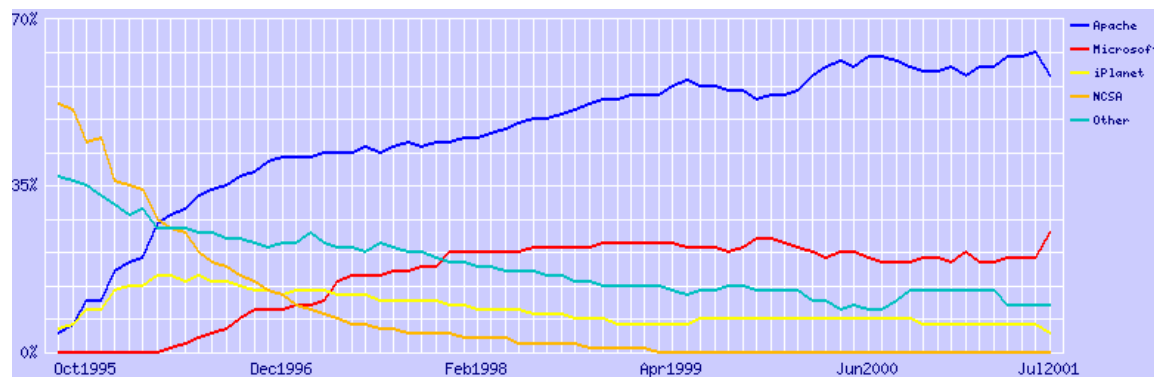


Figure 1. Web server market share

The survey on which this figure is based was done by Netcraft, which queries millions of servers on the Web. Details of the survey are found at the following Web site:

<http://www.netcraft.com/survey/>

Linux

The term Linux is used for two concepts:

- A UNIX-like operating system kernel
- A number of operating system *distributions*, which include the Linux kernel, many hundreds of other tools or packages, and software to install the operating system

Sometimes the latter flavor of the term is referred to as GNU/Linux (especially by Richard Stallman), but in general when people use the term “Linux”, they are referring to the distribution, which includes the kernel.

Linux kernel

Linux as an operating system was initially created as a hobby by Linus Torvalds at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in 1991 when he released version 0.02. In 1994, version 1.0 of the Linux Kernel was released.

Today there are an estimated 10 million users of Linux. The kernel has been ported to many different types of hardware, including S/390 (which has become quite popular since late 1999).

As it pertains to open source software, Linus Torvalds writes the following:

The power of Linux is as much about the community of cooperation behind it as the code itself. If Linux were hijacked--if someone attempted to make and distribute a proprietary version--the appeal of Linux, which is essentially the open-source development model, would be lost for that proprietary version.

Linus Torvalds, *The Linux Edge, Open Sources: Voices from the Open Source Revolution*

1.3 The development model of open source software

It is often asked about open source software projects “How is the project managed?” and “How is source code development done?” Open source software projects that are developed by programmers from all over the world

have to be managed much differently than software projects developed in an enterprise.

For insight on this subject, we again turn to the *Open Sources* book, to the article “Open Source as a Business Strategy” by Brian Behlendorf of Apache. At the end of his paper, Brian discusses the logistics of how the highly successful Apache Web server is managed. The salient points are:

- The source code control system
- Discussion forums
- Bug tracking software

To quote Behlendorf:

We have a nice set of available, well-maintained tools used in the Apache Project for allowing our distributed development process to work.

Most important among these is CVS, or Concurrent Versioning System. It is a collection of programs that implement a shared code repository, maintaining a database of changes with names and dates attached to each change. It is extremely effective for being able to allow multiple people to simultaneously be the “authors” of a program without stepping over each others' toes. It also helps in the debugging process, as it is possible to roll back changes one by one to find out exactly where a certain bug may have been introduced. There are clients for every major platform, and it works just fine over dial-up lines or across long-distance connections. It can also be secured by tunneling it over an encrypted connection using SSH.

The Apache project uses CVS not just for maintaining the actual software, but also for maintaining our “STATUS” file, in which we place all major outstanding issues, with comments, opinions, and even votes attached to each issue. We also use it to register votes for decisions we make as a group, maintain our web site documents with it, manage development documents, etc. In short it is the asset and knowledge management software for the project. Its simplicity may seem like a drawback--most software in this space is expensive and full-featured--but in reality simplicity is a very strong virtue of CVS. Every component of CVS is free--the server and the clients.

Another essential element to an open-source project is a solid set of discussion forums for developers and for users. The software to use here is largely inconsequential — we use Majordomo, but ezmlm or Smartlist or any of the others would probably be fine. The important thing is to give each development effort their own list, so that developers can self-select their interests and reasonably keep up with development. It's also smart to

create a separate list for each project to which the CVS server emails changes that get made to the CVS repository, to allow for a type of passive peer review of changes. Such a model is actually very effective in maintaining code standards and discovering bugs. It may also make sense to have different lists for users and developers, and perhaps even distinguish between all developers and core developers if your project is large enough. Finally, it is important to have archives of the lists publicly available so that new users can search to see if a particular issue has been brought up in the past, or how something was addressed in the past.

Bug and issue tracking is also essential to a well-run project. On the Apache Project we use a GNU tool called GNATS, which has served us very well through 3,000+ bug reports. You want to find a tool that allows multiple people to answer bug reports, allows people to specialize on bugs in one particular component of the project, and allows people to read bug reports by email and reply to them by email rather than exclusively by a web form. The overriding goal for the bug database is that it should be as easy and automated as possible both for developers to answer bugs (because this is really a chore to most developers), and to search to see if a particular bug has already been reported. In essence, your bug database will become your repository for anecdotal knowledge about the project and its capabilities. Why is a particular behavior a feature and not a bug? Is anyone addressing a known problem? These are the types of questions a good bug database should seek to answer.

The open-source approach is not a magic bullet for every type of software development project. Not only do the conditions have to be right for conducting such a project, but there is a tremendous amount of real work that has to go into launching a successful project that has a life of its own. In many ways you, as the advocate for a new project, have to act a little like Dr. Frankenstein, mixing chemicals here, applying voltage there, to bring your monster to life. Good luck.

For more information about GNATS, refer to Chapter 17, “Gnats” on page 89 in this redbook.

1.4 OS/390 resources

There is at least one news group (or list server) and three Web sites where open source software for OS/390 is discussed and can be downloaded.

1.4.1 News group

The OS/390 UNIX list server, called mvs-oe, is a great resource. The participants are customers who use OS/390 UNIX, and developers from IBM. Subscribers to the mailing list participate by asking and answering questions, which are all received as e-mail.

You can also search the archives, which are maintained on the Web at:

<http://www.marist.edu/htbin/wlvindex?mvs-oe>

Subscribing to mvs-oe

To subscribe to the listserver, you can send an e-mail to `listserv@vm.marist.edu` and in the first line put:

```
sub mvs-oe your_first_name your_last_name
```

After you are subscribed, you will receive further instructions on how to use the mailing list. Read the instructions carefully and save the e-mail for future reference. You can then append to the list by sending an e-mail to `mvs-oe@vm.marist.edu`. Remember, administration is done via the e-mail address `listserv@vm.marist.edu`.

Unsubscribing from mvs-oe

To unsubscribe from the list, send an e-mail with the first line:

```
signoff mvs-oe
```

Additional resources are described on the OS/390 UNIX *Forums* Web page at:

<http://www.s390.ibm.com/oe/bpxaldis.html>

1.4.2 Web sites

Although the open source packages addressed in this redbook were obtained from many different Web sites, most of them can be found at these three sites:

OS/390 UNIX Tools and Toys Web site

The OS/390 UNIX Tools and Toys Web site contains a set of free software packages maintained as part of the OS/390 UNIX site. Many of the tools relate to the MVS side of OS/390. Note that while all of the packages are free, *they are not all open source*.

This site is located at:

<http://www.s390.ibm.com/oe/bpxaltoy.html>

MKS OS/390 GNU utilities

The Mortice Kern Systems (MKS) OS/390 GNU utilities ftp site is:

<ftp://ftp.mks.com/pub/s390/gnu/>

At the time of writing, these packages were available:

- Autoconf 2.12
- Binutils and libbfd 2.8.1
- Bison 1.25
- Emacs 19.34
- Flex 2.5.4a
- m4 1.4
- Make 3.76.1
- Perl 5.004.03
- RCS 5.7
- Samba 1.9.18p1
- WebAgent 1.6

Most of these utilities are included on the CD-ROM that accompanies the hardcopy version of this redbook.

GNU

Though many of the GNU software packages will not build cleanly on OS/390 UNIX without modification, some of them will build cleanly. The GNU Web site is still probably the best source of open source software:

<http://www.gnu.org/software/software.html>

1.4.3 An OS/390 TSO resource

The following Web Site (hosted by Dignus, LLC) contains open source software for TSO users:

<http://www.dignus.com/freebies/>

It advertises the following packages, which seem to be focused on C/C++ development:

- byacc - the Berkeley YACC utility for parser generation
- bwbasic - Bywater BASIC interpreter
- dig - the DIG utility for querying DNS servers
- flex - the Berkeley Fast LEXical generator
- grep - the standard utility to search for text in files
- indent - a program to "pretty print" C source
- printps - a program to convert text to postscript
- udsmtplib - Batch SMTP client (send e-mail from jobs)

- what - the WHAT utility for object versions

The packages are distributed in TSO XMIT format.

Chapter 2. Licensing software

“The GPL basically says that you may copy and distribute the software licensed under the GPL at will, provided you do not inhibit others from doing the same, either by charging them for the software itself or by restricting them through further licensing. The GPL also requires works derived from work licensed under the GPL to be licensed under the GPL as well.”

Chris DiBona, Sam Ockman, Mark Stone, *OpenSources: Voices from the Open Source Revolution*

There are many different licenses under which open source software is distributed. To many users, the license under which they are using the software is not important; they just want to know that the software is free. To other users, however, and especially to developers of open source software, licensing is an important issue. There are ongoing discussions about the pros and cons of the various licenses. This chapter briefly addresses these issues by suggesting sources of material that seem to be well accepted in the community.

2.1 The GNU General Public License (GPL)

The GPL is by far the most commonly used license under which open source software packages are shipped. The philosophy of the GPL is to give everyone certain liberties with each software package distributed under it. They are:

- To run the software
- To copy the software
- To modify the software
- To redistribute modified versions

However, it *does not* allow restrictions to be added to the software. This ensures that once a software package is distributed under the GPL, it will remain free or open. Additionally, if a software package is written that includes software distributed under the GPL, it too is supposed to be distributed under the GPL. This is the “viral” nature of the GPL.

A copy of the GPL can be found in Appendix A, “The GNU General Public License” on page 289. It is also on the Web at:

<http://www.gnu.org/copyleft/gpl.html>

2.2 Other licenses

Some people may assume that open source software is in the *public domain*. This is not true. By far, the majority of open source software is licensed, very often via the GPL. However, there are many other licenses, including these popular ones:

- Free Software Licenses in General
- The GNU Lesser General Public License (LGPL)
- The X, BSD, and Apache Licenses
- The Artistic License
- The Netscape Public License and the Mozilla Public License

In the paper “The Open Source Definition”, published in the book *Open Sources: Voices from the Open Revolution*, Bruce Perens discusses open source software and the nature of different licenses. He also has a section that provides an excellent comparison of public domain, the GPL, and all of the preceding licenses.

It should be noted that the IBM does not necessarily agree with all of the thoughts in this paper, but reading it may help you understand why licensing issues are important to the open source software community. This paper is freely available on the Web at:

<http://www.oreilly.com/catalog/opensources/book/perens.html>

For reference, another important paper cited in “The Open Source Definition” is “The Cathedral and the Bazaar” by Eric S. Raymond. It is on the Web at:

<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>

2.3 The Open Source Definition

Though not a license, the Open Source Definition is an important document. It is derived from the Debian Free Software Guidelines. Bruce Perens composed the original draft and it was refined using input from developers in mid-1997. It was composed so the corporate world might be more accepting of the whole concept of freeware.

A copy of the Open Source Definition can be found in Appendix B, “The Open Source Definition” on page 297. The most current version is maintained on the Web at:

<http://www.opensource.org/osd.html>

2.4 The IBM Public License

Besides the other, more widely known licenses, IBM has created a license for use with open source software. The IBM Public license is less restrictive to programmers than the GPL. It does not have the “viral nature” of the GPL in that only changes to the code shipped under the license need to be made available. If you want to include code under the IBM Public License with your code under a different license, there is nothing that forces you to make the latter code available. Additionally, the IBM Public License pertains to documentation as well as source code.

It has been certified as Open Source by the Open Source Initiative and has been labeled as a “free software” license by the Free Software Foundation. The complete text of the IBM Public License can be found in Appendix C, “The IBM Public License” on page 299.

Chapter 3. What's on the CD-ROM

The CD-ROM included with the hardcopy version of this redbook contains the following four files and one subdirectory. The packages Infozip, the jpeg library, the Korn shell, MySQL and Squid are not included due to legal concerns.

README.txt	A file describing how to move the other files to z/OS UNIX and use them.
5944-01src.tar	A tar file of all the packages described in this book except those noted in the first paragraph. This file is not compressed because it consists largely of other compressed archives.
5944-01bin.tar.Z	A tar file of all the executables, scripts and man pages except those noted in the first paragraph, but no source code.
SG245944.pdf	The redbook in PDF form.

These files are also available on the OS/390 Tools and Toys site at:

<http://www-1.ibm.com/servers/eserver/zseries/zos/unix/redbook/index.html>

As of March 2002, the files on the CD and the Web site were identical. However, the files on the Web may be more current.

3.1 Package installation ordering

Some of the packages that are installed are required to build other packages, and these prerequisites can be tricky. The order of package installation is important. Specifically:

- The gzip package is required for the many packages that are gzip-compressed.
- GNU make must be used for Perl, infozip, groff and perhaps some others.
- GNU patch is required for groff, nedit and Perl.
- Bison is required for bash, cvs, flex and m4.
- Perl is required for autoconf and automake.
- m4 is required for autoconf, automake, and Perl.
- The ext script is required for Apache and libpng. It is used to extract both text and binary data from the tar file.
- Openssh requires Openssl, zlib and autoconf.
- gd requires libpng and jpeg libraries.
- Webalizer requires gd and zlib

Therefore, install the packages in this order:

1. gzip
2. GNU make
3. GNU patch
4. Bison
5. Perl
6. m4
7. atools
8. zlib
9. gd

After these are installed, the order in which you install the other packages should not matter.

Because these packages are used in the install process of other packages immediately, they must be found in the user's PATH environment variable. See 3.6, "Setting the PATH and MANPATH environment variables" on page 23 for the path variables.

3.2 Installing the packages from source

This section discusses the installation of the open source software packages. The packages are distributed in source format in the file 5944-01src.tar and in binary format in the file 5944-01bin.tar.Z. On the CD that accompanies this redbook, we also have two directories: src (which contains the source for the individual packages) and binaries (which contains the individual binaries). Any text files in both archives are in EBCDIC, so no conversion from ASCII is necessary. (However, the source distribution contains archives with text in ASCII, so conversion will be necessary later for most packages).

3.2.1 Source distribution installation

Installing the source distribution consists of getting the tar file to OS/390, unwinding it, and optionally invoking the install script 5944-01.install. The install script honors the required package installation ordering described in 3.1, "Package installation ordering" on page 17.

There is also a small wrapper around 5944-01.install to put the build in the background and capture the output:

```
$ cat inst
echo "time nohup ./5944-01.install /usr/local > installLog.txt 2>&1 &"
time nohup ./5944-01.install /usr/local > installLog.txt 2>&1 &
```

3.2.2 Installing quickly

So the quick way to install all the code is the following:

```
$ cd /usr/local
$ id
uid=0(AAAAAAA) gid=0(SYS1) groups=87854(SYS2)
$ ls
5944-01src.tar
$ pax -rf 5944-01src.tar
$ ls
5944-01src.tar  src
$ cd src
$ ./inst
executing: nohup ./5944-01.install /usr/local > installLog.txt 2>&1 &
```

3.2.3 Details on installing

You can move the tar file from the PC to OS/390 via FTP in binary. Here is an example from a DOS session:

```
D:>ftp wtsc69oe
Connected to wtsc69oe.itso.ibm.com.
User (wtsc69oe.itso.ibm.com: (none)): mikem
331 Send password please.
Password:
ftp> cd /usr/local
ftp> bin
ftp> put 5944-01src.tar
...
```

Once the files are on OS/390, they can then be unwound via the `pax` command:

```
$ pax -rf 5944-01src.tar
```

The tarball contains one directory called `src` that resembles the one on the top level of the CD. Each package is in a directory with an obvious name below that main directory:

```
$ cd src
$ ls -F
5944-01.executables  gd/                openssl/
5944-01.install*    gmake/            openssl/
5944-01.packages    gnats/            patch/
README              grep/             perl/
ansi.ti             groff/            perl56/
apache/             gzip/             php/
```

apache2/	idutils/	rcc/
atools/	itools/	rlogin/
autoconf/	jpeg/	rsh/
automake/	ksh/	rxvt.ti
bash/	less/	samba/
bison/	libpng/	samba22/
cvs/	lynx/	the/
diffutils/	m4/	vim/
emacs/	misctools/	webalizer/
fgres/	msntr/	xterm/
findutils/	mysql/	xterm.ti
flex/	nedit/	zlib/

In addition, the following files are created:

README	The README file
5944-01.install	The install script
5944-01.packages	The list of package names and corresponding tar files
5944-01.executables	The list of expected executables
ansi.ti, rxvt.ti, xterm.ti	Definition files for useful terminals

Then you can issue the `5944-01.install` command to build all software packages as follows:

```
$ time nohup 5944-01.install /usr/local > 5944-01.log 2>&1 &
[1]      16781148
```

The `time` prefix will output how long it takes to run. The `nohup` prefix allows you to exit the shell and leave the script running (ignore the HUP signal generated when you exit the shell), and `5944-01.install /usr/local` is the crux of the command: it installs all packages in `/usr/local`. The `> 5944-01.log 2>&1` suffix redirects stdout and stderr to the file `5944-01.log`, and the trailing `&` puts the whole process in the background.

It is suggested that you install as root (UID 0). This is specifically so `chmod u+s rlogin` works. Also, root should have RACF (or other security product) permission to the facility classes BPX.FILEATTR.PROGCTL so the program control extended attribute bit for `smbd` can be set. See 42.2, “Installation of Samba” on page 231.

When the previous command was run, the file `5944-01.log` was created. The synopsis of this output file follows:

```
Install script 5944-01.install was run on Tue Jun 27 10:16:01 EDT 2000
```

```
Creating necessary directories..
```

```
creating /usr/local/bin ...
creating /usr/local/lib ...
creating /usr/local/share ...
creating /usr/local/man ...
creating /usr/local/man/C ...
creating /usr/local/man/C/cat1 ...
creating /usr/local/man/C/cat5 ...
creating /usr/local/man/C/cat7 ...
creating /usr/local/man/C/cat8 ...
creating /usr/local/var ...
installing all 27 packages ...
```

```
Installing gzip-1.2.4.tar on Tue Jun 27 10:16:01 EDT 2000 ...
output is being written to /usr/local/gzip/gzip.log ...
done installing /usr/local/gzip ...
```

```
Installing make-3.76.1.os390.tar.Z on Tue Jun 27 10:16:15 EDT 2000 ...
output is being written to /usr/local/gmake/gmake.log ...
done installing /usr/local/gmake ...
.....(all other packages)
```

Another way of validating the build is to look for compiler errors in the log files. The following command searches for the string ERROR followed by the C/C++ prefix of CBC. The errors that are found *appear* to be benign.

```
$ cat look4errors
for i in */*Log.txt
do
    echo $i
    grep "ERROR CBC" $i
    grep "return code from" $i
done
```

Also, you can install a single package in one of two ways:

1. Via the 5944-01.install script with the syntax `5944-01.install installDir [package ...]`. For example, to install only the m4 and autoconf package:

```
$ 5944-01.install /usr/local m4 autoconf
```

2. Via the package's specific install script; for example:

```
$ cd m4
$ ./m4.install m4-1.4.os390.tar.Z /usr/local
```

In this redbook, each of the source packages is provided individually for convenience (rather than having to open up the big tarball containing all packages). Thus, each of the packages can be individually ftp'd and installed using the package's install script.

3.3 Installing binaries

Besides the source code, there is also a binary distribution in the file 5944-01bin.tar.Z. The binaries were built on an OS/390 V2 R9, OS/390 V2 R10, and a Z/OS R1 system. If you have V2 R8 or earlier, you will need to build the binaries from the source distribution.

The tar file can be moved to OS/390 in binary in the same fashion as the source code distribution. There is no install script as it is simply a hierarchy of executables, man pages and additional files.

The binary distribution can be "installed" as follows:

```
$ ls -l 5944-01bin.tar.Z
-rw-rw-r-- 1 STC      OMVSGRP  96155136 Aug 29 14:16 5944-01bin.tar.Z
$ pax -rzf 5944-01bin.tar.Z
$ ls -F
5944-01bin.tar.Z  com/          include/      logs/         share/
bin/              examples/     lib/          man/          var/
cgi-bin/          htdocs/       libexec/      proxy/
```

In this redbook, each binary package has been provided for convenience under a directory 5944-01bin. Thus, in order to install the binaries for a particular package, one would have to move the respective binary in the same fashion as the source code distribution. The tar for each package consists of a simple hierarchy of executables, man pages and other additional files.

3.4 Installation size

Use the 5944-01src.tar file if you are interested in the source code or would like to build the packages yourself. It is about 66 MB and when unwound requires an additional 67 MB, for a total of about 133 MB. When all the packages are built, it occupies much more space than that, so a minimum of 750 MB is recommended.

Use the 5944-01bin.tar.Z file if you only want to use the packages and are not concerned with the source code. It is about 72 MB and when unwound requires an additional 177 MB, for a total of 249 MB.

3.5 Installation directory

It is recommended that these packages be put in the `/usr/local` directory. This could be created as a separate HFS, so that when OS/390 is upgraded, this file system is unaffected.

If you want to have as few system-specific HFSs as possible, you could use a large customized `/etc` file system. To do this you could symbolically link `/usr/local` to perhaps `/etc/local`. For example:

```
ln -s /etc/local /usr/local
```

If you do not have read/write access to `/usr/local`, the packages can be installed elsewhere; however, this scenario has not been completely tested. Some packages may still have hardcoded dependencies to `/usr/local`.

3.6 Setting the PATH and MANPATH environment variables

We suggest that you add `/usr/local/bin` to your `PATH` environment variable and both `/usr/local/man/%L` and `/usr/local/man` to your `MANPATH` environment variable. This way, all of the commands (executables) and man pages from the CD are available to you. For example, here are some values for these environment variables:

```
$ echo $MANPATH
/usr/local/man/%L:/usr/local/man:/usr/man/%L
$ echo $PATH
/u/mikem/bin:/usr/local/bin:/bin:.
```

When the open source software packages are used, either via building them or via the binary distribution, *many commands with the same name as those in the /bin directory will be created*. Therefore, an important consideration for using this open source software is the relationship of `/usr/local/bin` to `/bin`.

If `/usr/local/bin` is *before* `/bin`, the open source software commands will be picked up first. The files that collide can be interrogated using the following command:

```
$ for i in *
> do
>   ls /bin/$i 2>/dev/null
> done
/bin/ctags
/bin/ex
/bin/find
/bin/make
```

```
/bin/man  
/bin/mcp  
/bin/patch  
/bin/xargs  
/bin/zcat
```

We recommend that `/usr/local/bin` (or wherever you installed the code) come before `/bin` so GNU `make` will be picked up over `/bin/make`, but this choice must be considered carefully.

Chapter 4. Open source software packages

This chapter discusses open source software from a general point of view: packaging, building, contributing, and so forth.

It is unusual to get precompiled (binary) open source programs for OS/390, so do not expect anything other than source packages. On other operating systems, most notably GNU/Linux, Solaris, and so on, there are complete binary distributions.

Our project is an exception, as the CD-ROM in this redbook includes compiled binaries for many important tools. It is mainly for people with no C/C++ compiler or no time to go into details. We tried to keep these binary packages useful and up to date, as well as reliable and efficient.

Binary distributions - pros and cons

There are arguments for and against getting and installing binary distributions. The main argument for using binary distributions is that the executables are already built, so they are faster and easier to use.

Be forewarned that installing even precompiled binary packages is very different from getting and installing a new program product from the usual IBM support channel. Even if it runs as smoothly and as fast as a software installation on a PC, some thought should be given to the disadvantages of self-installing, prepackaged binaries.

Bear in mind that open software packages are released much more quickly than program products. New releases come days or even hours after a bug has been found, and this is not unusual. But they are normally released as *tarballs* first, and the prepackaged binaries follow days, weeks, even months later.

A current binary package is typically at least a couple of minor versions behind the latest *tarball*. So if you wish to keep up with all the “bleeding edge” software, you may not want to wait for binaries to be released.

Another issue is that the tarball package may well be more complete, have more options, and lend itself better to customization and localization. The binary version may be missing some of the functionality of the full release.

Also, some prepackaged binaries will not properly install, and even if they do install, they could crash. They may depend on library versions different from those present in your system, or they may be improperly prepared or just

plain broken. In any case, you inevitably rely on the expertise of the persons who have packaged it.

Finally, it helps to have the source code on hand, in order to be able to modify it according to your needs and to learn from it.

For more on this topic and for a comprehensive guide to building and installing *generic* UNIX software distributions, especially under GNU/Linux, see the document *Building and Installing Software Packages for Linux*, by Mendel Cooper on the Internet at:

<http://personal.riverusers.com/~thegrendel/build.html>

4.1 What you find in open source packages

After downloading a package from the Internet, uploading to the host, and unwinding, you will probably find something similar to this in open source packages:

```
$ ls ~/src/groff/s390build
BUG-REPORT  INSTALL  PROBLEMS  TODO      configure  man
COPYING     Makefile.in  PROJECTS  VERSION   configure.in  src
ChangeLog   NEWS       README    alocal.m4  doc
```

All of these files are important, and we explain their uses in the following sections.

If you want to go on with your port very quickly, you can start with the `README` file (the packages you find on the CD-ROM in this redbook have an additional `README.OS390`, which gives specific details on this platform). The `README` explains to you what the package is for and gives basic information about the contacts you can use if you get into trouble.

It is also important to check the license of the package (you will often find it in `COPYING`). This file is usually a copy of the GNU General Public license (see Appendix A, “The GNU General Public License” on page 289), or its equivalent.

Then you can proceed to the `INSTALL` file. It will tell you the prerequisites of the installation, which libraries and other packages should be previously installed, and so on. It will also list your options, including or leaving out some optional parts or user interfaces (most notably the X Window system). Additionally, it can help you avoid the installation of unwanted or complex prerequisites.

4.2 Overall source structure

This section is aimed at readers knowledgeable about using the C language and the UNIX operating system.

The source code for the package generally attempts to adhere to industry standards wherever possible. This means that the package uses library routines that are specified by the ANSI C standard and by the POSIX operating system interface standard.

Using conditional compilation, the C/C++ programs may contain a great deal of machine- or operating system-dependent code. The program source code is supplied in the form of program and header files (program files with the extensions .c or .CC, sometimes .cc or .cpp; header files with .h).

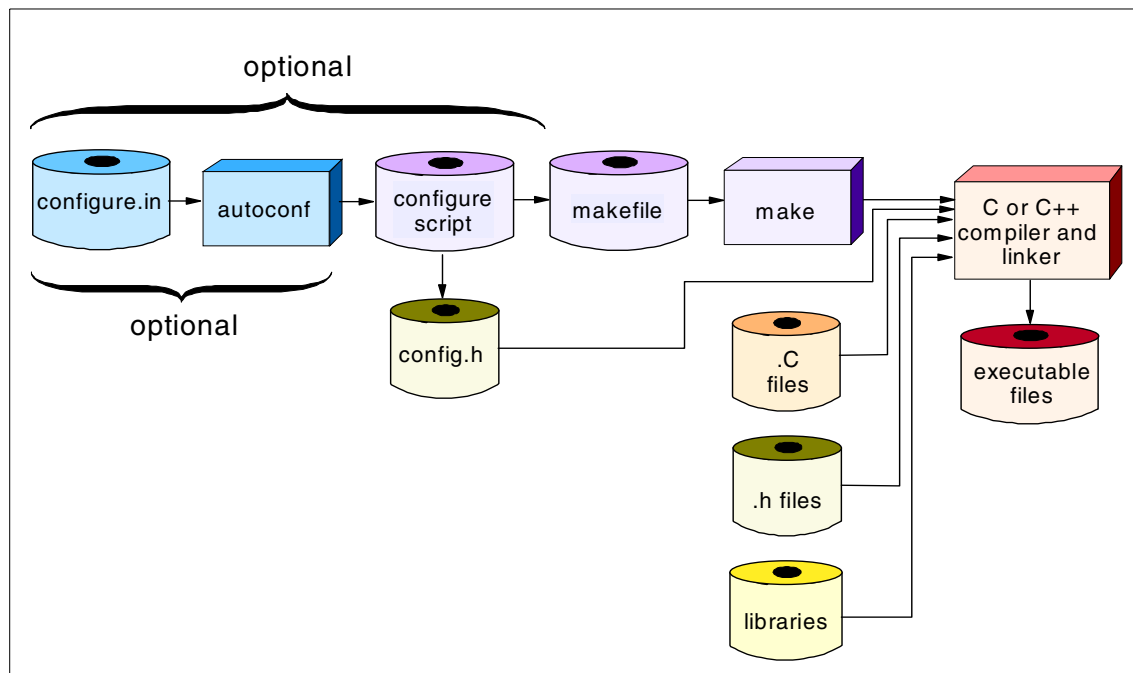


Figure 2. Overall build process

The packages have a “main” part, containing “programs”, the .c, .h files, libraries, and various other sources such as macro definitions for m4, yacc/lex syntax descriptions, Lisp programs, and so on. There are also “control” files which manage and control the building process; refer to Figure 2.

Even though the C/C++ language is highly portable, each UNIX system is a bit different. The build therefore should be done in an intelligent way, if you want it to be done automatically.

4.3 Configuring and compiling

The central element in configuring and compiling packages is the (often huge) configure script. It executes a full series of small tests or questions, each of which reveals some fact about the compilation environment. The questions are:

- Can the C program assume having a given piece of functionality implemented (for example, random numbers or some math function), or should it use its own implementation (which can be less efficient on the given platform)?
- Which type of compiler is given, and therefore which set of compilation flags are to be used?

These tests are prepared and specified by the authors of the package, and they are put together to form the configure script. This is, of course, an error-prone and difficult process. However, it can be automated by the autoconf-like programs. You find an excellent discussion of the details in Chapter 6, “Autoconf” on page 43 and Chapter 23, “Ittools” on page 117.

Note: If you encounter problems with the configure script, save time and effort by regenerating it from the sources, instead of trying a “quick-fix” by editing it.

You can find a useful cookbook, *Using GNU configuration tools*, by Dirk Vermeir on the Internet at:

<http://tin2.vub.ac.be/~dvermeir/manuals/autocookbook/cookbook.html>

4.3.1 The configuration process

The aim of the configure script is to output a new header file `config.h` (in order to control the conditional compilation), and a makefile (to control and execute the compile/build job).

Most of the open source packages come with a quality configure script, which usually runs without any problems.

For the simplest and quickest build you need only the following commands:

```
./configure
make all
su
make install
```

Your success depends on many factors, such as whether the package was carefully built by the maintainer, whether your system has all the necessary compilers and header files installed, and so on.

If a problem occurs when running the configure script, check the INSTALL file first. It usually tells you which are the optional parts or libraries for the package. You may want to get help from the script itself, too, as follows:

```
./configure --help
Usage: configure [options] [host]
Options: [defaults in brackets after descriptions]
Configuration:
  --cache-file=FILE      cache test results in FILE
  --help                 print this message
...
Host type:
  --build=BUILD           configure for building on BUILD [BUILD=HOST]
  --host=HOST             configure for HOST [guessed]
  --target=TARGET         configure for TARGET [TARGET=HOST]
Features and packages:
  --disable-FEATURE       do not include FEATURE
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)
  --x-includes=DIR        X include files are in DIR
  --x-libraries=DIR       X library files are in DIR
```

Sometimes you have to find out how some special environment variables are used by a package. For example, there is often a RANLIB variable, set to the value `ranlib`, and the makefile will try to simply execute `$RANLIB`, which is then missing¹. You can safely replace `ranlib` with `echo` by defining the variable `RANLIB=echo`, or in the worst case, by editing the makefile.

¹ The GNU `ranlib` generates an index to the contents of an archive and stores it in the archive. An archive with such an index speeds up linking to the library and allows routines in the library to call each other without regard to their placement in the archive. The GNU `ranlib` program is another form of GNU `ar`. It means if you do not have GNU `ar` (which is our case on OS/390) you can safely replace `ranlib` with a do-nothing command, for example with `echo`.

You may consider defining the following variables in your profile:

```
export MYFLGS="-O2 -D_ALL_SOURCE -DIS_EBCDIC_HOST"
export _CXX_CCMODE=1
export _CC_CCMODE=1
```

Use the environment variable MYFLGS in the following way:

```
CC=cc CFLAGS=-DSOME_EXTRA_FLAGS" $MYFLGS" CCC=c++ CCFLAGS="-+"
RANLIB=echo ./configure --host=$MACHTYPE
```

The environment variable \$MACHTYPE is automatically set on OS/390 by bash to i370-ibm-mvs. However, it is better to define it as s390-ibm-os390; see the Alias definitions in 6.6, “Porting autoconf to OS/390” on page 46. See also the discussion of problems we encountered during the porting of various packages in 50.2, “Problems encountered” on page 276.

The config.h header file, created by the configure program, contains definitions that describe features of the operating system you’re using. For the most part, it lists which standard subroutines are (or are not) available. For example:

```
/* Define if you have the <termio.h> header file. */
/* #undef HAVE_TERMIO_H */
/* Define if you have the <termios.h> header file. */
#define HAVE_TERMIOS_H 1
/* Define if you have the <time.h> header file. */
#define HAVE_TIME_H 1
```

The config.h file also defines constants that describe facts about your variant of UNIX.

```
/* Sizes of various buffers. */
#define CMDBUF_SIZE 512 /* Buffer for multichar commands */
#define UNGOT_SIZE 100 /* Max chars to unget() */
#define LINEBUF_SIZE 1024 /* Max size of line in input file */
#define OUTBUF_SIZE 1024 /* Output buffer */
```

Conditionally compiled code in the other source files relies on the definitions in the config.h file.

Besides creating config.h, configure produces a makefile from the file Makefile.in. There are a number of lines in Makefile.in that are system- or feature-specific; most prominently, the parameter definition for your compiler call:

```
CFLAGS =-O2 -D_ALL_SOURCE -DIS_EBCDIC_HOST
```

Another part of the compile command describes which macro names should be defined for this compilation:

```
# -DLONG_FOR_TIME_T          if localtime() takes a long * not a time_t *
# -DHAVE_STRUCT_EXCEPTION    if <math.h> defines struct exception
# -DRETSIGTYPE=int           if signal handlers return int not void
DEFINES= -DHAVE_UNISTD_H=1 -DHAVE_DIRENT_H=1 -DHAVE_LIMITS_H=1
-DHAVE_STDLIB_H=1 -DHAVE_CC_LIMITS_H=1 -DRETSIG_TYPE=void
-DHAVE_UNISTD_H=1 -DHAVE_GETPAGESIZE=1 -DHAVE_FMOD=1 -DHAVE_STRTOL=1
-DHAVE_GETCWD=1 -DHAVE_STRERROR =1 -DHAVE_PUTENV=1 -DHAVE_RENAME=1
-DHAVE_MKSTEMP=1 -DHAVE_STRCASECMP=1
```

If you must edit config.h, you need to determine which library routines you do have and which you do not have (consult your system documentation). In the worst case, simply do not define any of the macros for missing subroutines. When you compile the package, the final link-editing step will fail. The link editor will provide you with a list of unresolved external references, and these are the missing subroutines. Edit config.h again and recompile, and you should be set.

4.3.2 Modifying the configure script

The structure of the configure script depends heavily on the configure.in file used in the package building process and the version of the `autoconf` program used by the package maintainer.

Sometimes you have to edit the script, even though the first line of the makefile contains the following:

```
# This file was generated automatically by configure. Do not edit.
```

This line simply means that it was *generated* by `autoconf`, and the next time you run `autoconf`, it will be overwritten.

4.4 Troubleshooting, reporting problems, and bugs

Arriving at a correct configuration is likely to be an iterative process, so don't be discouraged if it takes you several tries. However, if you have no luck whatsoever, it is suggested you report your system type, and the steps you took, to the maintainers. Once you do have a working configuration, it is useful to send it back to the maintainers, because support for your system can be added to the official release.

One of the files you should find in an open source software package is named `BUG-REPORT`. If you feel you have found a legitimate bug, you can use this file to contact the maintainers of the package.

Many of the packages in this redbook have been ported to tens (if not hundreds) of different systems, and there is continuously ongoing development. You can communicate your comments and ideas to the open community by sending an e-mail to the maintainer (you can find the address in the README file), or by joining a mailing list (also found in the README file). Usually there are mailing list archive addresses, too, where you can browse for your questions; or you can use the Power Search at:

http://www.deja.com/home_ps.shtml

4.5 Improving package performance on OS/390

In the brief course of this redbook project, we had practically no time to improve or optimize the programs, so the program quality may be at any level.

In contrast, if you have time to invest for improving performance in the OS/390 environment, you will find recommendations given in the *Porting Guide*, a Web document available at:

<http://www.s390.ibm.com/oe/bpxalpor.html>

4.6 Modifying the source code

Even if you install the programs in this redbook from the CD-ROM in binary, you might want to also obtain the source code of the programs. Why? This will allow you to change our modifications or maybe improve the code. It will also allow you to control any potential security exposures.

As previously discussed, you can examine the source code for problems and share your corrections, modifications and improvements.

4.7 Sharing your work

Important motivations for people participating in the open source movement are getting credit for their (mostly or totally) unpaid work, and “giving back” or sharing their work. Having the gratifying feeling that they’ve served the community is one of the major forces driving the open source software movement.

Chapter 5. Apache

This chapter discusses the Apache open source software package.

5.1 Introduction to Apache

Apache is an http server based on the National Center for Supercomputing Applications (NCSA) Web server. It has been developed by the members of the Apache Project. The target for this group is to provide a robust and freely available Web server with many features, which fulfills the requirements for private as well as for commercial users.

Development was started in 1995 by Rob McCool at the National Center for Supercomputing Applications, University of Illinois, Champaign-Urbana. The effort was quickly joined by a number of core contributors, who formed the foundation of the Apache Group. They not only fixed existing bugs and added new features, they even did a general overhaul and redesign of the complete code base.

Today, Apache is probably the most widely used Web server on the market (see <http://www.netcraft.com/survey/>). Its main strength is that it is freely available. Unlike a purchased software package, users feel responsible for improving and enhancing the product. More information can be found in the README and About_apache file in the Apache package.

All OS/390-related changes have been integrated into the mainstream of the Apache 1.3.19 server. You can use either the normal ASCII source distribution or you can download a binary distribution for a quick start.

On the mainframe this is an alternative to the Domino Go Web Server, especially for organizations which decided to use the Apache Web servers as their standard. They can now use the same APIs on all platforms and they can keep a common httpd configuration file.

According to our testing, the current implementation of Apache on OS/390 still has some performance disadvantages compared with the HTTP server, and it does not provide tight integration into the mainframe security system. Some of this will change with the next release of Apache.

5.2 Installation of Apache

Follow the instructions in this section to install the Apache package. The version of Apache on the included CD is 1.3.22, however, an installation of

version 1.3.19 is described. The installation of the two versions is nearly identical.

5.2.1 Step 1: Load the product

The package is a gzip file containing EBCDIC source code, so you'll need the GNU zip utility gunzip. Before you can install Apache, see Chapter 20, "Gzip" on page 105. Note that unlike most other packages, this tar file does not need to be translated; it's already in EBCDIC.

```
/usr/local/apache>ls
apache_1.3.19-OS390.tar.gz
/usr/local/apache>gunzip apache_1.3.19-OS390.tar.gz
/usr/local/apache>pax -rf apache_1.3.19-OS390.tar
/usr/local/apache>ls
apache_1.3.19  apache_1.3.19-OS390.tar
/usr/local/apache>cd apache_1.3.19
/usr/local/apache/apache_1.3.19>ls
ABOUT_APACHE      LICENSE              README.configure
config.layout       logs
Announcement       Makefile.tmpl       WARNING-NT.TXT
configure           src
INSTALL            README              cgi-bin             htdocs
KEYS               README.NT           conf                icons
/usr/local/apache/apache_1.3.19>
```

The product on the CD is a binary distribution, so you can skip the next two steps of the installation and go directly to 5.2.4, "Step 4: Install the product" on page 36. However, you can follow the next steps if you want to have a customized version for your system.

5.2.2 Step 2: Configure the product

You can find detailed information about which configuration parameters can be changed in the README.configure and the INSTALL file in the product directory. The main one is the directory PREFIX, in which you want to install the Apache software. This can be directly specified as a parameter on the configure command. A `./configure --show-layout` command will display the actual directory structure for your installation without actually starting the process.

```
/usr/local/apache/apache_1.3.19>./configure
--prefix="/usr/local/apache"
Configuring for Apache, Version 1.3.19
+ using installation path layout: Apache (config.layout)
Creating Makefile
Creating Configuration.apaci in src
```

```

Creating Makefile in src
+ configured for OS390 platform
+ setting C compiler to c89
+ setting C pre-processor to c89 -E
+ checking for system header files
+ adding selected modules
+ checking sizeof various data types
+ doing sanity check on compiler and options
Creating Makefile in src/support
Creating Makefile in src/regex
Creating Makefile in src/os/os390
Creating Makefile in src/ap
Creating Makefile in src/main
Creating Makefile in src/lib/expat-lite
Creating Makefile in src/modules/standard

```

5.2.3 Step 3: Compile and link the product

This process is done by simply entering the `make` command to execute the Makefile previously prepared by the configure process.

```

/usr/local/apache/apache_1.3.19>make
====> src
make[1]: Entering directory `/usr/local/apache/apache_1.3.19'
make[2]: Entering directory `/usr/local/apache/apache_1.3.19/src'
====> src/regex
make[3]: Entering directory `/usr/local/apache/apache_1.3.19/src/regex'
sh ./mkh -i _REGEX_H regex2.h regcomp.c regerror.c regexec.c regfree.c
> ../include/hsregex.h
sh ./mkh -p regcomp.c >regcomp.i
c89 -I. -I../os/os390 -I../include -DOS390 -DCHARSET_EBCDIC
-D_ALL_SOURCE -DUSE_HSREGEX -DUSE_EXPAT -I../lib/expat-lite
-DNO_DL_NEEDED `../apaci` -DPOSIX_MISTAKE -c -o regcomp.o regcomp.c
..
..
c89 -DOS390 -DCHARSET_EBCDIC -D_ALL_SOURCE -DUSE_HSREGEX -DUSE_EXPAT
-I../lib/expat-lite -DNO_DL_NEEDED `../apaci` -o ab -L../os/os390
-L../ap ab.o -lap -los
sed <apxs.pl >apxs \
-e 's%@TARGET@%httpd%g' \
-e 's%@CC@%c89%g' \
-e 's%@CFLAGS@% -DOS390 -DCHARSET_EBCDIC -D_ALL_SOURCE -DUSE_HSREGEX
-DUSE_EXPAT -I../lib/e
xpat-lite -DNO_DL_NEEDED `../apaci`%g' \
-e 's%@CFLAGS_SHLIB@%g' \
-e 's%@LD_SHLIB@%g' \
-e 's%@LDFLAGS_MOD_SHLIB@%g' \

```

```

        -e 's%@LIBS_SHLIB@%'&& chmod a+x apxs
make[2]: Leaving directory
`/usr/local/apache/apache_1.3.19/src/support'
<=== src/support
make[1]: Leaving directory `/usr/local/apache/apache_1.3.19'
<=== src

```

5.2.4 Step 4: Install the product

This process will differ slightly depending on whether you did a complete source installation or want to install the binary distribution. For the binary distribution follow the instructions in the README.os390 file and execute the `./INSTALL.bindist` script instead of doing the `make install`. The end result will be the same, the only difference being that the `make install` will need some information from the previous `./configure` process, which is not performed during a binary installation.

```

/usr/local/apache/apache_1.3.19>make install
make[1]: Entering directory `/usr/local/apache/apache_1.3.19'
===> [mktree: Creating Apache installation tree]
./src/helpers/mkdir.sh /usr/local/apache/bin
mkdir /usr/local/apache/bin
./src/helpers/mkdir.sh /usr/local/apache/bin
./src/helpers/mkdir.sh /usr/local/apache/libexec
mkdir /usr/local/apache/libexec
./src/helpers/mkdir.sh /usr/local/apache/man/man1
mkdir /usr/local/apache/man
mkdir /usr/local/apache/man/man1
..
..
./src/helpers/install.sh -c -m 644 ./conf/magic
/usr/local/apache/conf/magic
<=== [config]
make[1]: Leaving directory `/usr/local/apache/apache_1.3.19'
+-----+
| You now have successfully built and installed the |
| Apache 1.3 HTTP server. To verify that Apache actually |
| works correctly you now should first check the |
| (initially created or preserved) configuration files |
| |
| /usr/local/apache/conf/httpd.conf |
| |
| and then you should be able to immediately fire up |
| Apache the first time by running: |
| |
| /usr/local/apache/bin/apachectl start |
+-----+

```

5.2.5 Heart of the Apache.install script

For quick reference, following is the heart of the install script named `apache.install`, which is included on the CD (where `$installDir` is typically `/usr/local`):

```
gzip -c -d apache_1.3.19-OS390.tar.gz| pax -r
ln -s apache_1.3.19 apache
cd apache
./configure --prefix=$installDir
make
make install
```

5.3 Customization and operation of Apache

This section discusses various aspects of using Apache.

5.3.1 Modifying the server configuration `conf/httpd.conf` file

Before you actually start the Apache Web server, you should review the `httpd.conf` file in the `conf` directory. Even if the Web server is started as a superuser (UID 0), it should never run with this high authority and it will automatically switch to another userid in the startup process.

By default this userid is *nobody*, with an OMVS group ID of (-1). Both will not exist in your environment and you should change the `httpd.conf` file accordingly.

5.3.2 RACF setup for the Apache server

Most installations have decided to define the RACF facility class `BPX.DAEMON`. This provides the highest possible integration of the UNIX System Services into the OS/390 security environment. As soon as this is defined in RACF, every UNIX program which wants to use protected commands, such as `setuid` and `setgid`, requires `READ` access to this facility class. The Apache Web server will use these commands during the startup process to switch to a non-superuser ID after establishing the TCP/IP connection.

You have to enter the following RACF commands to set this up:

```
permit BPX.DAEMON CLASS(FACILITY) ID(APACHE) ACCESS(READ)
```

```
setropts raclist(facility) refresh
```

In addition, you have to declare the Apache programs as *program controlled* and *apf authorized*. You can do that with the following shell commands:

```
/usr/local/apache/apache_1.3.19>cd /usr/local/apachi/bin
/usr/local/apachi/bin>extattr +ap *
/usr/local/apachi/bin>ls -E
total 3656
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP   98304 Mar 29 13:31 ab
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP    7065 Mar 29 13:31 apachectl
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP   20487 Mar 29 13:31 apxs
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP    6456 Mar 29 13:31 dbmmanage
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP   77824 Mar 29 13:31 htdigest
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP  131072 Mar 29 13:31 htpasswd
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP 1417216 Mar 29 13:31 httpd
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP   61440 Mar 29 13:31 logresolve
-rwxr-xr-x  aps  1 AAAAAAA OMVSGRP   45056 Mar 29 13:31 rotatelog
```

The `ap` flags indicate that the programs are now running in the correct state. Keep in mind that you are not authorized to use the `extattr` command by default. Not even being a superuser is enough for this. You have to define the following RACF facility classes and you need the proper access to them:

```
RLIST FACILITY BPX.FILEATTR.PROGCTL ALL
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(EGONT) ACCESS(READ)
RLIST FACILITY BPX.FILEATTR.APF ALL
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) ID(EGONT) ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH
```

Note that the user ID EGONT in the commands is only an example; you have to enter your personal RACF ID or the userid of your system administrator who set up the environment for Apache.

5.3.3 Starting the Apache server

You can start the server by going to the `./bin` directory and issuing the command:

```
/usr/local/apache/bin>apachectl start
```

Afterwards, you can access the server from every Web browser in your TCP/IP environment. The initial INDEX page is in the `./htdocs` directory of the installation path. You can replace it with your own file any time.

You will also find a link to a complete set of documentation which is now available on your server.

If you have problems accessing the server, there are two places you can get information about the current situation:

1. Check what tasks are running via the `ps` command.

```
EGONT 83887352 16778494 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
EGONT 50332922 16778494 - 14:01:59 ? 0:00 /usr/local/apachi/bin/httpd
EGONT 16778491 16778494 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
EGONT 33555708 16778494 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
EGONT 16778493 16778494 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
AAAAAAA 16778494 1 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
EGONT 50332928 16778494 - 14:01:38 ? 0:00 /usr/local/apachi/bin/httpd
MIKEM 83887362 853 - 14:06:39 ttyp0001 0:01 sh -L
AAAAAAA 83887402 50332527 - 14:06:42 ttyp0002 0:07 ps -ef
```

You should find a number of tasks running the `httpd` program waiting for Web requests. By default there are five tasks defined, but you can change this in the `httpd.conf` file and it will be automatically adjusted to a maximum value if there is more load on your system.

2. Check a task listening on the HTTPD port 80 via the `netstat` command.

```
/usr/local/apache>netstat
MVS TCP/IP netstat CS V2R7          TCP/IP Name: TCPIPOE          14:10:32
User Id Conn Local Socket          Foreign Socket          State
-----
BPXOINIT 00019 0.0.0.0..10007          0.0.0.0..0             Listen
DBH1DIST 001C5 9.12.2.26..33367        0.0.0.0..0             Listen
DBH1DIST 001C4 0.0.0.0..33366          0.0.0.0..0             Listen
DB2QDIST 00026 9.12.2.26..33323        0.0.0.0..0             Listen
DB2QDIST 00025 0.0.0.0..33322          0.0.0.0..0             Listen
EGONT9 00693 0.0.0.0..80           0.0.0.0..0             Listen
FTPDOE1 00016 0.0.0.0..21             0.0.0.0..0             Listen
```

In this example, it's the server task running on ID `EGONT9` listening on the Web serving port.

5.3.4 Stopping the Apache server

You can stop the server by going to the `./bin` directory and issuing the following command:

```
/usr/local/apache/bin>apachectl stop
```

5.3.5 Running CGI programs

You can find two example CGI programs as part of the software distribution. The default configuration file `httpd.conf` already contains the necessary setup parameters, but you have to do some customization on the file modes and in the CGI programs.

The programs are in the /usr/local/apache/cgi-bin directory.

```
EGONT @ SC69:/usr/local/apache/cgi-bin>ls -la
total 48
drwxr-xr-x  2 STC      SYS1      8192 Mar 29 15:08 .
drwxr-xr-x 13 EGONT    SYS1      8192 Mar 29 15:08 ..
-rwxr-xr-x  1 STC      SYS1      274  Mar 29 16:26 printenv
-rwxr-xr-x  1 STC      SYS1      757  Feb 24 13:04 test-cgi
```

Be sure that the userid you specified in the httpd.conf file has execution rights for the cgi programs in the HFS file system.

The test-cgi is a simple shell script that will work without any modification:

http://wtsc69oe/cgi-bin/test-cgi

CGI/1.0 test script report:

argc is 0. argv is .

```
SERVER_SOFTWARE = Apache/1.3.19 (OS390)
SERVER_NAME = wtsc69oe.itso.ibm.com
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
REQUEST_METHOD = GET
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
image/png, */*
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
REMOTE_HOST =
REMOTE_ADDR = 9.12.2.104
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```

The printenv is a Perl script. To use it, you first have to change the initial line to let the system know where to find your Perl program. In our case it is: #!/usr/local/bin/perl; this can be different in your location.

Then you can call it with the command:

http://wtsc69oe/cgi-bin/printenv

DOCUMENT_ROOT="/usr/local/apache/htdocs"


```

GATEWAY_INTERFACE="CGI/1.1"
HTTP_ACCEPT="image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
image/png, */*"
HTTP_ACCEPT_CHARSET="iso-8859-1,*,utf-8"
HTTP_ACCEPT_ENCODING="gzip"
HTTP_ACCEPT_LANGUAGE="en"
HTTP_CONNECTION="Keep-Alive"
HTTP_HOST="wtsc69oe"
HTTP_USER_AGENT="Mozilla/4.61 [en] (WinNT; U)"
PATH="/usr/local/bin:/usr/lpp/java18/J1.1/bin:/usr/lpp/Printsrv/bin:/bin:."
QUERY_STRING=""
REMOTE_ADDR="9.12.2.104"
REMOTE_PORT="4568"
REQUEST_METHOD="GET"
REQUEST_URI="/cgi-bin/printenv"
SCRIPT_FILENAME="/usr/local/apache/cgi-bin/printenv"
SCRIPT_NAME="/cgi-bin/printenv"
SERVER_ADDR="9.12.2.32"
SERVER_ADMIN="you@your.address"
SERVER_NAME="wtsc69oe.itso.ibm.com"
SERVER_PORT="80"
SERVER_PROTOCOL="HTTP/1.0"
SERVER_SIGNATURE="<ADDRESS>Apache/1.3.19 Server at
wtsc69oe.itso.ibm.com Port 80</ADDRESS>\n"
SERVER_SOFTWARE="Apache/1.3.19 (OS390)"
TZ="EST5EDT"
UNIQUE_ID="OONpYwkMAiAFAABAEz0"

```

These two examples should help you to write your own cgi scripts for your installation. For more information about Perl see Chapter 35, “Perl 5.005p3” on page 189.

5.4 License

The Apache Web server is distributed under the Copyright (c) 1995-1999 of the Apache Group. Details are available in the LICENSE file as part of the software distribution.

In general it is free for everybody to use, but the distribution and usage of the name of the server is still attached to a number of conditions and approvals set by the Apache Group.

5.5 Getting Apache from the Internet

You can download the normal ASCII source distribution, as well as the binary distribution for OS/390, from the Internet. The binary distribution can be found at:

<http://www.apache.org/dist/binaries/os390/>

The normal ASCII source file can be found at:

<http://www.apache.org/dist/>

You can perform the complete source installation process (including configure/make/make install) with both distributions. Keep in mind the ASCII-to-EBCDIC translation option during the pax process if you decide to use the normal Apache file.

The distribution file contains a mixture of binary and ASCII files; therefore, you have to perform a selective translation. To assist you in this process you can download the tool “ext” from the UNIX System Services Tools and Toys page at:

<http://www.s390.ibm.com/oe/bpxalty2.html>

After doing a complete load with the translate option it will check the individual file names and reload all binary files without translation. In the case of Apache, these are all gif files which are distributed as part of the software package.

5.6 Porting Apache to OS/390

All changes to Apache are fully integrated into the mainline of the code. There are no additional changes necessary to run it on OS/390.

Chapter 6. Autoconf

This chapter discusses the autoconf open source software package.

6.1 Introduction to autoconf

The role of autoconf is to create a *configure* shell script, which is capable of determining many facets of the machine/system configuration on which it runs, and then propagating such settings into other files—typically one or more Makefiles or config.h C/C++ headers.

6.2 Installation of autoconf

Your OS/390 UNIX system must have the following prerequisite packages present:

- Perl5.004 or higher
- GNU m4 1.4
- GNU make 3.76.1 or higher

Decide where you want the binary executable files and supporting data files installed. In this example, we have decided that the root directory for autoconf is /usr/local¹, which will result in the following directories being created by the installation procedure, if necessary:

/usr/local/bin <- Executables go here

/usr/local/info <- Info files go here²

/usr/local/share/autoconf <- m4 macro files go here

Next, you are ready to configure the package:

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

Now build and install the package:

```
make install
```

¹ It happens that /usr/local is the default assumed by autoconf. If your choice differs, you *must* provide the full pathname to the base installation directory with the 'configure' --prefix= option.

² The *.info files would go in this location; however, we don't generate any in the OS/390 install.

6.3 Customization and operation of autoconf

There are two ways to use autoconf:

1. Porting an existing package to a platform for the first time
2. Creating a new portable package

When porting an existing package, you may or may not need to re-run autoconf. If you are porting a package *to* OS/390, we recommend that you copy the ported versions of `config.guess` and `config.sub` to the current directory, then run the `autoconf` command to regenerate the configure script. In this way, you will be guaranteed a script that conforms to all EBCDIC conventions. If you fail to do this, the configure script that came with the package *may* fail in `sed`, depending on which autoconf macros are present in `configure.in`.

The goal of autoconf is to produce a configure script, which the package installer runs in order to generate the proper files (such as `Makefile` and/or `config.h`, as well as possibly other files, depending on how `configure.in` was written), prior to actually building and installing the package. If you followed the installation instructions, you've already worked with an example of an autoconf-generated configuration script. As you saw, the next step was to run `make`.

When creating a new portable package, you'll write a `configure.in` template from scratch, or use the `autoscan` command to generate a generic `configure.in` template. If you use `autoscan`, you'll doubtlessly end up editing the generated `configure.in` template to match your exact needs.

The `configure.in` template consists of autoconf m4 macro calls, which begin with the prefix “AC_”. There are approximately 210 such macros provided with autoconf; far too many to list here. Refer to the documentation in the autoconf package for complete descriptions of all the macros.

The `configure.in` template *must* begin with a call to `AC_INIT`. If you are generating a `config.h` (or similar) header file, a call to `AC_CONFIG_HEADER` must immediately follow. The `configure.in` template must also end with an `AC_OUTPUT` call; this is the point at which the output files (usually `Makefile` and/or `config.h`) are created. Between the required beginning and ending m4 macro calls, you may place other m4 macro calls or open shell code, including the definitions and calls of your own custom m4 macros, should you want or need to.

The creation of m4 macros via the `AC_DEFUN` macro is covered in detail in the documentation in the autoconf package. However, the best source of

autoconf coding techniques is other people's `configure.in` and `Makefile.in` files. A couple of examples can be found in the CD distribution itself, as autoconf is used to configure the autoconf package. Other packages included on the CD are also configured with autoconf.

Typically, you'll use autoconf to create a `Makefile`. This will require that you also create a `Makefile.in` template file, which otherwise looks like a normal `Makefile` (variables, targets, and rules) except that the items that you wish to have autoconf substitute (based on its findings) are surrounded with '@' characters and are the names of autoconf variables. For example, the following suffix rule uses whatever autoconf determined your net CFLAGS variable should be:

```
.C.o :  
    $(CC) @CFLAGS@ -o $@ $<
```

You can use the autoconf substitution syntax anywhere you deem necessary in your `Makefile.in`. All you need to ensure is that autoconf properly sets the variable to which you refer. If you haven't done so, autoconf will complete without complaint, but your resulting attempts to use the `Makefile` are likely to fail and result in various cryptic errors. If you suspect this kind of error, look for unresolved @<symbol>@ tokens in your final `Makefile`.

Those who need to understand the details of autoconf operation are referred to the GNU documentation provided with the package. Unfortunately, these documents are in GNU (emacs) texinfo format, and neither the info nor the TeX packages have been ported to OS/390. A printed book form of the texinfo documentation can be obtained by converting the file `autoconf.ps` to ASCII ('type ascii' ftp transfer to an ASCII machine is highly recommended), then spooling it to a PostScript printer. The file `autoconf.ps` is in the same directory from which you untarred the distribution tar file.

6.4 License

The terms of use and redistribution of autoconf are governed by the GPL. See Appendix A, "The GNU General Public License" on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

6.5 Getting autoconf from the Internet

You can download autoconf from the GNU Web site starting at:

<http://www.gnu.org/software/autoconf/autoconf.html>

Click **Downloading Autoconf**.

6.6 Porting autoconf to OS/390

The first order of business was to remove all “ASCII-isms” from the supporting Perl programs and shell scripts, particularly the rather annoying existence of hard coded ASCII newline octal tokens (\012) in `tr` commands. All such occurrences were replaced with the `\n` token. According to the ChangeLog, the `\n` tokens were replaced with the octal \012 tokens in 1992 owing to some unspecified portability issue. This port reverses that action, since replacing the \012 tokens with '\025' (the EBCDIC newline) would have been hopelessly unportable.

The next order of business was to have the System/390 operating systems (OS/390, VM/CMS, TPF) recognized and possibly even guessed at, if possible. This impacts two very important autoconf portability macros: `AC_CANONICAL_HOST` and `AC_CANONICAL_SYSTEM`. The shell script `config.guess` was modified to read the output of the OS/390 UNIX System Services `uname` command and interpret it as the canonical system name `s390-ibm-os390`. The shell script `config.sub` was modified to allow the following canonical system names to pass:

Alias	Full name
s390-linux	s390-ibm-linux-gnu
mvs	i370-ibm-mvs
os390	s390-ibm-os390
i370-linux	i370-ibm-linux-gnu
vmcms	s390-ibm-vmcms
tpf	s390-ibm-tpf

Only the `s390-ibm-os390` system name can be successfully guessed. People writing later `configure.in` files based upon this port should use `s390-ibm-os390` to mean OS/390 UNIX System Services.

Either the aliases or the full canonical system names are permitted as operands to the `--host=`, `--build=`, or `--target=` options of the `configure` command.

The next order of business was to modify certain m4 macros to behave properly under OS/390. As a result, macro `AC_FUNC_VFORK` was modified to remove the resulting `CEEDUMP.*` file that occurs when a Sparc-only check for `fork()` parent-child data access test is run. Rather than remove the test, it was deemed more appropriate to remove the byproduct of the inevitable dump, which is presumed to occur because of an attempt to write to `fd -1`.

Macro `AC_PATH_XTRA`, which determines the expected command line compiler and linkage option flags to point to the X11 libraries, was changed as follows when OS/390 is determined as the target operating system:

1. `X_CFLAGS` now returns “-Wc,DLL”
2. `X_LIBS` now returns “-Wl,DLL” plus a string of *all* installed X11 definition side deck files (*.x)

The net effect of both of these changes is to force a dynamic link instead of a static link. Not only did static linkage fail certain linktime and runtime tests, but it also creates a horribly bloated final load module. Dynamic linkage works much better.

The last order of business was to provide for a `config.h`-resident signal that the character set in use is EBCDIC. A new macro, `AC_C_EBCDIC` was written to support this. This macro takes no operands, and causes the term “`#define EBCDIC 1`” to be inserted into `config.h` when `autoheader` is used *and* `AC_C_EBCDIC` is called.

For installation purposes on OS/390, which does not yet have working ports of the GNU texinfo and TeX packages, the `configure.in` and `Makefile.in` templates were changed to search for the `makeinfo` and `texi2dvi` programs. If they are not found, the commands for these programs are changed to `:` (do nothing) in the final `Makefile` produced by `configure`. By doing this, we assure that the installation `make` commands will always complete.

6.7 Caveats

The supported manner of identifying the system is to call `AC_CANONICAL_SYSTEM`. This requires that the following files (included with the distribution) be distributed with all `autoconf`-configured packages:

- `config.guess`
- `config.sub`
- `install-sh`

Chapter 7. Automake

This chapter discusses the automake open source software package.

7.1 Introduction to automake

The role of automake is to produce at least an autoconf-compatible Makefile template for completion by autoconf. Typically, the person creating an automake/autoconf package will create an automake template file (usually called Makefile.am) that identifies which end products are to be built and/or installed.

7.2 Installation of automake

Your OS/390 UNIX system must have the following prerequisite packages present:

- Perl5.004 or higher
- GNU m4 1.4
- GNU make 3.76.1 or higher
- Autoconf 2.13 or higher

Decide where you want the executable files and supporting data files installed. In this example, we have decided that the root directory for automake should be /usr/local¹, which will result in the following directories being created by the installation procedure, if necessary:

- /usr/local/bin <- Executables go here
- /usr/local/share/aclocal <- autoconf m4 macro extensions go here
- /usr/local/share/automake <- automake miscellaneous files go here

Now we can configure automake:

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

Now we can install and test automake:

```
make install
make check
```

The make check phase will put automake through a set of tests that take approximately five minutes to complete. All should pass except insthook.test,

¹ It happens that /usr/local is the default prefix established by autoconf. You need only express the --prefix=option of the autoconf command if your choice of installation base directory is different.

which contains a call to `grep -A2`. The `-A2` option is not supported by the XPG4.2 standard. The results of this test were manually isolated and verified.

7.3 Customization and operation of automake

Autoconf reads a `Makefile.am` file and produces a `Makefile.in` template from it. The `Makefile.am` file usually consists of only a few lines that call macros and/or define targets, and explodes that into a `Makefile.in` that is intended to be passed to `autoconf`. There are many options and `automake` macros that impact not only the contents of `Makefile.am`, but also the contents of `configure.in` (`autoconf`). The documentation (see below) should be consulted for a full description of options and `Makefile.am` targets.

Typically, the process of exploiting `automake` for a newly created package (one you're not porting) would include the following steps:

1. Create a `Makefile.am` file that describes the location of your source files and the intended output of your build (executables, documentation, or whatever else you think necessary).
2. Create a `configure.in` file that meets both `automake` and `autoconf` requirements. The `autoscan` command may be helpful as a starting point for this task. (The `autoscan` command is mentioned in 6.3, "Customization and operation of `autoconf`" on page 44, and is discussed in detail in the GNU `automake` documentation. This command attempts to guess, based on the source files present in a named directory, what the `configure.in` file may need to contain. Such files usually need manual editing.)
3. Run `aclocal`
4. Run `automake`
5. Run `autoconf`

For packages you're porting to OS/390 that are based on `automake`, you may wish to rerun `automake` and `autoconf` before running `configure`; this assures that you end up with a `configure` script that meets the basic requirements (for example, EBCDIC character-based `sed` snippets) for OS/390 operation.

In either case (creating or porting an `automake`-based package), `automake` will try to lay the groundwork for a final `Makefile` that conforms to a document known as *The GNU Makefile Standards Document*. Typically, this assumes the existence of several utilities which are not yet available under OS/390, and furthermore imposes standards on subdirectory names and roles. The final `Makefiles` tend to be long and difficult to understand without experience. However, these `Makefiles` are supported by `autoconf` more or less perfectly.

When considering whether or not to use automake in your new project, consider carefully the trade-off between a long Makefile and the degree to which automatic system configuration is necessary. Should it turn out that you don't need all of the automake-supplied details, it may be a wiser decision to merely base your Makefile creation on autoconf alone.

Also refer to Chapter 23, “Ittools” on page 117 for yet another alternative in automatic Makefile creation.

The documentation for automake is GNU info-based. Unfortunately, the GNU info package has not yet been ported to OS/390. In lieu of this, we have created an Adobe PostScript(tm) file from the original 'automake.texinfo' file and included it in this distribution. It is called automake.ps; it will be un-tarred into the same directory in which your tar file resides. This file is suitable for printing once you have converted it to ASCII; this can be accomplished by ftp-ing it to an ASCII machine with the 'type ascii' ftp option and spooling it to a PostScript printer from there.

7.4 License

The terms of use and redistribution of automake are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

7.5 Getting automake from the Internet

You can download automake from the GNU Web site starting at:

<http://www.gnu.org/software/automake/automake.html>

From there, click **Downloading Automake**.

7.6 Porting automake to OS/390

This is a very simple port, which consists of the following steps:

1. Remove the sole “ASCII-ism”: a `tr \012` command in `depend2.am`. This was changed to `tr \n` for compatibility with OS/390.²

2. Replace the shell scripts `config.guess` and `config.sub` with the versions from the OS/390 port of `autoconf`.³
3. Modify `Makefile.am` to remove 'gnits' from the `AUTOMAKE_OPTIONS` and add 'no-installinfo' that is 'foreign' to it. This permits the `make uninstall` target to run to completion without a call to `install-info`, which is a part of the `texinfo` distribution and not available under OS/390 yet.
4. Rerun `autoconf` against the unmodified `configure.in` file to generate a `configure` script that can run on OS/390 UNIX.
5. Rerun `automake` against the modified `Makefile.am` file to correct the installation `Makefile.in` template.
6. Rerun `automake` again to generate the proper OS/390 `Makefile` for `automake` installation.

² This is the same change that was made in several of the `automake` parts. The background behind this is that not all UNIX implementations of `tr` are required to handle the C `printf()`-style escaped characters, like `\n`. Instead, they use the octal code (012) to represent a newline. An EBCDIC newline is octal 025. The only middle ground is to revert to the `\n` semantics, which may not be portable to certain XPG4.2 or POSIX machine architectures. Fortunately, Linux and OS/390 UNIX handle the `\n` semantics properly.

³ These changes permit the automatic identification of an OS/390 UNIX System Services platform and make canonical the token "s390-ibm-os390" to represent it.

Chapter 8. Bash

This chapter discusses the bash open source software package.

8.1 Introduction to bash

Bash is a shell or command language interpreter. It is compatible with the standard Bourne shell and incorporates useful features from the Korn shell and C shell. It is intended to conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools standard. It offers functional improvements over `sh` for both programming and interactive use. In addition, most `sh` scripts can be run by Bash without modification.

Bash is the default shell in Linux and therefore very popular with the open source community.

8.2 Installation of bash

The tools `gunzip` and `bison` are needed to install `bash`.

Here is the core of the install script named `bash.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d bash-2.03-mvs.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s bash-2.03-mvs bash
cd bash
ln -s /usr/include/stdlib.h ./alloca.h
export _CC_INCDIRS="/usr/include $PWD"
export LOCAL_LDFLAGS=-Wl,EDIT=NO
./configure --prefix=$installDir
ln -s /bin/touch ./size
export PATH=$PATH:$PWD
make
cp bash $installDir/bin
cp doc/bash.1 $installDir/man/$LANG/cat1
```

The installation instructions are in the file `README.MVS` that accompanies the package.

8.3 Customization and operation of bash

Assuming it's in your PATH, you can get a bash shell simply by typing `bash`. Probably the most important feature of bash that people rely on is command or filename completion. To try this feature out, open a bash shell and start typing the first few characters of a filename, and press the tab key. If those characters uniquely identify a file, the filename will be completed. If they identify more than one file, the point at which it is ambiguous will be completed. If you then press the tab key again, bash will either show you the file names that will complete the pattern, or the command, depending on the context.

If you like bash, you may want to change it to your default script. This can be done in TSO with the `ALU` command. For example:

```
tsocmd ALU MIKEM OMVS(program('/usr/local/bin/bash'))
```

8.4 License

The terms of use and redistribution of bash are governed by the GPL. See Appendix A, "The GNU General Public License" on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

8.5 Getting bash from the Internet

The bash package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.03. The package was found on the Internet at:

<http://www.gnu.org/software/bash/bash.html#downloading>

The tar file was FTPed to OS/390 UNIX in binary, and the two informational files FTPed in text mode.

8.6 Porting bash to OS/390

The bash program was ported to OS/390 UNIX by Greg Smith of Lockheed Martin. During our redbook project, we ran into some issues building. The

configure script ran fine, then the make command chugged along, but failed with the error:

```
/bin/c++ -DPROGRAM="bash" -DCONF_HOSTTYPE="i370"
-DCONF_OSTYPE="mvs" -DCONF_MACHTYPE="
i370-ibm-mvs" -DCONF_VENDOR="ibm" -DSHELL -DHAVE_CONFIG_H
-DNO_MAIN_ENV_ARG -D_ALL_SOURC
E -I. -I. -I./lib -I/usr/local/include -g -c xmalloc.c
ERROR CBC3343 ./xmalloc.c:50 Redclaration of sbrk differs from
previous declaration on line 676 of "/usr/include/unistd.h".
```

It appears that the function sbrk should *not* be defined in xmalloc.c, if it is being defined in /usr/include/unistd.h. The file xmalloc.c has the following code near line 50:

```
#if defined (HAVE_SBRK) && !defined (SBRK_DECLARED)
extern char *sbrk();
#endif
```

The file config.h, which is created by the configure script, had the macro SBRK_DECLARED commented out. To work around this problem, the following line was added to config.h:

```
#define SBRK_DECLARED 1
```

Now the file xmalloc.c compiled, but the bash executable did not link. It failed with the error:

```
/bin/c++ -L./builtins -L./lib/readline -L./lib/readline -L./lib/glob
-L./lib/tilde -L./lib/malloc -L./lib/sh -g -o bash shell.o eval.o
y.tab.o general.o make_cmd.o print_cmd.o dispose_cmd.o execute_cmd.o
variables.o copy_cmd.o error.o expr.o flags.o jobs.o subst.o hashcm
d.o hashlib.o mailcheck.o trap.o input.o unwind_prot.o pathexp.o sig.o
test.o version.o alias.o array.o braces.o bracecomp.o bashhist.o
bashline.o siglist.o list.o stringlib.o locale.o findcmd.o redir.o
xmalloc.o -lbuiltins -lreadline -lhistory -lcurses -lglob -ltilde
-lmalloc -lsh
IEW2456E 9207 SYMBOL timeval_to_secs UNRESOLVED. MEMBER COULD NOT BE
INCLUDED FROM THE DESIGNATED CALL LIBRARY.
```

The function timeval_to_secs() is in the source file general.c, but it appears to be preprocessed out:

```
#if defined (HAVE_TIMEVAL)
/* Convert a pointer to a struct timeval to seconds and thousandths of a
second, returning the values in *SP and *SFP, respectively. This does
rounding on the fractional part, not just truncation to three places.
*/
void
```

```
timeval_to_secs (tvp, sp, sfp)
...
#endif /* HAVE_TIMEVAL */
```

This was also worked around by adding a line to config.h:

```
#define HAVE_TIMEVAL 1
```


Chapter 9. CVS

This chapter discusses the CVS open source software package.

9.1 Introduction to CVS

CVS is a version control system (like RCS or SCCS), which allows you to keep old versions of files (usually source code) and to keep a log of when and why changes occurred, who performed them, and so on. It allows a client-server relationship, similar to FTP, such that source files can be modified via the Internet.

CVS provides a repository, or library system, for managing different versions of files. It supports access by multiple developers, both locally and over a TCP/IP network. Because of the support for many people working with the repository over the Internet, it is the repository of choice for most open software development projects.

CVS also provides an effective distribution system for source code updates because of its ability to efficiently integrate changes to the central repository into a local copy of the repository.

The port of CVS to OS/390 supports the use of local CVS repositories as well as the ability to interact with remote repositories. CVS on OS/390 cannot be used to manage a repository which is accessed from other systems.

9.2 Installation of CVS

There are no requirements beyond base OS/390 for building CVS, other than a C compiler and gzip.

Building CVS results in one executable file—cvs, which can be placed in any directory listed in your PATH environment variable, such as /usr/local/bin.

9.2.1 Core of the cvs.install script

For quick reference, here is the core of the install script named cvs.install, which is included on the CD (where \$installDir is typically /usr/local):

```
pax -rzf cvs-1-11-0-5-mvs3.tar.Z
ln -s cvs-1-11-0-5-mvs3 cvs
cd cvs
./configure --prefix=$installDir
make
```

```
make -i install
```

9.2.2 Limitations of CVS on OS/390

CVS on OS/390 cannot act as a server; in other words, it cannot be used to manage a repository which is accessed from other systems.

CVS on OS/390 does not handle binary files properly when communicating with a server across the network. Binary files are created with a special tag so that translation is not performed. However, the OS/390 port does not detect the tag and translation is performed, therefore destroying the file contents. In most CVS repositories, few if any files are binary files. As long as you don't attempt to update the server with such a file from OS/390, there will be no problems.

9.3 Customization and operation of CVS

No special customization of CVS is required for OS/390. As on other platforms, users can set environment variables or set up default options (in `~/cvsrsc`).

Generally, the user wishing to interact with a CVS repository will set the `CVSROOT` environment variable to point to that repository, at which point CVS commands will work properly.

For detailed information about how to use CVS, you are referred to a HOW-TO on the linuxdoc.org Web site at:

http://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/html_single/

From there, search for the document *CVS-RCS-HOWTO.html*.

We conducted the “make check” test for CVS and received a final “OK” for this test (which includes the test for `sanity.sh`).

9.4 License

The terms of use and redistribution of CVS are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

9.5 Getting CVS from the Internet

You can download CVS-1-11-0-5 ported to MVS from:

<http://dccmn.com/cvs-1-11-0-5-mvs3.tar.Z>

9.6 Porting CVS to OS/390

This version of CVS was ported by Wayne Johnson. No notes were found regarding the port.

9.6.1 Future work needed on the OS/390 port of CVS

These tasks would make good projects for people interested in CVS on OS/390:

1. Currency with CVS on other platforms.

Update the port of CVS to OS/390 with fixes and enhancements which are available on other platforms.

2. Integrate OS/390 changes into the master CVS sources.

Even if OS/390-specific changes are not desired, some of the changes required to run on OS/390 are more general. For example, changes were made to handle `getenv()` returning the address of a buffer which is overlaid on subsequent calls to `getenv()`. That is an allowable behavior, so those changes could help CVS on other platforms.

There are many places where `\012` in the source was changed to `\n` to work on OS/390. It would be great to change the master CVS sources to use a preprocessor symbol such as `NL` for new-line and have `NL` be defined to `\n` almost everywhere, but to `\012` on the one platform where `\n` isn't normally `\012`.

3. Proper handling of binary files.

Step one: recognize that a file is binary and skip processing.

Step two: transfer the file in binary mode.

4. Server mode.

Currently, only `cvs login` will work when accessing a CVS server on OS/390 across the network. Much more work is required to support other operations.

5. Socksify.

Many OS/390 machines are behind a firewall and many CVS repositories are on the Internet. It would be nice to provide a mechanism for linking CVS with socks libraries.

9.7 Other resources

Pascal Molli maintains a set of links to CVS resources at:

<http://www.loria.fr/cgi-bin/molli/wilma.cgi/doc>

One of the links is to the CVS reference manual, at:

http://www.loria.fr/~molli/cvs/doc/cvs_toc.html

This reference manual is the best resource for learning more about CVS operations.

Chapter 10. Diffutils

This chapter discusses the diffutils open source software package.

10.1 Introduction to diffutils

The diffutils package consists of the `diff`, `cmp`, `diff3` and `sdiff` commands. A description of diffutils was taken from the GNU Web site:

Computer users often find occasion to ask how two files differ. Perhaps one file is a newer version of the other file. Or maybe the two files started out as identical copies but were changed by different people.

You can use the 'diff' command to show differences between two files, or each corresponding file in two directories. 'diff' outputs differences between files line by line in any of several formats, selectable by command line options. This set of differences is often called a 'diff' or 'patch'. For files that are identical, 'diff' normally produces no output; for binary (non-text) files, 'diff' normally reports only that they are different.

You can use the 'cmp' command to show the offsets and line numbers where two files differ. 'cmp' can also show all the characters that differ between the two files, side by side.

You can use the 'diff3' command to show differences among three files. When two people have made independent changes to a common original, 'diff3' can report the differences between the original and the two changed versions, and can produce a merged file that contains both persons' changes together with warnings about conflicts.

You can use the 'sdiff' command to merge two files interactively.

10.2 Installation of diffutils

Here is the core of the install script named `diffutils.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d diffutils-2.7.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s diffutils-2.7 diffutils
cd diffutils
export CC=c89
export CFLAGS=-D_ALL_SOURCE
export LDFLAGS=-Wl,EDIT=NO
./configure --prefix=$installDir
make
```

```
echo "installing via the command 'make install' ..."  
make install
```

10.3 Customization and operation of diffutils

Assuming the diffutils package is installed properly, the commands `diff`, `diff3`, `cmp` and `sdiff` are in your PATH and ready to execute. The `diff` command is used to show the difference between two files, and `diff3` is used for comparing three files. The `cmp` command is used to show the offsets and line numbers where two files differ. Often the return code from `cmp` is used for a basic yes/no answer that determines whether two files are identical. The `sdiff` command can be used to merge two files interactively.

Unlike `/bin/diff`, which is part of OS/390 UNIX, the diffutils version can be used to compare both text and binary files. Here is an example of `diff` comparing with two binary files:

```
$ which diff  
/usr/local/bin/diff  
$ diff /bin/link /bin/lex  
Binary files /bin/link and /bin/lex differ  
$ /bin/diff /bin/link /bin/lex  
diff: FSUM8821 file "/bin/link" is binary  
diff: FSUM8821 file "/bin/lex" is binary
```

Note that `/bin/diff` does not actually compare the files because they are binary.

Here is an example of using both the diffutil package's and OS/390's `cmp` command:

```
$ which cmp  
/usr/local/bin/cmp  
$ cmp /bin/link /bin/lex  
/bin/link /bin/lex differ: char 18, line 1  
$ /bin/cmp /bin/link /bin/lex  
/bin/link /bin/lex differ: char 18, line 1
```

In this case, both versions of the command give the same results.

10.4 License

The terms of use and redistribution of diffutils are governed by the GPL. See Appendix A, "The GNU General Public License" on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

10.5 Getting diffutils from the Internet

The diffutils package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.7. The package was found at:

<http://www.gnu.org/software/diffutils/diffutils.html#downloading>

The package was moved to OS/390 in binary via FTP. It was unwound using the `pax` command with the text translation flag `-o from=ISO8859-1,to=IBM-1047` (stored in the environment variable `A2E`) to convert from ASCII to EBCDIC:

```
$ ls -l
total 640
-rw-r----- 1 MIKEM  SYS1      312330 Feb 21 15:49
diffutils-2.7.tar.gz
$ gunzip *.gz
$ ls -l
total 2688
-rw-r----- 1 MIKEM  SYS1      1167360 Feb 21 15:49 diffutils-2.7.tar
$ pax -rf *.tar $A2E
$ ls -l
total 2312
-rw-r----- 1 MIKEM  SYS1      1167360 Feb 21 15:49 diffutils-2.7.tar
drwxr-xr-x  2 MIKEM  SYS1          8192 Oct  2 1994 diffutils-2.7/
$ ln -s diffutils-2.7 diffutils
```

10.6 Porting diffutils to OS/390

The Makefile was created with no problem using the command `./configure`. However, the code would not compile, due largely to command-line flags in case statements with hard-coded values of 129 and above.

A patch file was found on the MVS-OE listserver which corrects the problem. Thanks to Peter Prymmer of Forte for the patch file he put on the MVS-OE listserver. His rationale for these changes is:

Since these cases appear to be command line switch handling and I was only interested in `-u` and `-l` and I have never used an eight bit switch on an ASCII computer I can't see what harm this'd do: omitted on EBCDIC but retained on ASCII machines.

A function prototype for `fnmatch` in `fnmatch.h` was preprocessed out because it would not compile as is. Also, the function `private_strerror` was preprocessed out in `error.c`

You can grep for `__MVS__` in the code to see the changes

```
$ grep __MVS__ *. [ch]
diff.c:#ifndef __MVS__
diff3.c:#ifndef __MVS__
error.c:#ifndef __MVS__
fnmatch.h:#ifndef __MVS__
sdiff.c:#ifndef __MVS__
```

The man pages for `diff`, `sdiff`, `diff3` and `cmp` were obtained from a Linux system, converted to flat text and added to the package:

```
$ ls *.1
sdiff.1  diff3.1  diff.1  cmp.1
```

The most important features that the `diffutils` package adds to OS/390 are the ability to compare binary files with the `diff` command and the `diff3` command which is needed by some other packages.

Chapter 11. Emacs

This chapter discusses the emacs open source software package.

11.1 Introduction to emacs

The name emacs comes from Editor MACroS. Emacs is based on a macro language written in 1976 for the editor TECO (Text Editor and COrrector) on a PDP-10.

Today, emacs is one of the most popular editors in the open software environment. Developed under the GPL, it is available on many platforms and it has become, in some ways, a standard user interface.

The characteristics of emacs are:

- *It's a display editor:* You can see the text you edit on the screen and it is updated as you type.
- *It's a real-time editor:* The screen is updated for nearly every key you type on the keyboard.
- *It's advanced:* It provides all facilities which are required for a modern editor, not only simple update, insert, and deletion. You can open multiple files, control subprocesses, use formatted editing functions and work with words, lines, sentences, paragraphs, and pages, as well as expressions and comments.
- *It's self-documenting:* You can use `CTRL-h` anytime you need help or to access the internal documentation. It explains every command and error message and even contains a tutorial for first time users.
- *It's customizable:* You can change the definition of emacs commands and you can redefine the control keys.
- *It's extensible:* You can use the Lisp language to write and run new functions as part of the editor.

Emacs has been ported to OS/390 by MKS Software in partnership with IBM. We are not aware of any plans to integrate the changes into the mainstream of the product.

11.2 Installation of emacs

Even though most of the installation steps are described in this chapter, we strongly advise you to read the `README` and `INSTALL` file, which are part of the

emacs distribution directory. It contains the latest information about the installation and customization process.

11.2.1 Step 1: Load the software

The emacs package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is the OS/390 Beta Release 4 of GNU emacs 19.34.

After you have copied the package to your PC, you have to transfer it in binary mode to the OS/390 HFS file system. Don't ignore the `mkdir emacs` and the `cd emacs`. The pax file does not contain additional path information and it will mess up your installation if you don't install into an empty directory.

```
$ export $A2E="-o from=ISO8859-1,to=IBM-1047"
$ mkdir emacs
$ cd emacs
$ pax -rf /u/egont/emacs-19_34_os390_tar.Z $A2E
```

11.2.2 Step 2: Configure and build emacs

The complete installation of emacs will require about 100 MB of disk space and will take at least one hour. You don't have to watch the screen all the time. The `configure` and the `make install` steps will require no input and will run on their own.

Decide where you want to install emacs and export the name of the directory into a variable called `INSTALLDIR`. This is very important. The make process will copy the emacs program into `$INSTALLDIR/bin` and emacs will always issue warnings if it has started out of another location.

If you are not certain that you have proper access rights to the installation directory and the recently loaded distribution, you probably want to run this process as a superuser.

```
$ export INSTALLDIR='/usr/local'
$ ./configure i390-ibm-omvs --with-x=yes --with-x-toolkit=no \
--prefix=$INSTALLDIR
$ make install
```

11.2.3 Troubleshooting

It has been observed that the emacs build process can *hang* (get stuck) on OS/390 V2R9. This seems to be related to the `tee` command. We suggest that you do not use `tee` with the install scripts, `5944.install` or `emacs.install`.

11.3 Customization and operation of emacs

You can run emacs within your normal UNIX session or as an X/Window application. It automatically checks whether you have set the DISPLAY variable and starts the program accordingly. The X/Window mode gives you a full-screen editor and full mouse support, but it requires the installation of an X-Window server on your desktop. For more information and useful tips regarding X/Window, see:

- 50.1.4, “Setting the DISPLAY variable for X Window applications” on page 273
- 50.1.6, “Downloading an X Window server from the Internet” on page 274

In both cases, emacs offers to run a tutorial (`Ctrl+h t`) after startup. We strongly recommend this tutorial to any first-time user. If you have used emacs before on other systems you are ready to go. You will quickly learn how convenient it is to have a help facility which you can call with `Ctrl+h` at any time.

There is a full copy of online documentation on the GNU Web Page. It's available in all formats and can be downloaded for online viewing or to print a copy. You can find this documentation at:

<http://www.gnu.org/manual/emacs-20.3/emacs.html>

11.4 License

Emacs is covered by the GNU General Public License, which is explained in detail in Appendix A, “The GNU General Public License” on page 289.

11.5 Getting emacs from the Internet

The version of the package on the CD is the OS/390 Beta Release 4 of GNU Emacs 19.34. It was found on the Internet at:

<ftp://ftp.mks.com/pub/s390/gnu/>

The ASCII source package was downloaded to the PC and then moved to OS/390 in binary via FTP.

11.6 Porting emacs to OS/390

This version of emacs has been ported by MKS Software in cooperation with IBM. We don't have any information about what has been changed and how

difficult it was to port the package to OS/390. We are also not aware if MKS will port future releases or if the changes will be integrated into the mainline of the code. Please report your successful installation or any kind of problems to MKS to make them aware that emacs is used on this platform.

There is also a binary version of the product available on the MKS ftp server or on the CD, but it requires that you install the product in exactly the same directory paths, as it was during the build process.

If you are troubled by warnings that appear every time you start emacs, we suggest that you repeat the installation of the source package.

Please note that the installation of Emacs will take some time and disk space, but the result is really worth the effort.

Chapter 12. Fgres

This chapter discusses the fgres open source software package.

12.1 Introduction to fgres

The term fgres is an acronym for Fast GREp and Substitute. Its main function is to substitute or delete character patterns. It can operate on binary and text files and can be used with pipes.

fgres can do fast, efficient replacements in multiple files. Unlike sed, fgres can replace text in an existing document. The sed tool typically uses a temporary file to accomplish this.

12.2 Installation of fgres

Here is the core of the install script named `fgres.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d fgres.os390.tar.gz|pax -o from=ISO8859-1,to=IBM-1047 -r
cd fgres
make
cp fgres $installDir/bin
chmod 755 $installDir/bin/fgres
cp fgres.1 $installDir/man/man1/fgres.1
chmod 644 $installDir/man/man1/fgres.1
```

12.3 Example of using fgres

As shown on the Consortium Web site, to replace the text foo with foobar in all your source code, use the following command:

```
$ fgres foo foobar *. [ch]
```

12.4 License

The following license was found in the file README:

“Copyright (c) 1994, W. Stumvoll, Germany

Use this software package at your own risk. The programmer cannot be held liable for any incurred damages, directly or indirectly due to the use or inability to use this software.

You are encouraged to distribute this package freely. This package is however not to be sold (minor transfer costs excepted) or included in any commercially sold software package (if you want to do this anyway, contact me (address below), and we'll work something out).

If you distribute it, please leave the package intact. You are allowed to take parts from this distribution and distribute these separately as long as you retain the copyright messages. If you redistribute any part of this package in a modified form, be sure to mark the parts you changed. If you have some important changes that might be useful to the rest of the world, contact me instead.

stumvoll@pool.informatik.rwth-aachen.de"

12.5 Getting fgres from the Internet

You can download fgres from the Internet site starting at:

<http://ftp.sunet.se/ftp/pub/usenet/ftp.uu.net/comp.sources.unix/volume28/fgres/>

12.6 Porting fgres to OS/390

We needed to make the following changes in order to get fgres to run under OS/390.

To avoid the message:

```
"Sys_nerr and Sys_errlist unresolved"
```

in messages.c at line 24, substitute the following:

```
return((errno<sys_nerr) ? sys_errlist[errno] : def);
```

with:

```
return(strerror(errno));
```

In the makefile you need to specify which compiler to use (we used cc with optimization level 2), and to force it not to use SYSEXIT (otherwise, the compilation fails). The following statements accomplish this:

```
CC=cc -O2  
SE_CPP=-DNOSYSEXIT
```

Chapter 13. Findutils

This chapter discusses the findutils open source software package.

13.1 Introduction to findutils

The findutils package provides tools to find files which meet specified criteria. The programs which are used to perform this task are `find`, `locate`, and `xargs`.

The `find` command is the GNU version of the `find` program shipped with OS/390 UNIX. There are some additional parameters (like `-regex`), but also some missing (like `-audit`).

The `xargs` command reads lines from standard input, appends them to a command given as parameter, and executes this command. An `xargs` program is also shipped with OS/390 UNIX.

The `locate` command is used to search for files in a database created by the program `updatedb`. The default `updatedb` starts in the root directory and searches through the entire directory structure, adding all found files and directories to its database. Because of this model, the `locate` command is much faster than the `find` command which is commonly used.

As `find` and `xargs` are also part of the OS/390 UNIX distribution, our main interest is in `locate` and `updatedb`. The `updatedb` command can use parameters, so it only searches a given directory and stores the database at a given place. In this fashion, a user could maintain a database to search their own data, and not the entire system's.

13.2 Installation of findutils

If you want to install the binary distribution from the CD, refer to Chapter 3, "What's on the CD-ROM" on page 17.

If you want to compile it yourself, you need the source code from the CD. Building findutils on OS/390 UNIX is easy. Just do the following:

```
$ export CC=cc
$ export CFLAGS=-D_ALL_SOURCE
$ export _CC_CCMODE=1
$ ./configure
creating cache ./config.cache
checking for gcc... cc
checking whether we are using GNU C... no
```

```

...
creating config.h
$ make
for subdir in lib find xargs locate doc testsuite; do \
    echo making all in $subdir ; \
    (cd $subdir; make all); \
done
making all in lib
...
cc -c -DHAVE_CONFIG_H -I.. -I. -D_ALL_SOURCE getopt.c
WARNING CBC3180 /usr/include/stdio.h:138  Redclaration of built-in
function _gtca ignored.
FSUM3065 The COMPILE step ended with return code 4.
cc -c -DHAVE_CONFIG_H -I.. -I. -D_ALL_SOURCE getopt1.c
WARNING CBC3180 /usr/include/stdio.h:138  Redclaration of built-in
function _gtca ignored.
FSUM3065 The COMPILE step ended with return code 4.
...
make[1]: Entering directory `/u/ralfs/src/findutils-4.1/xargs'
cc -c -DHAVE_CONFIG_H -I.. -I../lib -D_ALL_SOURCE xargs.c
WARNING CBC3296 ./xargs.c:67  #include file <sys/param.h> not found.
FSUM3065 The COMPILE step ended with return code 4.
...
$ make install
...

```

The warnings about the redeclaration of built-in function `_gtca` can be ignored. Even the missing file `sys/param.h` doesn't prevent the compiler from compiling and linking successfully.

13.3 Customization and operation of findutils

The command `locate` works in conjunction with the update database which is created by the command `updatedb`. It is common to run `updatedb` every night, so the database is always up to date. If the database is older than eight days, `locate` will issue a warning.

We used the following line in our crontab (use `crontab -e` to edit your crontab):

```
07 5 * * * /usr/local/bin/updatedb >/dev/null 2>&1
```

With this entry the database is refreshed every morning at 05:07 a.m.

The `find` command is GNU's version, with parameters that differ from OS/390's. The command `xargs` reads from the standard input, appends to a command given as a parameter, and executes the command.

13.4 License

The `findutils` package is released under GNU General Public License. See Appendix A, "The GNU General Public License" on page 289.

13.5 Getting `findutils` from the Internet

The `findutils` package Version 4.1 is available on the CD that accompanies the hardcopy of this redbook. The package was found at:

<ftp://ftp.gnu.org/gnu/findutils/findutils-4.1.tar.gz>

The archive is ASCII and can be extracted with:

```
$ gzip -d findutils-4.1.tar.gz
$ pax -ofrom=ISO8859-1 -rf findutils-4.1.tar
$ ls -l
total 2720
drwxr-xr-x  8 RALFS  SYS1          8192 Mar  3 16:03 findutils-4.1/
-rw-r--r--  1 RALFS  SYS1       1187840 Mar  3 16:01 findutils-4.1.tar
```

13.6 Porting `findutils` to OS/390

It was not necessary to modify the `findutils` package; it built cleanly from the Web.

Chapter 14. Flex and Bison

This chapter discusses the flex and bison open source software packages.

14.1 Introduction to flex and bison

Flex and Bison are tools that aid in programming lexical analysis and syntactic analysis, which are the first two architectural stages of a compiler. Because these two tools are typically used together, they are addressed in the same chapter.

The name flex was derived from Fast LEX. Here is a description of flex:

Flex is a fast lexical analyzer generator. It is a tool for generating programs that perform pattern-matching on text. There are many applications for flex, including writing compilers in conjunction with GNU Bison. Flex is a free implementation of the well-known Lex program. It features a Lex compatibility mode, and also provides several new features such as exclusive start conditions.

This description was obtained on the Internet at:

<http://www.gnu.org/software/flex/flex.html>

Here is a description of bison:

Bison is a general-purpose parser generator that converts a grammar description for an LALR context-free grammar into a C program to parse that grammar. Once you are proficient with Bison, you can use it to develop a wide range of language parsers, from those used in simple desk calculators to complex programming languages.

Bison is upward compatible with Yacc: all properly-written Yacc grammars ought to work with Bison with no change. Anyone familiar with Yacc should be able to use Bison with little trouble. You need to be fluent in C programming in order to use Bison.

This description was obtained on the Internet at:

<http://www.gnu.org/software/bison/bison.html>

14.2 Installation of flex and bison

The tool gunzip is needed to install flex and bison. Additionally, flex requires bison, so bison should be built first and the directory where the executable resides should be put in your PATH.

14.2.1 Installation of bison

Here is the core of the install script named bison.install, which is included on the CD that accompanies the hardcopy of this redbook (where \$installDir is typically /usr/local):

```
gzip -c -d bison1.25.os390.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s bison-1.25 bison
cd bison
export _C89_CCMODE=1
./configure --prefix=$installDir
make
cp bison $installDir/bin/bison
cp bison.1 $installDir/man/$LANG/cat1/bison.1
cp bison.simple $installDir/share
cp bison.hairy $installDir/share
```

14.2.2 Installation of flex

Here is the core of the install script named flex.install, which is included on the CD that accompanies the hardcopy of this redbook (where \$installDir is typically /usr/local):

```
pax -o from=ISO8859-1,to=IBM-1047 -rzf flex-2.5.4a.os390.tar.Z
ln -s flex-2.5.4a flex
cd flex
LDFLAGS=-Wl,EDIT=NO ./configure --prefix=$installDir
make
cp flex $installDir/bin/flex
cp flex.1 $installDir/man/$LANG/cat1/flex.1
```

14.3 Customization and operation of flex and bison

Assuming the packages were properly installed, flex and bison should be in your PATH and ready to execute. They should be drop in replacements for the corresponding OS/390 UNIX lex and yacc tools. Some packages' Makefiles explicitly call flex and bison.

There are variables in the make command's default rules file (/etc/startup.mk) which specify some default rules for *.l and *.y files:

```
YACC      := yacc          # LEX&YACC
LEX       := lex

LEXYY     := lex.yy        # LEX output
YTAB      := y.tab         # YACC output

YACCR     := yacc -r       # for System V compatibility
```

```
YACCE      := yacc -e
```

In this file, the `lex` and `yacc` commands could be replaced with `flex` and `bison`.

14.4 License

The terms of use and redistribution of both `flex` and `bison` are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

14.5 Getting flex and bison from the Internet

The Flex and Bison packages are available on the CD that accompanies the hardcopy of this redbook. The version of the Flex package on the CD is 2.5.4a. The version of the Bison package on the CD is 1.25. These packages (“Flex source” and “Bison source”) were found on the MKS ftp server at:

```
ftp://ftp.mks.com/pub/s390/gnu/
```

Both packages are in ASCII, compressed format. The Flex package was unwound using the following `pax` command (where the A2E environment variable contains `-o from=ISO8859-1,to=IBM-1047`):

```
$ ls -l
total 1120
-rw-r--r--  1 MIKEM   SYS1          573440 Feb 22 11:19
flex-2_5_4a_os390_tar.Z
$ pax -rzf *.Z $A2E
$ ls -l
total 1152
drwxr-xr-x  4 MIKEM   SYS1          8192 Feb 22 11:40 flex-2.5.4/
-rw-r--r--  1 MIKEM   SYS1          573440 Feb 22 11:19
flex-2_5_4a_os390_tar.Z
$ ln -s flex-2.5.4 flex
```

The Bison package was unwound in a similar fashion:

```
$ ls -l
total 968
-rw-r--r--  1 MIKEM   SYS1          492840 Feb 22 11:19
bison-1_25_os390_tar.Z
$ pax -rzf *.Z $A2E
```

```
$ ls -l
total 1000
-rw-r--r--  1 MIKEM   SYS1      492840 Feb 22 11:19
bison-1_25_os390_tar.Z
drwxr-xr-x  3 MIKEM   SYS1      8192 Oct 30  1997 bison-1.25./
$ ln -s bison-1.25. bison
```

14.6 Porting flex and bison to OS/390

Because these ports were done at MKS, there is not much detail on the porting process. Building the packages should be a simple re-configure/compile.

In Bison, we made a change in file version.c from :

```
char *version_string = "GNU Bison version 1.25.os390\n";
```

to :

```
char *version_string = "GNU Bison version 1.25\n";
```

This to avoid some program not to be able to check the version number due to non-numeric characters in version number.

14.6.1 An issue with bison

Invoking the command `make` resulted in the following error on the first compile:

```
# make
c89 -c -DXPFILE=\"/usr/local/share/bison.simple\"
-DXPFILE1=\"/usr/local/share/bison.hairy\" ./files.c -o files.o
FSUM3008 Specify a file with the correct suffix (.c, .i, .s, .a, .o, .x,
.-o, or .?), or a c.
make: *** [files.o] Error 1
```

Setting the environment variable `_C89_CCMODE` to 1 allows a liberal ordering of command line parameters. This fixed the problem and bison built successfully.

```
# export _C89_CCMODE=1
# make
c89 -c -DXPFILE=\"/usr/local/share/bison.simple\"
-DXPFILE1=\"/usr/local/share/bison.hairy\" ./files.c -o files.o
c89 -c -DSTDC_HEADERS=1 -DHAVE_STRING_H=1 -DHAVE_STDLIB_H=1
...
```

14.6.2 An issue with flex

One issue was encountered during the build of flex:

```
$ make
/bin/c++ -c -I. -I. -g ccl.c
/bin/c++ -c -I. -I. -g dfa.c
/bin/c++ -c -I. -I. -g ecs.c
/bin/c++ -c -I. -I. -g gen.c
/bin/c++ -c -I. -I. -g main.c
/bin/c++ -c -I. -I. -g misc.c
/bin/c++ -c -I. -I. -g nfa.c
yacc -d ./parse.y
/bin/c++ -c -I. -I. -g parse.c
WARNING CBC3209 ./parse.c:207 Character constants must end before
the end of a line.
FSUM3065 The COMPILER step ended with return code 4.
/bin/c++ -c -I. -I. -g scan.c
...
```

Line 207 of parse.c (which was actually line 271 due to a #line preprocessor directive) contained the following:

```
{ "''''", 127, 0},
```

It can be seen from the preceding line that the source file parse.c is generated by calling yacc with the input file parse.y. This improperly escaped string appears to be an error in the output of yacc. When bison is built before flex, the error is fixed:

```
building via the command 'make'
c89 -c -I. -I. -g ccl.c
c89 -c -I. -I. -g dfa.c
c89 -c -I. -I. -g ecs.c
c89 -c -I. -I. -g gen.c
c89 -c -I. -I. -g main.c
c89 -c -I. -I. -g misc.c
c89 -c -I. -I. -g nfa.c
bison -y -d ./parse.y
c89 -c -I. -I. -g parse.c
c89 -c -I. -I. -g scan.c
```

Then parse.c compiled without a warning. However, the file parse.c was generated by the preceding command yacc -d ./parse.y. Therefore it appears that yacc is generating incorrect code. The counterpart to flex, bison was built first and put in the PATH.

Chapter 15. Gd

This chapter discusses the gd open source software package.

15.1 Introduction to gd

This gd library is also known as “Boutell GD”. It is a graphics library that allows your code to draw images with lines, arcs, text and multiple colors. It allows you to cut and paste from other images, perform “flood fills”, and write out the result as a PNG or JPEG file. This library is particularly useful in Web applications, where PNG and JPEG are two image formats accepted by most browsers.

15.2 Installation of gd

Your OS/390 UNIX system must have the following prerequisite packages present:

- zlib
- libPNG
- jpeg-6b (optional but recommended)

This is the heart of the gd.install script:

```
gzip -c -d gd-1.8.4.os390.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s gd-1.8.4 gd
cd gd
chmod u+w *
export _C89_CCMODE=1
make
make install
```

Assuming you installed in the directory /usr/local, binaries should be in /usr/local/bin, the library libgd.a should now be in /usr/local/lib, and header files should be in /usr/local/include.

15.3 Customization and operation of gd

The gd library lets you create PNG or JPEG images on the fly. To use gd in your program, include the file gd.h, and link with the libgd.a library. Following is a short example of using gd, together with jpeg and png libraries:

```
/* Bring in gd library functions */
#include "gd.h"
```

```

/* Bring in standard I/O so we can output the PNG to a file */
#include <stdio.h>
int main() {
/* Declare the image */
    gdImagePtr im;
/* Declare output files */
    FILE *pngout, *jpegout;
/* Declare color indexes */
    int black;
    int white;
/* Allocate the image: 64 pixels across by 64 pixels tall */
    im = gdImageCreate(64, 64);
/* Allocate the color black (red, green and blue all minimum).
Since this is the first color in a new image, it will
be the background color. */
    black = gdImageColorAllocate(im, 0, 0, 0);
/* Allocate the color white (red, green and blue all maximum). */
    white = gdImageColorAllocate(im, 255, 255, 255);

/* Draw a line from the upper left to the lower right,
using white color index. */
    gdImageLine(im, 0, 0, 63, 63, white);
/* Open a file for writing. "wb" means "write binary", important
under MSDOS, harmless under Unix. */
    pngout = fopen("test.png", "wb");

/* Do the same for a JPEG-format file. */
    jpegout = fopen("test.jpg", "wb");

/* Output the image to the disk file in PNG format. */
    gdImagePng(im, pngout);

/* Output the same image in JPEG format, using the default
JPEG quality setting. */
    gdImageJpeg(im, jpegout, -1);

/* Close the files. */
    fclose(pngout);
    fclose(jpegout);

/* Destroy the image in memory. */
    gdImageDestroy(im);
}

```

15.4 License

The following license is found in the file readme.txt:

“Portions copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.

Portions copyright 1996, 1997, 1998, 1999, 2000 by Boutell.Com, Inc.

Portions relating to GD2 format copyright 1999, 2000 Philip Warner.

Portions relating to PNG copyright 1999, 2000 Greg Roelofs.

Portions relating to libtiff copyright 1999, 2000 John Ellison (ellson@lucent.com).

Portions relating to JPEG copyright 2000, Doug Becker and copyright (C) 1994-1998, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group.

Portions relating to WBMP copyright 2000 Maurice Szmurlo and Johan Van den Brande.

Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation.

This software is provided "AS IS." The copyright holders disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this code and accompanying documentation.”

15.5 Getting gd from the Internet

You can download gd from the Internet starting at:

<http://www.boutell.com/gd/>

15.6 Porting gd to OS/390

Note that gd does not have a configure script; rather, you must modify the Makefile. It is not difficult. We made these changes to Makefile:

```
COMPILER=c89
CFLAGS=-O -DHAVE_LIBPNG -DHAVE_LIBJPEG -D_OSD_POSIX
.....
LIBS=-lgd -lpng -lz -ljpeg -lm
.....
INCLUDEDIRS=-I. -I/usr/local/include
LIBDIRS=-L. -L/usr/local/lib
.....
```

The source program gdkanji.c was also modified, as follows:

```
62,63c62
< static debug(format, ...)
< char *format;
---
> static void debug(const char *format, ...)
82,83c81
< static error(format, ...)
< char *format;
---
> static void error(const char *format, ...)
```

After these patches, run the following:

```
$ export _C89_CCMODE=1; gmake
$ su; gmake install
```

Chapter 16. Gmake

This chapter discusses the open software GNU version of make, and in this redbook it is referred to as *gmake*.

16.1 Introduction to gmake

Like *make*, gmake is a utility which helps you to build a product. It manages all components of a project, like source and object files, and their relationship to each other.

OS/390 comes with the POSIX-compliant versions of make to provide maximum portability. Unfortunately many open software products have not been written according to these strict rules. Therefore installing the GNU version gmake is one of the prerequisites to install open software on OS/390.

To keep both versions available on the system you may choose to name the GNU version `gmake` and leave the standard `make` command unchanged, or you may want to only use the GNU version of make and specifically invoke `/bin/make` when the standard make is needed.

16.2 Installation of gmake

The version of the package on the CD is 3.79.1. If you want to obtain the latest copy of the product, refer to 16.5, "Getting gmake from the Internet" on page 86.

The complete process to uncompress, reload, and translate the package to EBCDIC can be done with the pax command all at once.

To prevent errors during the make process you have to set the environment variable `CC=cc`. This will force the install process to use the `cc` compiler instead of the more strict `c89` ANSI compiler.

The core of the script to install gmake is as follows:

```
gzip -c -d make-3.79.1.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s make-3.79.1 make
cd make
export CFLG='-O2 -D_ALL_SOURCE'
CC=cc CFLAGS="$CFLG" CCC=c++ CXXFLAGS="-+ $CFLG" ./configure
--prefix=$installDir
make
make install
```

If you want to enable the usage of the GNU make man pages, copy them into the default location for man pages:

```
$ cp make.man /usr/local/man/C/cat1.make.1
```

16.3 Customization and operation of gmake

There appears to be an inconsistency in the open source community regarding the name of GNU make. It was originally named “gmake,” following the convention of prefixing all GNU tools with the letter *g*. It is now accepted, especially on Linux, to simply name GNU make, “make.” When the gmake make process completes, the newly created tool is named make. However, some install packages, notably Info-ZIP, specifically call `gmake` to build the tools.

Due to these inconsistencies, and after building GNU make, the `gmake.install` script copies the executable and man page with the name `make`, and then creates symbolic links to them with the name `gmake`. Therefore, if you have `/usr/local/bin` in your `PATH` before `/bin`, either command, `make` or `gmake`, will invoke GNU make. You can always explicitly call the standard OS/390 make with the command `/bin/make`.

If you want to know more about the GNU enhancements to make or if you need more information, go to the GNU Web site:

<http://www.gnu.org/manual/manual.html>

Here you can find the complete documentation for all GNU tools in HTML or PDF format.

16.4 License

Even if you download the software from the MKS ftp server, it is still covered by the GNU General Public License. See Appendix A, “The GNU General Public License” on page 289, for details.

16.5 Getting gmake from the Internet

You can get gmake from the Internet at:

<ftp://ftp.mks.com/pub/s390/gnu/>

The package is in compressed ASCII Source code and you must transfer it in binary to the HFS file system on OS/390.

16.6 Porting gmake to OS/390

While compiling the sources, we got the following error:

```
-g -c arscan.c
WARNING CBC3296 ./arscan.c:236  #include file <ar.h> not found.
ERROR CBC3045 ./arscan.c:382  Undeclared identifier ARMAG.
ERROR CBC3007 ./arscan.c:431  "struct ar_hdr" is undefined.
ERROR CBC3043 ./arscan.c:532  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:537  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:624  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:696  The operand of the sizeof operator is not
valid.
ERROR CBC3007 ./arscan.c:730  "struct ar_hdr" is undefined.
ERROR CBC3007 ./arscan.c:773  "struct ar_hdr" is undefined.
ERROR CBC3043 ./arscan.c:788  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:788  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:793  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:793  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:812  The operand of the sizeof operator is not
valid.
ERROR CBC3043 ./arscan.c:812  The operand of the sizeof operator is not
valid.
```

In order to overcome this error, we modified the `#ifndef` statement in the file `arscan.c` as follows:

```
#ifndef WINDOWS32
#if !defined(__BEOS__) && !defined(__MVS__)
# include <ar.h>
# else
/* BeOS 5 doesn't have <ar.h> but has archives in the same format
 * as many other Unices. This was taken from GNU binutils for BeOS.
 */
# define ARMAG "!<arch>\n" /* String that begins an archive file.
 */
# define SARMAG 8 /* Size of that string. */
# define ARFMAG "`\n" /* String in ar_fmag at end of each
header. */
struct ar_hdr
{
```

```

        char ar_name[16];          /* Member file name, sometimes /
terminated. */
        char ar_date[12];          /* File date, decimal seconds since
Epoch. */
        char ar_uid[6], ar_gid[6]; /* User and group IDs, in ASCII decimal.
*/
        char ar_mode[8];          /* File mode, in ASCII octal. */
        char ar_size[10];         /* File size, in ASCII decimal. */
        char ar_fmag[2];          /* Always contains ARFMAG. */
    };
# endif

```

When we ran gmake check, we got the following errors:

```

functions/sort ..... FAILED
options/dash-q ..... Error running
/u/tot62/src/make-3.79.1/tests/./make (2048):
/u/tot62/src/make-3.79.1/tests/./make -f work/options/dash-q.mk -q
three
Error running /u/tot62/src/make-3.79.1/tests/./make (2048):
/u/tot62/src/make-3.79.1/tests/./make -f work/options/dash-q.mk -q
seven

targets/INTERMEDIATE ..... FAILED
3 Tests Failed (See .diff files in work dir for details) :-(

```

On checking the .diff files for the first and third errors respectively, it was found that these errors were due to interchanging of the order of certain files (this is because EBCDIC sorts with lower case first and then upper case while ASCII does it the other way round). However, this should not affect the working of the system.

Make installs cleanly.

Chapter 17. Gnats

This chapter discusses the gnats open source software package.

17.1 Introduction to gnats

The name *gnats* is a famous GNU recursive abbreviation, and it stands for “GNats: A Tracking System”. It is a bug-tracking system, based upon the paradigm of one or more central sites or organizations, which receives problem reports and negotiates their resolution by e-mail. It is sufficiently generalized so that it could be used for handling system administration issues, project management, or any number of other applications.

gnats is a GNU package under the GPL license, copyrighted by the FSF, but maintained by Cygnus Software. There are some historical signs of Cygnus Solutions having a commercially supported version of gnats under the name PRMS, but recently that is not available. The reason for trying to change the name is that there is another GNU package with a very similar name (Gnat - the Gnu Ada Translator).

gnats has been officially in beta status for years, but it is a vary mature package, used by such large organizations as the FSF, the FreeBSD.org and apache.org.

Figure 3 on page 90 shows the architectural block diagram of gnats.

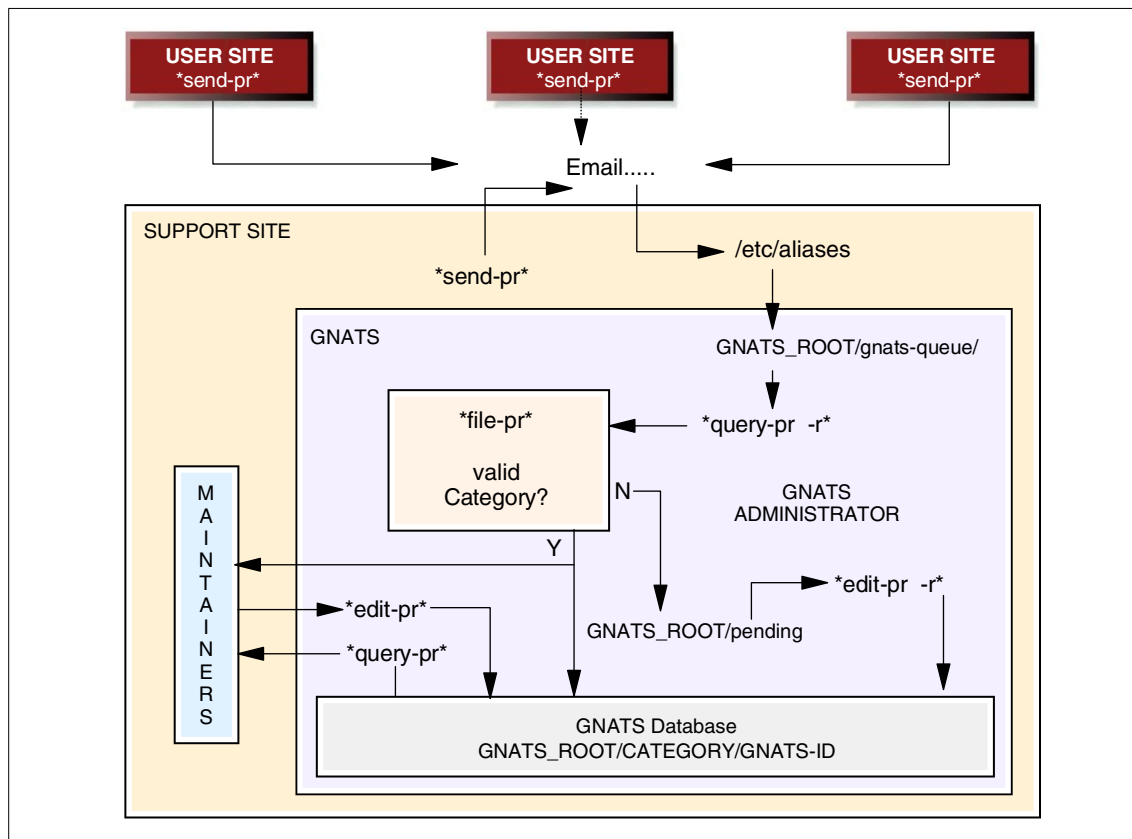


Figure 3. gnats block diagram

17.2 Installation of gnats

Unlike the other binary packages on our CD, the installation of gnats is complex and has many system administration prerequisites; therefore, we do not supply an installation script. Instead the CD has a small patch file containing some minor changes necessary to get a running system. For more information, you should get a copy of *“Keeping Track” - Managing Messages With GNATS - The GNU Problem Report Management System* from:

http://sourceware.cygnus.com/gnats/gnats_toc.html

17.3 Customizing and operating gnats

Do not start anything with gnats unless you already have a stable, working, UNIX-compatible mailing system. gnats prefers sendmail, but you are free to change this in the configuration.

Note: The central working mechanism of gnats is built on *mail aliases*, as explained in the documentation. If you must use a mailing system other than sendmail, you will have to figure out how to implement the same behavior.

You have to decide whether to use emacs or not—gnats is fully functional with any editor. Also, refer to the script send-pr to understand the use of other editors and mailers.

Be sure to set up cron to call queue-pr on a regular basis, in order to get the incoming reports into the database.

17.4 The gnats licence

The terms of use and redistribution of gnats are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

17.5 Getting gnats from the Internet

The gnats package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 3.113.

The package was originally found on the Internet at the GNU Web site. After unwinding the package the source files seemed to be very old, all dating from 1993. The INSTALL file states:

Installation procedures for GNATS

```
*** This is a very brief discussion of the installation procedure ***
*** for GNATS. It is strongly recommended that anyone attempting ***
*** to install and operate GNATS refer to the file gnats.texi.      ***
```

Since we did not have TeX and texinfo, we looked for more conventional documentation, and used the document *Keeping Track - Managing Messages*

With GNATS - The GNU Problem Report Management System, which we referred to previously.

While this document varied slightly from our version of the code, it was clear and detailed. After trying to use this release for a day, it was evident that some pieces of the package were missing (most notably the gnats daemon `gnatsd`). A search on the Web led us to the semi-official home page for gnats, <http://sourceware.cygnum.com/gnats/>, which has the following information:

```
GNATS - The GNU Bug Tracking System
GNATS is a freely redistributable set of tools for tracking bug reports.
It is one of the most robust bug tracking systems on the net, and is in
production use by many companies and free software projects.
```

There is a lot of confusion about the status of GNATS. The last official release of GNATS was version 3.2 in 1993. **However, GNATS 3.2 is really broken, don't use it.** Many major advances have been made to GNATS in the last couple of years, including multiple database support, customized fields, and comprehensive WWW and Tk based front-ends.

The next "official" release of GNATS will be version 4.0. We have a list of documentation updates and cleanup work that we'd like to see done before 4.0 is released. Until that time, we are releasing snapshots on a periodic basis.

Once we found this site, downloading the package, unwinding, configuring, building, and installing gnats went much better. It really works!

17.6 Porting gnats to OS/390

Use the following commands to compile gnats:

```
$ export A2E="-o from=ISO8859-1,to=IBM-1047"
$ export MYFLGS="-O2 -D_ALL_SOURCE -D_X_OPEN_SOURCE_EXTENDED=1 \
-DIS_EBCDIC_HOST"
$ export _CXX_CCMODE=1
$ export _CC_CCMODE=1
$ gzip -c -d tars/gnats-current.tar.gz | pax $A2E -r
$ cd gnats-current
$ gnupatch -p 1 <../patch-390port
$ CC=cc CFLAGS=$MYFLGS CCC=c++ CXXFLAGS="-+" ./configure
--host=$MACHTYPE
$ gmake;su;gmake install
```

Happy gnatsing!

Chapter 18. GNU grep

This chapter discusses the GNU grep open source software package.

18.1 Introduction

One of the most useful searching tools in UNIX is grep. It basically searches a file for a specified pattern and prints all the matching lines (by default).

The following description of grep is taken from the GNU README file:

“GNU grep is based on a fast lazy-state deterministic matcher (about twice as fast as stock Unix egrep) hybridized with a Boyer-Moore-Gosper search for a fixed string that eliminates impossible text from being considered by the full regular expression matcher without necessarily having to look at every character. The result is typically many times faster than Unix grep or egrep. (Regular expressions containing back referencing will run more slowly, however.)”

18.2 Installation of GNU grep

The GNU grep package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.4.2. Installing grep will store several files to the dictionary \$installDir/share/grep-2.4.2 (where ‘\$installDir’ is typically /usr/local.)

Here is the core of the install script:

```
gzip -c -d grep-2.4.2-pds-csi.pax.Z|pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s grep-2.4.2 grep
cd grep
export BOTHFLAGS="-O2 -D _ALL_SOURCE -DIS_EBCDIC_HOST"
CC=cc CCFLAGS=$BOTHFLAGS CCC=c++ CXXFLAGS="-+ $BOTHFLAGS" ./configure \
--prefix=$installDir
gmake
gmake install
```

18.3 Using grep

The usage is the same as GNU grep:

```
grep [options] pattern [file-list]
```

However, the file list can consist of absolute or relative paths in the HFS. Dataset identifiers, in single quotes, are prefixed by '//', with PDS or PDS/E member names in parenthesis. Also wildcard dataset identifiers. '*' and '**' are valid, similar to ISPF panel 3.4.

If the "recursive search" flag is enabled (via -r, --recursive or --directories=recurse), grep will search members of any PDS or PDS/E specified, including those that result from a wildcard expansion (catalog search).

Since single quotes are a special character to the shell, they must be escaped, either prefixed by backslashes or within double quotes.

Wildcards and parentheses don't mix. To search all members of a PDS, specify -r and put the data set name; there is currently no way to specify a subset of the members of a data set without listing each one, or to specify a particular member of all data sets matching a pattern.

18.3.1 Examples

Following is a simple search of a PDS member:

```
$ grep 620010 "'//sys1.parmlib(bafprm66)'" | sed 's/^ *//'  
TYPE(BPXUEINT)                                00620010
```

The following does not work because of the wildcard:

```
$ grep 620010 "'//sys1.parm*(bafprm66)'" | sed 's/^ *//'
```

The following example shows the use of the recursion, line number, case-insensitive and whole-word flags. Also, the file specification has a wildcard:

```
$ grep -miw exec "'//nkri*sh*.jcllib'"  
//'NKRISH.JCLLIB(FOO)':2://FOO EXEC  
PGM=FOO, PARM='ENVAR(_CEE_ENVFILE=DD:FOOENV)'/'  
//'NKRISH2.JCLLIB(BATCHTST)':2://BPXBATCH EXEC PGM=BPXBATCH,  
//'NKRISH2.JCLLIB(BPXCKSUM)':2://BPXBATCH EXEC PGM=BPXBATSL,  
//'NKRISH2.JCLLIB(SPAWNCMP)':2://COB EXEC PGM=IGYCRCTL,  
//'NKRISH2.JCLLIB(SPAWNCMP)':24://LKED EXEC PGM=IEWL,  
//'NKRISH2.JCLLIB(SPAWNCMP)':40://LKED2 EXEC PGM=IEWL,  
//'NKRISH2.JCLLIB(SPAWNTST)':2://RUNIT EXEC PGM=SPAWNTST,  
//'NKRISH2.JCLLIB(SPAWNCMP2)':2://COB EXEC PGM=IGYCRCTL,  
//'NKRISH2.JCLLIB(SPAWNCMP2)':24://LKED EXEC PGM=IEWL,
```

The following example lists names of files with matches, and searches both HFS and data sets:

```
$ grep -rl DD "'/'nkrish.jcllib(foo)'" /u/nkrish/jcl "'/'nkrish2.jcllib'"
/'nkrish.jcllib(foo) '
/u/nkrish/jcl/foo.jcl
/'nkrish2.jcllib(BATCHTST) '
```

18.4 The GNU grep license

The terms of use and redistribution of GNU grep are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

18.5 Getting gnu grep from the Internet

The GNU grep package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.4.2. It can be found on the Internet on:

<http://www.gnu.org/software/grep/grep.html>

18.6 Porting gnu grep to OS/390

On compiling grep, we encountered the following error:

```
IEW2456E 9207 SYMBOL alloca UNRESOLVED. MEMBER COULD NOT BE INCLUDED
FROM THE DESIGNATED CALL LIBRARY.
```

Unfortunately, `alloca()` is only made visible under that name if the language level is set to extended (via adding a `#pragma langlvl(extended)` to the code, or the `-W "c, langlvl(extended)` cmd line option) and `stdlib.h` is included. However, `LANGVLV(EXTENDED)` is set, then `__STDC__` is *not* defined. So we defined it ourselves (as follows) in the configure script, based on the code in `stdlib.h`:

```
#if defined __MVS__ && defined __COMPILER_VER__ && \
__COMPILER_VER__ >= 0x220600
# ifndef alloca
#  define alloca(x) __alloca(x)
#  ifdef __cplusplus
extern "builtin"
#  else
#   pragma linkage(__alloca,builtin)
#  endif /* defined __cplusplus */
```

```
void *__alloca(unsigned int x);  
# endif /* !defined alloca */  
#endif
```

In addition, we made a few changes for EBCDIC cleanups. After building the code, we typed in the following to install grep:

```
gmake install
```

This, too ran cleanly. We ran **gmake check** to run self-tests that come with the package, and all eight tests passed.

Chapter 19. Groff

This chapter discusses the groff open source software package. Groff is a document formatting system based on a device-independent version of troff. The groff tool is useful because there are no analogous troff or nroff commands shipped with OS/390 UNIX. The OS/390 built-in man command understands only plain text. If you want to port any packages from other UNIX systems to OS/390, you must find a preformatted version of the man pages because most man pages are shipped as troff/nroff source. Using plain text does not allow for colors and fonts such as bold/italics.

Having good documentation at hand is essential for effective work. This is why, among other things, the GNU project includes so many text formatting and online information utilities. The groff package was written by James Clark; see his notes at:

<http://www.jclark.com/bio.htm>

It is maintained by Ted Harding (Ted.Harding@nessie.mcc.ac.uk) and Werner Lemberg (w1@gnu.org). As a direct result of this redbook project, the OS/390 (EBCDIC) changes have been sent back into the open source stream.

19.1 Introduction to groff

The following excerpt describes groff well:

GNU Troff (or groff¹) is a system for typesetting documents. Troff is very flexible and has been in existence (and use) for about two decades. It is quite widespread and firmly entrenched in the UNIX community.

Groff is of an older generation of document preparation systems, which operate more like compilers than the more recent interactive WYSIWYG systems. Groff and its contemporary counterpart, TeX, both work using a batch paradigm: the input (or source) files are normal text files with the text and embedded formatting commands. These files can then be processed by groff to produce a typeset document on a variety of devices.

Likewise, groff should not be confused with a word processor, since that term connotes an integrated system which includes an editor and a text formatter.

¹ This text is from "This is Edition XX of Groff, for the XX version of the GNU implementation of Groff", archived on the Net at <http://www.cs.pdx.edu/~trent/gnu/groff/groff.html#SEC2> Copyright (C) 1994 Free Software Foundation, Inc. Published by the Free Software Foundation 675 Massachusetts Avenue, Cambridge, MA 02139 USA. Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

19.2 History

Troff can trace its origins back to a formatting program called runoff which ran on MIT's CTSS system. This name came from the common phrase of the time: "I'll run off a document." Also see Matt Welsh's article about groff on:

<http://www2.linuxjournal.com/lj-issues/issue18/groff.html>

19.3 Installation of groff

The groff package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 1.17. Installing groff will store several files to the dictionary `$installDir/share/groff` (where '`$installDir`' is typically `/usr/local`.)

Here is the core of the install script:

```
gzip -c -d groff-1.17.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s groff-1.17 groff
cd groff
patch -i ../configure.patch configure
export BOTHFLAGS="-O2 -D_ALL_SOURCE -DIS_EBCDIC_HOST"
CC=cc CFLAGS=$BOTHFLAGS CCC=c++ CXXFLAGS="-+ $BOTHFLAGS" ./configure
--prefix=$installDir
make
make install
```

19.4 Customizing and operation of groff

You can now use groff for your text-formatting tasks. This tool is important if you want to have nicely formatted man pages. Assuming the groff package was properly installed, the commands `groff` and `troff` should be in your PATH and ready to execute.

Because groff now has EBCDIC support built-in, you must use the device named `cp1047` in conjunction with the `-T` flag. Here is an example of using `groff` to display the `gzip` man page:

```
$ groff -Tcp1047 -man /usr/local/man/C/cat1/gzip.1
```

For ease of use, you can create a script to process both flat text and formatted man pages. You can name this script `/usr/local/bin/man` and put `/usr/local/bin` in your PATH before `/bin`.

The script could be the following:

```
#!/bin/sh

TMPFILE=/tmp/man$$
man $* 2> $TMPFILE
#grep FSUMF061 Cannot format /usr/local/man/man1/
file=`grep FSUMF061 $TMPFILE | cut '-d ' -f4`
rm -f $TMPFILE

if test -n "$file" ; then
    groff -Tcp1047 -man $file | less
fi
```

Having this script would give you a man command that could handle both flat text and nroff (or groff) formatted man pages.

19.5 License

The terms of use and redistribution of groff are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

19.6 Getting groff from the Internet

This package was found on the Web at:

<http://www.gnu.org/software/groff/groff.html>

19.7 Porting groff to OS/390

In this section we discuss several issues encountered during our port of groff 1.17.

19.7.1 A configure problem

The configure script has to be patched due to an OS/390 Language Environment (LE) problem related to static constructors and destructors. The patch file effectively *short circuits* the test for the success of these. Though static constructors and destructors do work properly in OS/390, the return code is not correct and the configure script fails. The resulting patch comments out the following if statement as follows:

```

#if { (eval echo configure:957: \"$ac_link\") 1>&5; (eval $ac_link) 2>&5; }
  && test -s conftest${ac_exeext} && (./conftest; exit) 2>/dev/null
#then
  echo "$ac_t"yes" 1>&6
#else
# echo "configure: failed program was:" >&5
# cat conftest.$ac_ext >&5
# rm -fr conftest*
# echo "$ac_t"no" 1>&6;{ echo "configure: error: a working C++ compiler
  is required" 1>&2;exit 1; }
#fi

```

In this fashion, configure always reports static constructors and destructors as working.

19.7.2 Minor problems

Usually you can build programs using `make`, but it was not possible to use it for `groff`, so we used `gmake`. When compiling, we encountered the error:

```

/u/tot62/src/groff/groff-1.17/src/include/lib.h", line 73.7: CBC1089(S)
More than one function "strncasecmp" has non-C++ linkage.
This prototype should be compiled out in the "configure" file. We
short-circuited this by adding the following in the configure file:

```

```

#if !defined(_AIX) && !defined(sinix) &&
!defined(__sinix__)&&!define( __MVS__ )
#ifdef HAVE_STRNCASECMP
#ifdef NEED_DECLARATION_STRNCASECMP
extern "C" {
    // SunOS's string.h fails to declare this.
    int strncasecmp(const char *, const char *, int);
}
#endif /* NEED_DECLARATION_STRNCASECMP */
#endif /* HAVE_STRNCASECMP */
#endif /* !_AIX && !sinix && !__sinix__ */

```

Another error, is due to the incorrect behavior of C/C++ binder under OS/390 V2.9. When this step fails, the partially generated file isn't deleted. When the make process is rerun, the partial file is considered as a good one, making the new process fail again (incomplete library found, incorrect executable, etc). In these cases, the file must be deleted manually. This bug is not present in V2.10.

In this case, a file `libgroff.a` was generated which was deleted manually.

A function in `tmpfile.cc`, `atexit` was called with the wrong type of parameters. We modified part of this code to obtain the same functionality. The original code is as follows:

```
static void add_tmp_file(const char *name)
{
    if (xtmpfiles_to_delete == NULL)
atexit(remove_tmp_files);
}
```

This was replaced by:

```
static void add_tmp_file(const char *name)
#ifdef __MVS__
{
    if (xtmpfiles_to_delete == NULL)
    {
        remove_tmp_files();
        atexit;
    }
#else
if (xtmpfiles_to_delete == NULL)
atexit(remove_tmp_files);
#endif
```

We found a bug in `/groff-1.17/src/utils/addftinfo.cc`, where a variable was declared twice, although in GNU C this would work because the scope of the variables was different. However, this does not work in the OS/390 C/C++ compiler and was preprocessed out by the use of `__MVS__` symbol.

Another error we received during compilation time was as follows:

```
"abs(int)" has already been defined.
CBC1793(I) Compilation failed for file ./pre-html.cc. Object file not
created.
```

Since the function `abs` is already part of the OS/390 system `"stdlib.h"` library, it was also preprocessed out by the use of the `__MVS__` symbol.

After removing the above bugs, the code built without errors.

19.7.3 Font troubleshooting

The `gmake` generates font description files from template files during the build process. You will see something like this:

```
gmake[2]: Entering directory `/u/sandorb/groff-1.15/devascii'
```

```

Making R
Making I
Making B
Making BI
Making DESC
gmake[2]: Leaving directory `/u/sandorb/groff-1.15/devascii'
...

```

You may have or want to use a codepage different from the one we use to build (the US English codepage IBM-1047). If you see some strange characters displayed in your new man pages, it is definitively due to that. We supply you with a method to convert the font files into other codepages. You have to start with the file R.proto.ascii from the CD. Transfer it to OS/390, converting, as usual, to EBCDIC text. There is a small Perl program on the CD, named to a256.pl:

```

#!/usr/local/perl5/bin/perl -w
#
#
$Ascii[255]=0;
for ($i = 0; $i < 256; $i++) {

    $Ascii[$i] = chr($i);

}
print @Ascii;

```

Call this program with the following command:

```
/usr/local/perl5/bin/perl a256.pl >ascii.tab
```

You now will have a file named ascii.tab, containing 256 binary character from 0 to 255. Convert this file with the OS/390 `iconv` utility into another file, named ebcdic.tab, containing the corresponding EBCDIC characters. Use the following command:

```
iconv -f ISO8859-1 -t YOUR_CODEPAGE ascii.tab >ebcdic.tab
```

There is another Perl program on the CD, named a2efont.pl:

```

#!/usr/local/bin/perl
#

open(CONV, '<ebcdic.tab') or die "cannot input conv table";
read(CONV, $Ebcdic, 256);

while(<>)
{

```

```

if (/^(.*) (\d\d\d\d)$/)
{
    $Start=$1;
    $ascii=oct($2);
    $ebcdic=substr($Ebcdic,$ascii,1) ;
    $o=sprintf "%o", ord("$ebcdic");

    print "$Start" . "0" . "$o" . "\n";

}
else
{
    print $_;
}
}

```

This program can read in the original (ASCII) font description file, and convert it to the codepage of your choice. (It uses the auxiliary file `ebcdic.tab`, made by you in the second step earlier). Use the following command to call it:

```
$ a2efont.pl >R.proto <R.proto.ascii
```

Copy the output file to the `font/devascii` directory, and call `gmake`.

```

$ cp R.proto font/devascii/
$ cd font/devascii/
$ rm B BI DESC I R
$ gmake

```

The `gmake` will rebuild your font files. You can test them with the script `test-groff`:

```

$ cd groff-current
$ ./test-groff -man -Tascii groff/groff.n | less

```

If you are satisfied with the result, you have to reinstall the package to make your changes global.

Chapter 20. Gzip

This chapter discusses the *gzip* open source software package.

20.1 Introduction to gzip

Gzip is GNU's program for compressing and decompressing files.

Because of the requirement to save network bandwidth and space on disk volumes, there are a number of tools which compress source code to only a fraction of its original size. These compressed files are not usable until they are uncompressed by the appropriate utility. But they can be stored and transferred to any other platform in their compressed state.

Gzip reduces the size of a named file by using a special compression algorithm (Lempel-Ziv coding, LZ77). Whenever possible it will add a prefix of *.gz* to the compressed file. All information about the original name, ownership, permission bits, access and modification times are kept within the compressed file and can be used in the uncompression process.

Gzip was written to automatically recognize the format of compressed files; it can currently uncompress files created by *gzip*, *zip*, *compress* or *pack*.

The open software foundation provides most of the packages under their GNU license as compressed gzip archives, so this tool is a requirement for anybody who wants to install open software on their platform.

20.2 Installation of gzip

The version of the gzip package on the CD is 1.2.4. It is provided as a tar file which contains ASCII source code. You have to ensure that the code is translated to EBCDIC during the installation process.

```
$ export A2E="-o from=ISO8859-1,to=IBM-1047"
$ pax -rf gzip-1_2_4.tar $A2E
```

It is easy to build gzip on OS/390. The configure script creates the Makefile and the `make` command builds the executables:

```
$ cd gzip-1.2.4
$ ./configure
checking for gcc
checking how to run the C preprocessor
checking for underline in external names
...
```

```

checking for string.h
creating Makefile
$ make
/bin/c++ -c -DHAVE_UNISTD_H=1 -DDIRENT=1  gzip.c
...
rm -f gunzip zcat
ln gzip gunzip
ln gzip zcat

```

When it is built, there is really only one executable—`gzip`, which is created in the same directory as the source code. Hard links are created from the commands `gunzip` and `zcat` to the `gzip` executable.

The `gzip.install` script included on the CD will build `gzip`. See Chapter 3, “What’s on the CD-ROM” on page 17.

20.3 Customization and operation of gzip

The only necessary configuration is to make the commands available for all users. Do this either by creating symbolic links into a directory which is in the `PATH` variable of all users, or by copying the `gzip` program into this directory.

Keep in mind that you may use superuser authority to update this shared directory. The path `/usr/local/bin` is only an example and you can replace it with a correct path for your installation.

```

$ cd /usr/local/bin
$ ln -s /usr/local/gzip/gzip-1.2.4/gzip gzip
$ ln -s /usr/local/gzip/gzip-1.2.4/gunzip gunzip
$ ln -s /usr/local/gzip/gzip-1.2.4/zcat zcat

$ gzip --version
gzip 1.2.4 (18 Aug 93)
Compilation options:
NO_DIR UTIME HAVE_UNISTD_H

```

Or :

```

$ cp gzip /usr/local/bin/gzip
$ cp gunzip /usr/local/bin/gunzip
$ cp zcat /usr/local/bin/zcat

```

The command `gzip` takes filenames as parameters and compresses them. Each file is renamed to include the suffix `.gz`. The command `gunzip` uncompresses a `gzip` compressed file. Here is an example of each:

```

$ ls -l README

```

```

-rw-r--r--  1 MIKEM    SYS1          7299 Aug 10  1993 README
$ gzip README
$ ls -l README*
-rw-r--r--  1 MIKEM    SYS1          3400 Aug 10  1993 README.gz
$ gunzip README
$ ls -l README*
-rw-r--r--  1 MIKEM    SYS1          7299 Aug 10  1993 README

```

If the file does not have a .gz suffix, or it is not compressed, gunzip will not alter the file:

```

$ gunzip README
gunzip: README: unknown suffix -- ignored
$ mv README README.gz
$ gunzip README.gz

gunzip: README.gz: not in gzip format

```

The command `zcat` does an uncompress of a compressed file and sends the output to stdout. It looks for a .Z suffix:

```

$ compress README
$ ls -l README*
-rw-r--r--  1 MIKEM    SYS1          4174 Feb 14 16:17 README.Z
$ zcat README.Z > README
$ ls -l README*
-rw-r--r--  1 MIKEM    SYS1          7299 Feb 14 16:17 README
-rw-r--r--  1 MIKEM    SYS1          4174 Aug 10  1993 README.Z

```

A full set of documentation about gzip can be found at:

<http://www.gnu.org/manual/gzip-1.2.4/gzip.html>

20.4 License

Gzip is covered by the GNU GPL, which is explained in detail in Appendix A, “The GNU General Public License” on page 289.

20.5 Getting gzip from the Internet

The version of the package on the CD is 1.2.4. It was found on one of the GNU mirror Web servers at the following URL. Please select a server close to your location:

<http://www.gnu.org/order/ftp.html>

GNU also offers a tar file in a gzip format. Please ensure that you already have another version of gzip available before downloading a .gz file. For a first time user it's easier to download the unzipped tar file.

If you insist on downloading the compressed file and if you have gzip available on another platform, you can always do the gunzip on that system and transfer the uncompressed file back to OS/390.

20.6 Porting gzip to OS/390

There is no porting necessary for this product. It can be used as is from the GNU Web site.

Chapter 21. Id-utils

This chapter discusses the id-utils open source software package.

21.1 Introduction to id-utils¹

The id-utils package allows the creation of an ID database (binary file) containing a list of file names, a list of tokens, and a sparse matrix, indicating which tokens appear in which files. With this database and some tools to query it, many text-searching tasks become simpler and faster.

For example, you can list all files that reference a particular `#include` file throughout a huge source hierarchy, search for all the memos containing references to a project, or automatically invoke an editor on all files containing references to some function or variable. Anyone with a large software project to maintain or a large set of text files to organize can benefit from the ID utilities.

Although the name ID is short for identifier, the ID utilities handle more than identifiers only; they also treat other kinds of tokens, most notably numeric constants, and the contents of certain character strings. Therefore, this redbook will use the word *token* as a term that is inclusive of identifiers, numbers and strings.

The maintainer of id-utils is Greg McGary (gkm@gnu.org).

21.2 Installation of id-utils

If you want to install the binary distribution from the CD, refer to Chapter 3, “What’s on the CD-ROM” on page 17.

The tool `gunzip` is first needed to install `id-utils`.

Following is the core of the install script named `idutils.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d id-utils-3.2d-390.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s id-utils-3.2d-390 idutils
cd idutils
CC=cc CFLAGS=-D_ALL_SOURCE _CC_CCMODE=1 ./configure --prefix=$location
make
cd src
```

¹ Quoted from *ID database utility* by Greg McGary and Tom Horsley; used with permission from the authors.

```
echo "copying executables and man pages to $installDir/bin and man ..."  
for i in xtokid fnid fid lid mkid  
do  
    cp $i $installDir/bin  
    cp $i.1 $installDir/man/$LANG/cat1  
done
```

21.3 Customization and operation of id-utils

Id-utils uses the file `id-lang.map` to map filenames to a language. This file is normally located at `/usr/local/share/id-lang.map`. The comment at the beginning of the file explains the format.

For example, if you want a file with the extension `sqc` to be processed as a C file, you need the following line in this file:

```
*.sqc      C
```

21.4 License

The terms of use and redistribution of id-utils is governed by the GPL. See Appendix A, "The GNU General Public License" on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available, and you cannot put any restrictions on further redistribution.

21.5 Getting id-utils from the Internet

The id-utils package is available on the CD that accompanies the hardcopy of this redbook. This is the port of Version 3.2d, which is a bugfix release of 3.2. The ported package is currently not available on the Web.

21.6 Porting id-utils to OS/390

Porting id-utils was not that difficult. The main problem here was, like always, related to EBCDIC/ASCII conversion. The scanners of the program `mkid` have a table defining which characters can be part of an identifier, which are numbers, and so forth. For OS/390 UNIX, we had to add EBCDIC-specific tables.

In one file we found numeric character values, which had to be replaced with defines.

For conditional compilation, we added a test for EBCDIC to the configure.in script (see 50.1.4, “Setting the DISPLAY variable for X Window applications” on page 273).

Chapter 22. Info-ZIP

This chapter discusses the Info-ZIP open source software package. Unfortunately, due to legal concerns, this package cannot be distributed with the redbook.

22.1 Introduction to Info-ZIP

This package includes Info-ZIP's freeware zip and unzip packages. These packages process compressed, archived files that are compatible with PKZIP and Winzip.

22.2 Installation of Info-ZIP

The following commands can be used to build the Info-ZIP package (where `$installDir` is typically `/usr/local`):

```
pax -rzf infozip-os390.tar.Z
ln -s infozip-os390 infozip
cd infozip
ln -s zip-2.2 zip
ln -s unzip-5.32 unzip
cd zip
make -f unix/Makefile os390
for i in zip zipcloak zipsplit zipnote
do
    cp $i $installDir/bin
done
cp zip-2.2/MANUAL $installDir/man/$LANG/cat1/zip.1
cd ../unzip
make os390
make install
```

22.3 Customization and operation of Info-ZIP

Assuming two files are in a directory named `hello` and `hello.o`, here is an example of using `zip`, moving the zip file to another directory, and then using `unzip`:

```
$ zip hello.zip hello hello.o
  adding: hello (deflated 74%)
  adding: hello.o (deflated 68%)
$ mv zip foo
mv: zip: EDC5129I No such file or directory.
```

```
$ mv hello.zip foo
$ cd foo
$ unzip hello.zip
Archive:  hello.zip
  inflating: hello
  inflating: hello.o
$ ls
hello.zip  hello*      hello.o
```

22.4 License

The terms of use and redistribution of zip are governed by a unique copyright. The following text is from the beginning of the zip README file:

```
Copyright (C) 1990-1997 Mark Adler, Richard B. Wales, Jean-loup Gailly,
Orno van der Linden, Kai Uwe Rommel, Igor Mandrichenko and John Bush.
Permission is granted to any individual or institution to use, copy, or
redistribute this software so long as all of the original files are
included, that it is not sold for profit, and that this copyright notice
is retained.
```

The terms of use and redistribution of unzip are governed by a number of copyrights. The following text is from the beginning of the unzip COPYING file:

```
There are currently six explicit copyrights on portions of UnZip
code (at least, of which Info-ZIP is aware): the original Sam Smith
copyright on unzip 2.0, upon which Info-ZIP's UnZip 3.0 was based;
Igor Mandrichenko's copyright on his routines in vms.c; Greg Roelofs'
copyright on zipinfo.c and the new version of unshrink.c; Mike White's
copyright on the Windows DLL code (windll/*); Steve P. Miller's
copyright on the Pocket UnZip GUI (wince/*); and Norbert Pueschel's
copyright on the Amiga time.lib code. In addition, Mark Adler has
placed inflate.h, inflate.c, explode.c and funzip.c into the public
domain; i.e., these files may be used without any restrictions beyond
those of simple courtesy (credit where it's due). All of these are
discussed immediately below. The remaining code is covered by an im-
plicit copyright under US law. Frequently Asked Questions regarding
(re)distribution of Zip and UnZip are near the end of this file.
```

There are more details in this file.

Due to the restrictive statements in some of the licenses, Info-ZIP is *not* part of the software package on CD.

22.5 Getting Info-ZIP from the Internet

A commonly used release of Info-ZIP consists of zip version 2.2 and unzip version 5.32. There appear to be later versions available, such as zip 2.3 and unzip 5.4, but we used the package from the Tools and Toys “Ported Tools” Web page:

<http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxalty1.html>

The package source code was found at:

<ftp://ftp.software.ibm.com/s390/zos/tools/infozip/infozip-os390.tar.Z>

The package was downloaded to the PC, and then moved to OS/390 in binary via FTP. The text files are in EBCDIC, so it was unwound with the following command:

```
$ ls -l
total 3920
-rw-r--r--  1 MIKEM    SYS1      2003747 Feb 16 08:03 infozip-os390_tar.Z
$ pax -rzf *.Z
$ ls -l
total 3952
-rw-r--r--  1 MIKEM    SYS1      2003747 Feb 16 08:03 infozip-os390_tar.Z
drwxr-xr-x  4 MIKEM    SYS1           8192 Apr 28 1998 infozip-os390/
```

22.6 Porting Info-ZIP to OS/390

The OS/390 changes were made by Paul Von Berhen of Storage Technologies Inc. Some notes were found in the README.OS390 files.

22.6.1 Changes in the zip package

No notes were found regarding the changes made for OS/390. Certainly, greping through the code for the strings MVS and EBCDIC point to areas where changes were made.

22.6.2 Changes in the unzip package

Again, greping through the code for the strings MVS and EBCDIC show that a fair number of changes were made for OS/390. Also, the following notes were found in README.OS390:

Source Changes

ebcdic.h - the mapping table had bad values for CR and LF - in case of compatibility issues with CMS or MVS ports, I ifdef'd my changes.

unzpriv.h - I had to explicitly define 'near' and 'far' - I'm not sure why these didn't get defined, perhaps `__IBMC__` is defined by the compiler. This is `ifdef`'s OS390.

unix/Makefile - I added an 'os390' target. OS390 is very standards-oriented so I started by defining `SYSV` and `EBCDIC` - apparently the compiler does not define `UNIX`, so I added it. I also needed to define `NO_PARAM_H` to get rid of warnings (apparently nothing from `param.h` was needed). I also overrode `LF2` to eliminate the `-s` option - on OS/390, all options must precede the filenames. OS390 does not have a `strip-executable` capability.

These comments give you an idea of some of the changes that were necessary to port this package to OS/390.

Chapter 23. Itools

This chapter discusses the itools open source software package.

23.1 Introduction to itools

The itools package is one of the methods you can use to create one or more Makefiles with system/platform configuration options. (Other methods include GNU autoconf/automake, described in Chapter 6, “Autoconf” on page 43 and Chapter 7, “Automake” on page 49, and creating the Make file manually using an editor.) The imake utility, itools' most significant component, originated with the X11 (X Window) project.

The source version of imake (and its companion commands) was obtained from www.primite.wisc.edu in the itools-R6.3 release, dated April, 1997. Credit is given to Paul DuBois (dubois@primate.wisc.edu), who is the present maintainer of the itools package.

23.1.1 Imake versus automake/autoconf

The philosophy of imake is rather simple: merge hard-coded variable values together using the C compiler's preprocessor to arrive at one or more Makefiles. This philosophy is at odds with that of automake/autoconf, which attempts to dynamically determine what the variable values should be. In other words, the person porting an imake-configurable package to a new platform for the first time must do all the detailed work of researching how 'make' rules must appear, and then set up the hard-coded configuration files to cause the proper results. Fortunately, these configuration files tend to vary little between packages, so once you've done the work for your system, you rarely need to concern yourself with reconfiguration. Fortunately for OS/390 UNIX users, most of this detailed work was done as a result of this port.

For creation of a new package, the choice between using imake or automake/autoconf is one of not only personal preference, but also portability. Imake has none of the GNU/gcc dependencies of automake/autoconf, and hence tends to produce results more evenly on systems that land anywhere on the XPG4/POSIX conformance scale. On the other hand, the person receiving the automake/autoconf configurable package needs to know almost nothing about the characteristics of their system, regardless of whether or not the package has already been ported to it. In our experience with OS/390 UNIX, imake is a much better choice, since there are fewer exceptions to consider. However, imake is much less popular than automake/autoconf, so those who need to port packages to and from OS/390 UNIX should be aware

of and knowledgeable about both paradigms. The imake program is also "make-neutral"—it works well with all known versions of make, including GNU make as well as the make that comes pre-installed with OS/390 UNIX.

The automated determination of compile-time dependencies (that is, header files, if changed, should trigger a rebuild of a C language object file) is also much better supported in the OS/390 port of imake as opposed to the GNU automake utility. The GNU package critically depends on the -M switch option of the gcc compiler, for which the OS/390 C/C++ compiler provides neither direct support nor similar function. Hence, the use of GNU automake under OS/390 infers that all dependency rules in Makefiles must be manually created and maintained. On the other hand, imake includes the *makedepend* module, which works dependably (if not terribly efficiently) under OS/390. If both automake and imake are ported to your OS/390 system, you may choose to modify *depend2.am* in the automake system to call *makedepend* instead of setting `AUTOMAKE_OPTIONS = no-dependencies`.

Another interesting possibility is that of "marrying" autoconf and imake, such that imake *project* files could be configured through GNU Autoconf (see 23.1.3, below).

23.1.2 Imake and the X Window system

Imake was developed to permit the X Window system to be built on many different UNIX and UNIX-like systems, such as Linux and Windows NT. The imake configuration files, as distributed, make a rather bold assumption that the X Window support code (header files and libraries) are available in the standard places and forms. This definitely does not hold true for OS/390 UNIX. Part of the port to OS/390 was to remove the *X11.rules* and *X11.tmpl* files from their normal position in the imake hierarchy; no effort was undertaken to customize them for the OS/390 X installation quirks. The first person who attempts to reintroduce the X11 files into the OS/390 port of imake will certainly have a job on their hands.

23.1.3 Vanilla versus 'extensible architecture' imake

The authoritative reference on imake is a book called *Software Portability With imake*¹ (hereafter called SPWI), which describes the 'extensible architecture' (EA). EA permits imake configuration files to be much more usable for development projects. SPWI dedicates two entire chapters to defining exactly how the EA file structure should look; but, the imake package provides none of it. Therefore, the changes for EA are an individual post-installation activity, through which the SPWI book will guide you.

¹ See Appendix F, "Related publications" on page 311 for the complete reference.

It should be remembered that the *base* (non-EA) imake is *all* that a portable distributed package should be based upon. However, EA imake is a much more useful paradigm for internal development, and spreads among diverse programming teams very well. If you plan to base OS/390 UNIX development upon imake, you are advised to obtain the SPWI book and follow the EA instructions.

23.1.4 Imake versus itools

The imake command creates a Makefile from a file called 'Imakefile'. The process of creating an initial Imakefile through its final metamorphosis into a usable Makefile and the commands the Makefile itself may invoke in its rules and targets involves several commands which are part of the itools package:

- xmkmf
- imboot
- makedepend
- mkdirhier
- which
- imake
- msub

23.1.5 An example of imake

Following is an example of using imake to create a Makefile:

```
$ imake -I /usr/local/itools/itools-R6.3/config/cf -f Imakefile.in
```

23.2 Installing itools

The following software is prerequisite to the build and install of itools:

- Perl 5.004 or higher.
- An `install` command. If you don't already have one, the easiest way to get it is to unpack `install-sh` from the autoconf package (see Chapter 6, "Autoconf" on page 43), rename it to `install`, and place it somewhere in your PATH. Don't forget to `chmod` it to mode 0755.
- OS/390 C/C++ compiler.`make`

Before we begin the build of the itools package, we first need to build the `cpp` wrapper program, which is required for all phases of the build.

```
cd ${BASE}/itools-R6.3/cpp
```

```

(edit Makefile)
set INSTALL_DIR to equal the absolute path
value for some directory in your PATH.
(save Makefile)
make install

```

Now we can begin to configure itools for the build:

```

cd $BASE/itools-R6.3/config/cf (edit file site.def)
Set variable 'ProjectRoot' to ${BASE}/itools-R6.3
Set variable 'ConfigDir' to the directory in which you want the
configuration files installed.
Set variable 'BinDir' to the directory into which you wish the itools
binary executable files installed.
(save file site.def)
cd $BASE/itools-R6.3/config/extras
(edit file extras.tmpl)
Set variable CONFIGROOTDIR to the same value that you set site.def
ConfigDir to.
(save file extras.tmpl)
cd $BASE/itools-R6.3

```

Now we can commence the build:

```
make World BOOTSTRAPFLAGS="-Dos390"
```

Note that the double-quote pair is important and must not be omitted. This begins the build, which will compile many programs (including a *bootstrap* copy of *imake*), and will run for at least a few minutes, depending on the speed of your system.

A couple of errors will occur during the build: one in the compilation of *Indir.c* (which relates to a missing header file, *sys/param.h*), and another in an attempt to build *imboot.man* (which relates to the absolute refusal of the C preprocessor to handle troff macro symbols). Neither of these errors has an effect on the final product, so they can be safely ignored. The build will continue despite them.

Once the build has completed, you need to choose a location into which you'll store the final completed binary executables. You should have done this in a previous step, and set file *site.def*'s *BinDir* variable to this location. We'll refer to this location as *\$(bindir)* below.

Now you can install the products of the build:

```
make install BINDIR=$(bindir)
```

where *\$(bindir)* is the value you hopefully checked above.

Once you've finished the install, go back to `$(ConfigDir)/config/cf/site.def` and be sure you can live with the values of `ProjectRoot`, `BinDir`, and `ConfigDir`; change these *now* if necessary.

Important

The `ProjectRoot` value should be changed to point to whatever `ConfigDir` value you used during the build. Should this value prove inconvenient for some later user, that user can always override it in the `imake` file, even if the EA architecture is not in use.

23.3 Customization and operation of itools components

The documentation included with the `itools` package leaves quite a bit to be desired, and only consists of a handful of man pages that don't explain its components very well, if at all. `Imake` (and its cousins) are extremely powerful tools, with operational details that would take a full book to explain. Such a book is *Software Portability with Imake* by Paul DuBois. We strongly recommend that you study the material in this book before attempting to write `Imakefiles`.

There is also a handy little guide that tells which `imake` variables are defined and where they are defined, as well as the suggested method of override. This can be found in file `$BASE/itools-R6.3/config/cf/README`, and is an invaluable tool once you've read SPWI and start getting into issues about what's appropriate to change where. Be sure you isolate this file before you destroy the installation file source tree, though—it doesn't install anywhere.

23.4 License

The `itools` package is covered by what is commonly known as the "X11 License", which appears here:

Copyright (c) 1993, 1994 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

As it says, this software is free to modify and distribute, and comes with the usual "no warranty" disclaimer. The changes made for the OS/390 port of imake have *not* been placed back into the open source stream.

23.5 Getting itools from the Internet

Visit Paul DuBois' Web site at:

<http://www.prima.te.wisc.edu/software/imake-stuff>.

23.6 Porting imake to OS/390

The most significant barrier in porting imake to OS/390 was compensating for the OS/390 C/C++ compiler. imake critically and crucially depends on flexible function of the C preprocessor phase. Most C preprocessors can be invoked with the `cpp` command; but this facility is not present on OS/390. Instead, the `-E` switch is usable with the OS/390 C/C++ compiler (`c89`, `cc`, `cxx`, `c++` commands) to stop processing after the C preprocessor phase. However, the OS/390 implementation of the compiler has three serious variant² behaviors that needed to be compensated for in the port:

- The habit of outputting a line at the head of the output that implements a call to `#pragma filetype()`, using ANSI-C trigraphs
- The inability to accept piped input via `stdin`
- The inability to accept input files with suffixes other than `.c`

² Perhaps the term *variant* isn't fair—the OS/390 C/C++ compiler, including its preprocessor, adheres exactly to the spirit of all applicable standards, including XPG4, POSIX, and ISO-C. However, these standards all reflect a lowest-common-denominator behavior, which most other compilers/preprocessors have far exceeded. When we say variant here, what we mean is "different from the rest."

The first of these behaviors was probably the most serious; imake uses `cpp` to generate many things, up to and including shell scripts and perl programs. Unfortunately, these other things are very intolerant of `#pragma filetype` lines. Therefore, we had a choice: either port in a foreign C preprocessor, or massage the input to and output from the OS/390 C preprocessor. The end decision was to 'wrap' the OS/390 C preprocessor in a command process called `cpp`, which was custom-written for this port. As a result, the `cpp` program must be built prior to installation. This program is capable of handling `stdin` input (which is used by some of the macros in the various `.rules` and `.tmpl` files), adjusting the `_C89_CSUFFIX` environment variable to match the file being processed, and also filters out the `#pragma filetype` output.

The next task was to define the OS/390 platform type and create `os390.rules` and `os390.cf` files, which customizes imake for a 'typical' OS/390 UNIX installation. Two main goals were implicit in creating the OS/390 configuration files:

- Redefine macros which caused C preprocessor failures. In the OS/390 C preprocessor, it is a fatal error to call a preprocessor macro with a null argument. The imake macro `NullParameter` defines to the empty string, and is present in many imake macros. While we can successfully call the *first* macro with `NullParameter`, any attempt to pass that value forward into *subroutine* macros causes the preprocessor to fail. During initial testing, most of the imake macros determined to cause failure were those that supported the existence of third party SCM tools, such as CodeCenter, Sentinel, and ClearCase. Since the typical "vanilla" OS/390 system does not yet include these tools³, heavily-used imake macros that call subroutines to support these tools were redefined to omit the subroutine calls.
- Remove `X11.rules` and `X11.tmpl` from the "standard" rules and template file `#include` hierarchy. OS/390 UNIX does not support X Windows in anything resembling a standard manner: header files and libraries are either missing or not placed in segregated subdirectories. Furthermore, X programs don't build well with the OS/390 compiler using the standard `-l` command line switch when referring to the `X11 *.a` libraries; X programs should be dynamically linked with the `/usr/X11/*.x` definition side decks instead. Rather than attempt to redefine macros that an OS/390 package would rarely (if ever) use, it was deemed more expedient to remove the referring rules and template files entirely from the hierarchy. Included in this task was the definition of macros present in `X11.*` and assumed to be present in later rules/template files. These appear in file

³ We had to direct this port to some baseline system. None of the mentioned products have OS/390 UNIX versions available yet, and were therefore safe things to pick on. Should someone port these products to OS/390, this port will have to be modified.

config/cf/os390.rules. Using this port to build actual OS/390 X Windows programs will probably prove difficult. Therefore, we recommend that any user of this port create their own X compile and linking imake macros in file site.def, or extend files os390.rules and/or os390.tmpl to suit their purposes.

- Support the installation of the custom `cpp` command, whose reasons for existence have already been discussed.

Unlike most other ports we attempted, there were no “ASCII-isms” present in the imake package, since most substitution of terms is relegated to the C preprocessor.

23.7 Caveats

The OS/390 implementation of X11 is (stating it diplomatically) slightly non-standard. Two included configuration files, X11.tmpl and X11.rules, delve into details about building X programs. This port specifically avoids any and all X11 issues, and did not even attempt to verify the operation of any X11 rules, headers, or libraries.

Chapter 24. Jpeg library

This chapter discusses the jpeg library open source software package. Unfortunately, due to legal concerns, this package cannot be distributed with the redbook nor is it on the Tools and Toys Web site.

24.1 Introduction to the jpeg library

The jpeg library is a programming tool to generate .jpg images from your programs. It is required for graphic packages such as Webalizer and the gd library.

This library contains C code to implement JPEG image compression and decompression. JPEG is a standardized compression method for full-color and gray-scale images. JPEG is intended for compressing “real-world” scenes; line drawings, cartoons and other non-realistic images are not its strong suit. JPEG is lossy, meaning that the output image is not exactly identical to the input image. Hence you must not use JPEG if you need to have identical output bits.

However, on typical photographic images, very good compression levels can be obtained with no visible change, and remarkably high compression levels are possible if you can tolerate a low-quality image.

The complete description of the package can be obtained on the Internet at:

<http://www.ijg.org/>

24.2 Installation of the jpeg library

The gzip command is needed to install the jpeg library.

Here are the commands that can be used to install the jpeg library:

```
gzip -c -d jpegsrc.v6b.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s jpeg-6b jpeg
cd jpeg
chmod u+w *
./configure --prefix=/usr/local
make
make install
```

24.3 Usage of the jpeg library

JPEG is *not* going to displace GIF entirely; for some types of images, GIF is superior in image quality, file size, or both. One of the first things to learn about JPEG is which kinds of images to apply it to. Generally speaking, JPEG is superior to GIF for storing full-color or gray-scale images of “realistic” scenes, meaning scanned photographs, continuous-tone artwork, and similar material. Any smooth variation in color, such as occurs in highlighted or shaded areas, will be represented more faithfully and in less space by JPEG than by GIF.

Examples illustrating the use of this package are the GD library and Webalizer. See Chapter 15, “Gd” on page 81 and Chapter 47, “Webalizer” on page 261. Also refer to the code.

24.4 License

The following license was found in the README file:

“The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided “AS IS”, and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that “this software is based in part on the work of the Independent JPEG Group”.

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution."

24.5 Getting the jpeg library from the Internet

You can download the jpeg library from the Internet site starting at:

<http://www.ijg.org/>

The package is in ASCII, .gz format, and was unwound using the following commands:

```
$ gunzip jpegsrc.v6b.tar.gz
$ pax -rf jpegsrc.v6b.tar -o from=ISO8859-1,to=IBM-1047
```

24.6 Porting the jpeg library to OS/390

No code changes were needed; the library compiles cleanly on z/OS UNIX.

Chapter 25. Korn shell

This chapter discusses the Korn shell software package. Unfortunately, due to legal concerns, this package cannot be distributed with the redbook nor is it on the Tools and Toys Web site.

25.1 Introduction to the Korn shell

The Korn shell is an interactive command language. Ksh interprets this language and can be used as an alternative to the standard OS/390 UNIX shell to interact with the system. It is intended to conform to the Shell Language Standard (IEEE POSIX P1003.2).

The Korn shell language is also used to write prototypes or even full applications. It combines much of the functionality of other popular scripting languages such as awk, Perl or REXX.

25.2 Installation of the Korn shell

To install the Korn Shell, we downloaded the INIT and ast-ksh packages from the Internet and unwound it with the following command (where the environment variable A2E is set to “-o from=ISO8859-1,to=IBM-1047”):

```
$ ls -l
total 0
-rw-r--r--  1 MIKEM   SYS1   212020 Jan 17 09:16 INIT.2001-10-31.tgz
-rw-r-----  1 MIKEM   SYS1  1328492 Jan 16 16:01 ast-ksh.2001-10-31.tgz
$ gzip -c -d INIT.2001-10-31.tgz | pax $A2E -r
$ gzip -c -d ast-ksh.2001-10-31.tgz | pax $A2E-r
$ ls -F

NOTICE -- LICENSED SOFTWARE -- SEE README FOR DETAILS

INIT.2001-10-31.tgz
README
ast-ksh.2001-10-31.tgz
bin/
lib/
src/
```

The INIT package contains the **package** command in the ./bin directory. That script is used to install all packages.

```
$ bin/package make ast-ksh
```

The build process continues for some time (perhaps an hour). Everything built cleanly for us. When complete, the following commands were available in `./arch/mvs.390/bin`:

```
$ls arch/mvs.390/bin
cc      filter  ksh      ok      proto    regress  shcomp
crossexec  iffe      mamake   package  ratz     release  suid_exec
```

The file `arch/mvs.390/bin/ksh` is the Korn Shell.

25.3 Customization and operation of the Korn shell

Type `ksh` to start the Korn shell command language interpreter (The executable file must be in your `PATH`). The behavior of `ksh` can be customized in many ways. Edit one of the three start-up scripts: `/etc/profile`, `~/.profile`, or the file the variable `ENV` points to, if you want to set shell options permanently.

Features you might notice right at the beginning are emacs- or vi-like key bindings, filename completion, command line history. To start `ksh` with vi-like key bindings start `ksh -o vi`. If you also want command and file name completion working with the `TAB` key, enter the command `ksh -o vi -o viaw`. A man page for `ksh` is available with `ksh --help`.

You can change your default shell with the `TSO` command:

```
ALTUSER USERNAME OMVS (PROGRAM('pathToShell'))
```

25.4 License

The Korn shell is covered by the AT&T ast open source license. It can be found online at:

<http://www.research.att.com/sw/license/ast-open.html>

25.5 Getting the Korn shell from the Internet

The Korn shell software can be downloaded from:

<http://www.research.att.com/sw/download/>

You need the packages *INIT* and *ask-ksh*. The version of the package we worked with is dated October 31, 2001.

Chapter 26. Less

This chapter discusses the less open source software package.

The `less` command is a pager that allows you to browse files. It is similar to the `more` command, but it has additional function. So rather than saying “less is more”, we can say “less is more than more”. It has some useful extensions, the most important of which is the ability to go backwards in the file you are browsing.

Less was written by Mark Nudelman (markn@greenwoodsoftware.com). You can find the official home page for less on the Web at:

<http://www.greenwoodsoftware.com/less/>

26.1 Introduction to less

The following descriptions are excerpts reprinted with permission from the author, Mark Nudelman. See the whole text on the Web at:

<http://www.greenwoodsoftware.com/less/#what>

What less is

Less is a pager. A pager is a program that displays text files. Other pagers commonly in use are `more` and `pg`.

Less is not an editor. You can't change the contents of the file you're viewing. Less is not a windowing system. It doesn't have fancy scroll bars or other GUI (graphical user interface) elements. It was designed to work on simple text-only terminals.

Less has been used by thousands of people around the world since its release in 1985.

Why would you want to use less?

If you're not using a pager now, then you probably don't need less. But if you're using another pager, you should take a look at less. Less has some very nice features:

- Backwards movement

It's hard to believe some people are still using pagers that don't let you move backwards in the file. Even pagers that do let you move backwards often don't work when viewing “standard input”; that is, data that is being piped into the pager from another program.

- Searching and highlighting

Less lets you search for text in the file you're viewing. You can search forwards and backwards. Furthermore, less highlights the text that matches your search, to make it easy to see where the match is.

- Multiple files

Less allows you to switch between any number of different files, remembering your position in each file. You can also do a single search which spans all the files you are working with.

- Bold and underline

Some text files, notably UNIX-style “man pages”, use overstriking (backspaces) to simulate underlined and bold text. Less interprets this overstriking and displays the text in real underlined or bold mode.

- Advanced features

Less has a large number of fancy features for advanced users. Once you learn the basics, you'll find you can customize less to do almost anything you want it to do. You can change key bindings, set different tab stops, set up filters to view compressed data or other file types, customize the prompt, display line numbers, use “tag” files, and more.

- Portability

Less is available on many different machines, different system architectures, and different terminals. Pretty much wherever you go, you'll find less. You won't have to learn how to use a different pager on every system you use.

Where did less come from?

Back in 1983, I was working for a now-defunct company called Integrated Office Systems. Some of our software ran on a UNIX system and produced enormous log files of its transactions. We wanted to be able to search these log files for error messages. But the version of “vi” we were using couldn't handle files larger than a certain size, and our log files often exceeded that size. So we were forced to use “more” to view the files. The problem was, once we found an error message, what we really wanted to see was the transactions leading up to the error; that is, the lines in the log file immediately before the error message. But more didn't allow backward movement in the file.

We often complained about this problem. We said we needed a “backwards more”; someone (it wasn't me, but unfortunately I don't remember who it was) coined the name “less” as a joke on the concept of a “backwards more”. It didn't seem to me that it would be too difficult to

write a simple pager that would allow forward and backward movement. I wrote the first version in late 1983 and we began using it internally. I continued to enhance it and finally decided it was a useful enough tool that I should make it available to the world. I posted the first version to the newsgroup called (at that time) net.sources in May, 1985.

Making the program publicly available allowed an enormous number of people to use it. Many people have made their own modifications and donated them to me for incorporation into the official version, and many more have reported bugs they've found or made suggestions about features they'd like to see added. This was my first experience with the concept of what is now called "open source" or "free software", and it has, I believe, produced a very high quality product. I'd like to thank all the users of less for their invaluable comments and suggestions over the years. Less wouldn't be what it is today without you.

There were no OS/390 changes, so less is full in the open source stream.

26.2 Installation of less

The less package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 358.

Installing less will store files to the directories /usr/local/bin and /usr/local/man.

If you select to recompile less, do the following. The text in the package is in ASCII, so it can be unwound with the `pax` command and the `-o` flag:

```
$ export A2E="-o from=ISO8859-1,to=IBM-1047"
$ ls -l less-358.tar.gz
-rw-r--r-- 1 SANDORB SYS1 28268 Mar  8 18:36 less-358.tar.gz
$ gzip -c -d less-358.tar.gz | pax $A2E -r
$ cd less-358
$ CC=cc CFLAGS="-O2 -D_ALL_SOURCE" CCC=c++ CCFLAGS="-+" ./configure
$ make
$ su
$ make install
```

26.3 Customizing and operation of less

Most people will start less without any customization, or call less without any options, with the filename as a parameter (it can contain wildcards). However, there are several options available, which you can see in the help menu.

You can use the option `-?` if you want help. Or, if you are already within `less`, you can type `h` (or `H`). You will be brought into the online manual. Once there, check the chapter on options; with `less`, you can move in a file, search for some pattern; jump to other parts of the file, and so on.

If you are an expert user and you want to customize `less`, consult the man page of `lesskey`. There you can learn how to: define commands bound to some keys; redefine the default commands; enable or disable keys, and so on. Such changes may be defined locally, in your `.lessrc` file, or globally by the system administrator.

If you notice problems with the charset, consult the man page, where you will see the following:

```
In special cases, it may be desired to tailor less to use a character
set other than the ones definable by LESS-CHARSET. In this case,
the environment variable LESS-CHARDEF can be used to define a character
set. It should be set to a string where each character in the string
represents one character in the character set. The character "." is
used for a normal character, "c" for control, and "b" for binary. A
decimal number may be used for repetition. For example, "bccc4b."
would mean character 0 is binary, 1, 2 and 3 are control, 4, 5, 6 and 7
are binary, and 8 is normal. All characters after the last are taken to
be the same as the last, so characters 9 through 255 would be normal.
(This is an example, and does not necessarily represent any real
character set.) This table shows the value of LESSCHARDEF which is
equivalent to each of the possible values for LESSCHARSET:
```

ascii	8bcccbcc18b95.b
dos	8bcccbcc12bc5b95.b.
ebcdic	5bc6bcc7bcc41b.9b7.9b5.b..8b6.10b6.b9.7b9.8b8.17b3.3b9.7b9.8b8.6b10.b.b.b.
iso8859	8bcccbcc18b95.33b.

26.4 Licence for less

`Less` is "free software". It includes the GPL, and also includes the following license:

```
Copyright (C) 1984-2000 Mark Nudelman
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice in the documentation and/or other materials provided with the
distribution.
```

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE (END)

26.5 Getting less from the Internet

The package was found on the Web at:

<http://www.greenwoodsoftware.com/less/>

You can get it also from the GNU site at:

<ftp://www.gnu.org/pub/gnu/less/>

26.6 Porting less to OS/390

You can port less to OS/390 as described in 26.2, "Installation of less" on page 133. It is an easy port, thanks to the several previous ports, and especially to Thomas Dorner, who added EBCDIC support to the source.

During our work on this Redbook we received a patch for less from Paul Gilmartin. It adds support for:

- audible alert
- caret-escaped control characters
- the top half of IBM-1047.

This patch is integrated into the source and binary files on the CD.

Chapter 27. Libpng library

This chapter discusses the libpng library open source software package.

27.1 Introduction to the libpng library

The Portable Network Graphics (PNG) format was designed to replace the older and simpler GIF format and to some extent, the much more complex TIFF format.

For the Web, PNG really has three main advantages over GIF: alpha channels (variable transparency), gamma correction (cross-platform control of image brightness), and two-dimensional interlacing (a method of progressive display). PNG also compresses better than GIF in almost every case, but the difference is generally only around 5% to 25%, not a large enough factor to encourage folks to switch on that basis alone.

One GIF feature that PNG does not try to reproduce is multiple-image support, especially animations; PNG was (and is) intended to be a single-image format only. (A very PNG-like extension format called MNG was finalized in mid-1999 and is beginning to be supported by various applications, but MNGs and PNGs will have different file extensions and different purposes.)

For image editing, either professional or otherwise, PNG provides a useful format for the storage of intermediate stages of editing. Since PNG's compression is fully lossless—and since it supports up to 48-bit truecolor or 16-bit grayscale—then saving, restoring and re-saving an image will not degrade its quality, unlike standard JPEG (even at its highest quality settings).

And unlike TIFF, the PNG specification leaves no room for implementors to pick and choose what features they'll support; the result is that a PNG image saved in one application is readable in any other PNG-supporting application. (Note that for transmission of finished truecolor images—especially photographic ones—JPEG is almost always a better choice. Although JPEG's lossy compression can introduce visible artifacts, these can be minimized, and the savings in file size even at high quality levels is much better than is generally possible with a lossless format like PNG.)

The complete description can be obtained on the Internet at:

<http://www.libpng.org/pub/png/>

This library is required by other graphic packages, such as Webalizer and GD.

27.2 Installation of the libpng library

The tool gunzip is needed to install the libpng library. The zlib must already have been installed in your system (typically in /usr/local).

This package does not have a configure script to generate the Makefile, so the destination directory is fixed to /usr/local. If you want install it in another location, you must do a manual installation.

Here is the core of the install script named libpng.install, which is included on the CD that accompanies the hardcopy of this redbook.

```
gzip -c -d libpng-1.2.1.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s libpng-1.0.10 libpng
cd libpng
chmod u+w *
export _C89_CCMODE=1
make
make install
```

This script generates the library and installs it to /usr/local/lib. The header files are put in /usr/local/include.

27.3 Usage of the libpng library

Examples illustrating the use of this package are the GD library and Webalizer. See Chapter 15, “Gd” on page 81 and Chapter 47, “Webalizer” on page 261. Also refer to the code.

27.4 License

The following license was found on the libpng Web site:

libpng versions 1.0.7, July 1, 2000, through 1.0.10, March 30, 2001, are Copyright (c) 2000, 2001 Glenn Randers-Pehrson and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux
Eric S. Raymond
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger. Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc. For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

27.5 Getting the libpng library from the Internet

The libpng library packages are available on the CD that accompanies the hardcopy of this redbook. The version of the libpng package on the CD is 1.0.10.

You can download gd from the Internet starting at:

<http://www.libpng.org/pub/png/>

We found the package starting at:

<http://www.photodex.com/png/libpng.html>

The package is in ASCII, .gz format, and was unwound using the following ext command because it contains some binary image files:

```
$ gunzip libpng-1.0.10.tar.gz
$ ext -v libpng-1.0.10.tar
```

27.6 Porting the libpng library to OS/390

The libpng library built cleanly on OS/390 UNIX. The only changes we had to make were to the Makefile. We set the compiler to C89 and changed the location of the zlib library and include files:

```
CC=C89

# Where the zlib library and include files are located
ZLIBLIB=/usr/local/lib
ZLIBINC=/usr/local/include
#ZLIBLIB=../zlib
#ZLIBINC=../zlib
.....

RANLIB=echo
#RANLIB=ranlib
```

Then we set the environment variable `_C89_CCMODE` to 1 to allow liberal ordering of parameters and built with `gmake`:

```
$ export _C89_CCMODE=1
$ gmake
```

Chapter 28. Libtool

This chapter discusses the libtool software package.

28.1 Introduction to libtool

The copy of libtool distributed with this redbook was written from scratch by Jeff Trawick. It was written to be used with the installation of Apache and probably does not replace the entire libtool suite.

The following description was taken from the GNU Web site. See this site for more details:

<http://www.gnu.org/software/libtool/manual.html>

In the past, if a source code package developer wanted to take advantage of the power of shared libraries, he needed to write custom support code for each platform on which his package ran. He also had to design a configuration interface so that the package installer could choose what sort of libraries were built.

GNU Libtool simplifies the developer's job by encapsulating both the platform-specific dependencies, and the user interface, in a single script. GNU Libtool is designed so that the complete functionality of each host type is available via a generic interface, but nasty quirks are hidden from the programmer.

GNU Libtool's consistent interface is reassuring... users don't need to read obscure documentation in order to have their favorite source package build shared libraries. They just run your package configure script (or equivalent), and libtool does all the dirty work.

There are several examples throughout this document. All assume the same environment: we want to build a library, libhello, in a generic way.

libhello could be a shared library, a static library, or both... whatever is available on the host system, as long as libtool has been ported to it.

28.2 Installation of libtool

Here is the core of the install script named libtool.install, which is included on the CD that accompanies the hardcopy of this redbook (where \$installDir is typically /usr/local):

```
gzip -c -d libtool-os390.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s libtool-os390 libtool
```

```
cd libtool
export LIBTOOL_PREFIX=$installDir
echo "building via the command 'make install'"
make install
```

28.3 License

The libtool package is distributed under the Apache License. Details are available in the LICENSE file as part of the software distribution.

In general it is free for everybody to use, but the distribution and usage of the name of the server is still attached to a number of conditions and approvals set by the Apache Group.

28.4 Getting libtool from the Internet

You can get libtool from Jeff Trawick's CVS root:

```
anoncvs@cs390-1.raleigh.ibm.com:/cvsroot
```

or from the Tools and Toys page:

<http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxaltoy.html>

Chapter 29. Lynx

This chapter discusses the Lynx software package. Lynx is a text-only browser program. It provides users with the functionality to access WWW computers, transfer files (ftp), use gopher menus, connect to other computers (telnet), and connect to Internet search programs such as archie, veronica, and so on.

OS/390 does not have the functionality of a browser. Thus, providing a package that allows one to browse the Internet is always a useful resource. The advantage of Lynx is that it is a text-only browser, making it faster than other graphical browsers which display image files. Thus, Lynx is a good point to access text information on the World Wide Web.

29.1 Introduction to Lynx

Lynx is a World Wide Web browser for users on UNIX and VMS platforms. It is used with cursor-addressable, character terminals or emulators, including VT100 terminals and desktop-based software packages emulating VT100 terminals, such as Kermit, Procomm, and so forth. Lynx was originally conceived and developed by Academic Computing Services at the University of Kansas.

29.2 Installation of Lynx

The Lynx package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.8.3.

Installation of Lynx is quite simple. To build the makefile, we ran configure. In order to build it, we ran make. Lynx installed successfully without any modifications.

The core of the installation script is as follows:

```
gzip -c -d lynx2.8.3.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s lynx2-8-3 lynx
cd lynx
./configure --prefix=$installDir
make
make install
make install-doc
make install-help
```

29.3 Customization and operation of Lynx

We tested the operation of Lynx by opening the help file, as follows:

```
./lynx -cfg=../lynx2-8-3/lynx.cfg ../lynx2-8-3/lynx_help/lynx_help_main.html
```

We also tested Lynx on the locally running Apache server 2.0 by using the following command:

```
./lynx -cfg=../lynx2-8-3/lynx.cfg wtsc55oe
```

We set the proxy option in the lynx.cfg file to a internal squid server running on wtsc55oe on port 3128, and successfully opened HTML pages.

Since socks is not ported to OS/390, configuration with --with-socks or --with-socks5 does not work.

29.4 License

The terms of use and redistribution of Lynx are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

29.5 Getting Lynx from the Internet

This package was found on the Web at:

<http://www.lynx.browser.org/>

29.6 Porting Lynx to OS/390

Lynx built cleanly without any modifications.

Chapter 30. M4

This chapter discusses the m4 open source software package.

30.1 Introduction to m4

M4 is a C preprocessor. This used to be a stand-alone step before compilation. Today, preprocessing is built into C/C++ compilers. M4 is still required to build some packages (for example, autoconf and automake).

The OS/390 changes were not placed back in the open source stream.

30.2 Installation of m4

Here is the core of the install script named m4.install, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
pax -o from=ISO8859-1,to=IBM-1047 -rzf m4-1.4.os390.tar.Z
ln -s m4-1.4 m4
cd m4
LDFLAGS=-Wl,EDIT=NO ./configure --prefix=$installDir
cd lib
make
cd ../src
make
cp m4 $installDir/bin/m4
cp m4.1 $installDir/man/$LANG/cat1
```

30.3 Customization and operation of m4

An on-line m4 manual was found on the Web at:

<http://www.gnu.org/manual/m4/index.html>

30.4 License

The terms of use and redistribution of m4 are governed by the GPL. See Appendix A, "The GNU General Public License" on page 289 for the complete text.

In general, you are free to copy, modify and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

30.5 Getting m4 from the Internet

The m4 package on the CD is version 1.4. The package was found on the Internet at:

<ftp://ftp.mks.com/pub/s390/gnu/>

The source code package was downloaded in binary and FTPed to OS/390. It is in ASCII, so it can be unwound with the following `pax` command (where the environment variable `A2E` is set to '-o from=ISO8859-1,to=IBM-1047'):

```
$ ls -l
total 1024
-rw-r----- 1 MIKEM   SYS1      494407 Feb 29 14:05 m4-1_4_os390_tar.Z
$ pax -rzf *.Z $A2E
$ ls -l
total 1056
drwxr-xr-x  7 MIKEM   SYS1      8192 Feb 29 14:05 m4-1.4/
-rw-r----- 1 MIKEM   SYS1      494407 Feb 29 14:05 m4-1_4_os390_tar.Z
$ ln -s m4-1.4 m4
```

30.6 Porting m4 to OS/390

The porting of m4 to OS/390 was done by MKS; no notes were found regarding the details of the port.

To build the executables, the usual steps of configure, then make, were done:

```
$ export CFLAGS=-D_ALL_SOURCE
$ ./configure
creating cache ./config.cache
checking host system type... s390-ibm-os390
...
creating examples/Makefile
creating config.h
```

Initially, the following error was encountered:

```
$ make
/bin/c++ -c -DHAVE_CONFIG_H -I. -I. -I../lib -O2 -W c,dll \
-D_ALL_SOURCE m4.c
ERROR CBC3247 /usr/include/signal.h:435  Incompatible type specifier
"struct".
```

```
ERROR CBC3137 /usr/include/signal.h:431 Declaration must declare at
least one declarator, tag, or the members of an enumeration.
ERROR CBC3007 /usr/include/signal.h:491 "struct sigaltstack" is
undefined.
CBC1793(I) Compilation failed for file ./m4.c. Object file not created.
```

What apparently is happening is that the `stack_t` structure is already being defined in `config.h` and this causes the error in the declaration. The configure script, which creates `config.h`, seems to think that `stack_t` was not defined in `signal.h` and therefore provides its own. Using the `cc` compiler causes the configure script not to use `stack_t` and therefore the file compiles cleanly.

Chapter 31. MySQL

This chapter discusses the MySQL open source relational database package. Unfortunately, due to legal concerns, this package cannot be distributed with the redbook nor on the Tools and Toys Web site.

31.1 Introduction to MySQL

MySQL is a very widely used relational database management system (RDBMS) in the open source community. It implements most of the SQL-92 standard. It is very popular in dynamic Web sites, and interfaces to all major programming languages are available. MySQL is free software and the source code availability makes it highly customizable.

31.2 Installation of MySQL

MySQL requires Perl for its installation.

An install script named `mysql.install` is included on the CD. A manual installation is also possible:

After FTPing the ported source .tar file from the CD to the S/390 host, unwind it with the `pax` command:

```
$ pax -rf mysql-3.23.39-390.tar
```

Change to the newly created directory, then set the C++ compiler options via the `CXXFLAGS` environment variable:

```
$ export CXXFLAGS="-+ -W0,langlvl\ (extended\ ) -D_OPEN_THREADS \
-D_XOPEN_SOURCE_EXTENDED"
```

Set the location of the linker to use to a dummy value:

```
$ export LD=echo
```

Do the configuration with specification of the codepages to use:

```
$ ./configure --with-charset=ibm1047 --with-extra-charsets=latin1
```

Call make to compile and install it:

```
$ make
$ su
# make install
```

31.3 Customization and operation of MySQL

Before using MySQL, the server's grant tables have to be installed with the following command:

```
mysql_install_db
```

This creates default user accounts for the RDMS.

The MySQL server `mysqld` is started as a UNIX daemon that only accepts commands from a TCP connection. By default, it listens on port 3306, and that is also the default for all the MySQL clients. The MySQL TCP port does not need to be included into the `/etc/services` file, as `inetd` is not used with MySQL.

To start the server and put it into the background, the following command is invoked:

```
$ safe_mysqld &
```

`safe_mysqld` is a shell script that starts the MySQL server, `mysqld`, and makes sure it is automatically restarted when it gets killed to guarantee high availability.

The server now starts up and waits for connections. To use it, you need a client like the command line MySQL client, or you can contact the server directly from your program code with an interface.

MySQL uses its own independent user management. User data is stored in a database table, and MySQL accounts are not related in any way to the user management of the server machine.

For security purposes, first install a password for the MySQL root user. This is done with the `mysqladmin` command:

```
mysqladmin -u root -h <hostname> password <new password>
```

A MySQL ODBC driver is available, for Microsoft Windows and some UNIX systems, called MyODBC that makes nearly every database-enabled application running under these operating systems able to talk to a MySQL server. You can get this driver at the following site:

<http://www.mysql.com/downloads/api-mysdbc.html>

MySQL is especially suited for Web applications. It is natively supported by PHP, and connecting to it from Perl applications requires only the standard DBI/DBD:mysql modules.

For more information on the use of the MySQL server, see the accompanying documentation or visit the MySQL documentation site:

<http://www.mysql.com/documentation/index.html>

The MySQL documentation is available as one large HTML page at:

<http://www.mysql.com/documentation/mysql/full/>

31.4 License

The terms of use and redistribution of MySQL are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

31.5 Getting MySQL from the Internet

The ported MySQL package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.23.39 dated June 2001. The original package was found on the Internet at:

<ftp://ftp.mysql.com/MySQL-3.23/mysql-3.23.39.tar.gz>

For information about newer versions, visit the MySQL Web site:

<http://www.mysql.com>

31.6 Porting MySQL to OS/390

This section discusses the state of the port, as well as problems we encountered during porting with their solutions.

31.6.1 State of the port

The MySQL RDMS was ported especially for this redbook.

Porting MySQL to OS/390 UNIX took considerable effort, as it is quite a large program, consisting of more than 1000 C and C++ source files. For that reason, there was not enough time during this project to thoroughly test all features of the package, so there may be functionality that is not working correctly as it was not taken into consideration during the port.

31.6.2 Problems faced

Following is a list of the problems we faced during porting, along with solutions, in the order of their occurrence:

31.6.2.1 During configure stage

Problem:

The configure utility does not correctly recognize the type of machine.

Solution:

Use the `config.guess` and `config.sub` files from the ported `autoconf` package (see 6.3, “Customization and operation of `autoconf`” on page 44).

Problem:

configure: no acceptable `ld` found

Solution:

no linker needed, so `ld` is set to `echo` by issuing the command `export LD=echo`

Problem:

configure complains that the compiler is not ANSI C compatible.

Solution:

It is, but it does not understand the option that makes other compilers ANSI C compatible. So the test for it (`AM_PROG_CC_STDC`) can be removed from the MySQL `configure.in` and the configure script can be newly generated with `autoconf`.

31.6.2.2 During compilation

Problem:

Duplicate definition of `ushort` found.

Solution:

The type `ushort` is already defined by OS/390, so make the `#define` in MySQL's `global.h` conditional:

```
#if !defined(__ushort)
#define __ushort
typedef unsigned short ushort;
#endif
```

Problem:

During compilation of `violite.c`, the header file `poll.h`, that defines the symbol `POLLIN`, is not found.

Solution:

`poll.h` is located in `usr/include/`, so change the include statement path in `violite.c` to that directory.

Problem:

During compilation of `my_init.c`, the structure `rusage` does not have the format MySQL would expect.

Solution:

The structure shall not be used under OS/390 UNIX. MySQL can handle this, just add `#undef HAVE_GETRUSAGE` to `config.h` or `my_init.c`.

Problem:

The C++ compiler does not accept source files with a `.cc` suffix.

Solution:

Adding the `-+` option, it does. (`export CXXFLAGS='-+'`)

Problem:

The C++ compiler does not recognize the `long long` data type.

Solution:

This is an extension to the standard C library features. To use it, you must set a compiler option (with the point mentioned before in mind: `export CXXFLAGS='-+ -W0,langlvl\ (extended\)'`)

Problem:

Threads won't compile.

Solution:

Tell the compiler you want to use threads. (`export CXXFLAGS='-+ -W0,langlvl\ (extended\) -D_OPEN_THREADS'`)

Problem:

Function `localtime_r` not found.

Solution:

See 31.6.3, "Necessary code changes" on page 158.

Problem:

Compiler does not find data type `sa_family_t`.

Solution:

Add `#include <sys/types.h>` to the file `sql/mysql_priv.h` which is included in all the files that need `sys/types.h`.

Problem:

Compiler does not find data type `mcontext_t`.

Solution:

Define it yourself: Add `typedef int mcontext_t[40];` to file `sql/mysql_priv.h` which is included in all files that need this definition.

Problem:

Identifier `EINPROGRESS` not found

Solution:

Define it in the source file: `#define EINPROGRESS 1103`

Problem:

Type `SOCKOPT__OPTLEN_TYPE` not found.

Solution:

Replace it by the `unsigned int` type.

Problem:

Function `gethostbyname_r()` not found.

Solution:

On OS/390, the thread-safe `gethostbyname_r()` function is not needed, as the `gethostbyname()` function itself is already thread-safe. The most convenient way to make the compiler use it is by adding a macro to the source file:

```
#define gethostbyname_r(name,a,b,c,d) gethostbyname(name)
```

Problem:

Type `ushort` not found.

Solution:

Replace by what stands for: `unsigned short`.

Problem:

Header files `netinet/in_system.h`, `netinet/ip.h`, `netinet/tcp.h` not found.

Solution:

These header files are not really needed, so the `#include` statements can safely be removed.

Problem:

Header file `sys/poll.h` not found.

Solution:

This header file is actually located in `/usr/include`, so the relative path `sys/` needs to be removed from the `#include` statement.

Problem:

`bzero` function not found.

Solution:

This function can be emulated with the `__memset()` function, add the following macro to the source file:

```
#define bzero(x,y) __memset(x,0,y)
```

Problem:

Identifier `finite` is undefined.

Solution:

Add `#define finite __fp_finite` to the source file.

Problem:

Type `id_t` not found.

Solution:

Define it by adding `typedef int id_t;` to the source file.

Problem:

The compiler says it can't convert a function to `extern "C"`

Solution:

This occurs often; functions that are pointed to by a function pointer need to be declared `extern "C"` for the OS/390 compiler. This declaration can be easily added to a function by a change such as this:

```
int myfunction(int a,int b);  
extern "C" int myfunction(int a, int b);
```

Sometimes it is necessary to remove a declaration of `static` from a function, as it cannot be both `static` and `extern "C"`. Removing the static declaration does not affect the behavior of the function other than making the function accessible from outside the source file it is implemented in.

In one case the problem was a little harder to solve: the `get_ptr_compare` function returns a function pointer, and declaring it as `extern "C"` would not work. The function selects a comparison function to which it returns a pointer. So we put the functionality of this function into the calling function and wrote a wrapper function that simply calls the already selected comparison function. The wrapper function can now be declared `extern "C"`:

```
extern "C" int ptr_compare_w(void *p1, byte *p2, byte *p3) { return  
    fptr_cmp_temp(p1, (void*) p2, (void*) p3); }
```

Problem:

During linking of the client tools, numerous symbols cannot be resolved.

Solution:

The `libmysqlclient.a` library does not seem to include all the functions required to link the client tools. The make process was altered, so that the library now includes all the object files in the `libmysql` directory, by adding the following line to the `libmysql` Makefile:

```
ar cru .libs/libmysql.a *.o *.lo
```

31.6.3 Necessary code changes

Apart from changing the code to make it compatible with the OS/390 C compiler, we also needed to make some changes to how the MySQL code works, mainly in regard to ASCII/EBCDIC-related issues.

31.6.3.1 gen_lex_hash.cc / lex_hash.h

The gen_lex_hash program creates a file called lex_hash.h. This file contains a function called get_hash_symbol() that takes a string and returns the keyword symbol that it is represented by. This is done during the installation of MySQL.

Gen_lex_hash creates a confusingly big lookup table for the purpose of finding keyword symbols. For some reason, it did not find a suitable table on OS/390, even after some ASCII dependencies were removed.

Therefore, we disabled gen_lex_hash during the installation and wrote a substitution for the generated lex_hash.h. The function most likely does not achieve the same performance as the native MySQL code, because it uses character comparison instead of table lookups.

Following is the source code of the substitute get_hash_symbol function:

```
/* 390: hash replacement written by Thorsten Brockmeier */
#include "lex.h"
inline SYMBOL *get_hash_symbol(const char *s,unsigned int length,bool
function) {
    int i,j, next_try;
    char* temp_s;
    SYMBOL* sim;
    sim=symbols; /* start at first symbol */
    for (i=0;i<=309;i++ ) {
        temp_s=(char*)s;
        next_try=0;
        for (j=0;j<length;j++ ) {
            if (!(to_upper_lex[*temp_s] ==to_upper_lex[sim->name[j]])){
                next_try=1;
                break;
            }
            temp_s++;
        }
        if (!next_try && length==strlen(sim->name))
            return sim;
        sim++; /* advance to next symbol */
    }
    if (function) {
        sim=sql_functions; /* start at first function */
```

```

    for (i=0;i<124 ;i++ ) {
        temp_s=(char*)s;
        next_try=0;
        for (j=0;j<length ;j++ ) {
            if (!(to_upper_lex[*temp_s] ==to_upper_lex[sim->name[j]])){
                next_try=1;
                break;
            }
            temp_s++;
        }
        if (!next_try && length==strlen(sim->name)) {
            return sim;
        }
        sim++; /* advance to next symbol */
    }
}
return (SYMBOL *)0;
}

```

This function can also be found in the file `sql/lex_hash.h` in the MySQL archive on the CD.

31.6.3.2 EBCDIC character set

MySQL supports different character sets, which are used by the SQL command scanner to find out which characters are letters of the alphabet, digits, or spaces. It was relatively easy to add a character set for EBCDIC (IBM-1047); the MySQL documentation contains a description of how to do so.

Creating EBCDIC support in the source code involves more effort. Following is a complete list of necessary changes:

1. Add `ibm1047` to `sql/share/charsets/Index` with an unused number like 31.
2. Create `sql/charsets/ibm1047.conf`, the file that contains the character definition arrays. It needs to contain the following:

```
# Configuration file for the ibm1047 ebcdic character set
```

```
# ctype array (must have 257 elements)
```

```

00
20 20 20 20 20 20 20 20 20 20 20 20 28 28 20 20 #0
20 20 20 20 20 28 20 20 20 20 20 20 20 20 20 20 #16
20 20 20 20 20 28 20 20 48 20 20 20 20 20 20 20 #32
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 #48
20 20 20 20 20 20 20 20 20 20 10 10 10 10 10 10 #64
10 20 20 20 20 20 20 20 20 20 20 10 10 10 10 10 #80

```

```

10 20 20 20 20 20 20 20 20 20 20 10 10 10 10 10 #96
20 20 20 20 20 20 20 20 20 20 20 10 10 10 10 10 #112
20 82 82 82 82 82 82 02 02 02 20 20 20 20 20 20 #128
20 02 02 02 02 02 02 02 02 02 20 20 20 20 20 20 #144
20 20 02 02 02 02 02 02 02 02 20 20 20 20 20 20 #160
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 #176
20 81 81 81 81 81 81 01 01 01 20 20 20 20 20 20 #192
20 01 01 01 01 01 01 01 01 01 20 20 20 20 20 20 #208
20 20 01 01 01 01 01 01 01 01 20 20 20 20 20 20 #224
84 84 84 84 84 84 84 84 84 84 20 20 20 20 20 20 #240

```

to_lower array (must have 256 elements)

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
E0 81 82 83 84 85 86 87 88 89 EA EB EC ED EE EF
F0 91 92 93 94 95 96 97 98 99 FA FB FC FD FE DF
E0 E1 A2 A3 A4 A5 A6 A7 A8 A9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

```

to_upper array (must have 256 elements)

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 7B 7C 7D 7E 7F
80 C1 C2 C3 C4 C5 C6 C7 C8 C9 8A 8B 8C 8D 8E 8F
90 D1 D2 D3 D4 D5 D6 D7 D8 99 9A 9B 9C 9D 9E 9F
A0 A1 E2 E3 E4 E5 E6 E7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

```



```
# sort_order array (must have 256 elements)
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 7B 7C 7D 7E 7F
80 C1 C2 C3 C4 C5 C6 C7 C8 C9 8A 8B 8C 8D 8E 8F
90 D1 D2 D3 D4 D5 D6 D7 D8 99 9A 9B 9C 9D 9E 9F
A0 A1 E2 E3 E4 E5 E6 E7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
```

3. Add the character set to the list of known charsets (`CHARSETS_AVAILABLE`) in `configure.in` like this:

```
CHARSETS_AVAILABLE="big5 cp1251 cp1257 croat czech danish dec8 dos estonia
euc_kr gb2312 gbk german1 greek hebrew hp8 hungarian koi8_ru koi8_ukr
latin1 latin2 latin5 swe7 usa7 win1250 win1251 win1251ukr ujis sjis tis620
ibm1047"
```

4. If the server uses the IBM-1047 character set, it will also want the client to use it. As the client doesn't normally have this character set, and we don't want a warning issued each time we connect to the server, the server is made to disguise its real character set by a small change in `sql_parse.cc`:

```
end[2]=MY_CHARSET_CURRENT;
```

is replaced by

```
end[2]=8;
```

Eight is the number of the standard character set, `latin1`.

5. The `to_upper` array also needs to be included in `sql_lex.cc` in a slightly different format, replacing the array used for ASCII characters:

```
uchar to_upper_lex[] = {
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
    32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
    48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
    64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
    80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
    96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
    112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,
```

```

128,193,194,195,196,197,198,199,200,201,138,139,140,141,142,143,
144,209,210,211,212,213,214,215,216,217,154,155,156,157,158,159,
160,161,226,227,228,229,230,231,232,233,170,171,172,173,174,175,
176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,
192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,
208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,
224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,
240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255
};

```

6. MySQL now has to be reconfigured with `./configure --with-charset=ibm1047 --with-extra-charsets=latin1`, recompiled and reinstalled.

31.6.3.3 ASCII/EBCDIC translation

The majority of code changes necessary to make MySQL function properly on OS/390 were translations from ASCII to EBCDIC and the other way round. To enable the OS/390 MySQL server able to talk to clients on other platforms, and the OS/390 client to talk to other servers, all the data sent across the network is coded as ASCII characters.

The processing inside the program like parsing statements and reading results from tables is all done in EBCDIC, so a translation has to be done on all character information that is sent/received. However, not everything may be encoded. Binary data needs to remain unchanged.

There are many places in the code where such translation was necessary, but we do not document them here; refer to the source code on the CD.

The following instructions from the OS/390 C run-time library were used for translation:

Function prototype	Description
<code>int __atoc(char*)</code>	Converts a string from ASCII to EBCDIC
<code>int __etoc(char*)</code>	Converts a string from EBCDIC to ASCII
<code>int __atoc_l(char*,int)</code>	Converts the specified number of characters from ASCII to EBCDIC
<code>int __etoc(char*,int)</code>	Converts the specified number of characters from EBCDIC to ASCII

Chapter 32. Nedit

This chapter discusses the nedit open source software package.

32.1 Introduction to nedit

One thing experienced OS/390 users will notice in a UNIX telnet session is the lack of a full-screen editor. Although most *UNIX developers* are happy using vi, vim, or emacs, these tools require a certain amount of time and effort to get used to. So for OS/390 users, importing the nedit package offers a quick and easy solution to the “editor problem”.

The nedit editor is a standard GUI-type editor for programs and normal text files. It has been written to make users of Macintosh and Windows feel as much at home as possible. It provides all the components of a full-screen editor, such as standard buttons, menus and mouse support. Being a 100% GUI tool, it's easy to learn and is the perfect choice not only for beginners but also for professional users.

32.2 Installing nedit

The nedit package on the CD is the latest stable release V5.1.1.

Preparing to install

As part of the normal process to install an open software package, read the REAME file in the installation directory and check for the latest information about the installation and customization process.

Check for yyparse.c

Before proceeding, check that the yyparse.c file is in the /etc directory. If it is not, copy it from the /samples directory in order to ensure that the nedit package will compile.

If you want to use the software as it is distributed on the CD, you can run the install.sh script. This will do everything for you and install the software into your /usr/local environment.

To install nedit into another location, or to install a later release from the Web, you have to execute the individual steps on your own.

Installation process

1. Load the software.

Because the nedit package is a gzip file containing ASCII source code, you'll need the GNU zip utility gunzip before you install nedit (see Chapter 20, "Gzip" on page 105").

Now use FTP to transfer the nedit file (nedit-5_1_1-src_tar.tar.gz) from the CD in binary mode to your OS/390 file system:

```
$ gunzip nedit-5_1_1-src_tar.tar.gz
$ ls
nedit-5_1_1-src_tar.tar
$ export $A2E="-o from=ISO8859-1,to=IBM-1047"
$ pax -rf nedit-5_1_1-src_tar.tar $A2E
$ ls
nedit-5.1.1          nedit-5_1_1-src_tar.tar
```

2. Apply patches for OS/390.

These changes are necessary to install the source code from the nedit mainline on OS/390. The patches on the CD are written *exactly for this release*; if you are going to install a newer release, the patches may fail. Following is the process that should be used to patch the code.

```
$ cd nedit-5.1.1
$ cp /u/egont/Makefile.os390 makefiles
$ cp /u/egont/os390.diffs ./
$ patch -bc -i os390.diffs -p 0
patching file ./source/file.c
patching file ./source/help.c
patching file ./source/highlight.c
patching file ./source/highlightData.c
patching file ./source/interpret.c
patching file ./source/macro.c
patching file ./source/menu.c
patching file ./source/nc.c
patching file ./source/nedit.c
patching file ./source/parse.y
patching file ./source/preferences.c
patching file ./source/search.c
patching file ./source/selection.c
patching file ./source/server.c
patching file ./source/shell.c
patching file ./source/shift.c
patching file ./source/smartIndent.c
```

```
patching file ./source/tags.c
patching file ./source/textBuf.c
Hunk #2 succeeded at 861 with fuzz 1.
patching file ./source/undo.c
patching file ./source/userCmds.c
patching file ./source/window.c
patching file ./util/fileUtils.c
patching file ./util/getfiles.c
patching file ./util/prefFile.c
patching file ./util/printUtils.c
```

3. Build nedit.

Continue with this command:

```
$ make os390
```

This will build the product in the ./source directory.

4. Copy nedit into the runtime directory.

You can now either copy the nedit program into your /usr/local/bin directory or create a symbolic link to run it from the installation environment. You can also choose any other directory which is in the normal program path for your installation.

Use either:

```
$ cp source/nedit /usr/local/bin
```

or

```
$ cd /usr/local/bin
$ ln -s /usr/local/nedit/nedit-5.1.1/source/nedit nedit
```

Afterwards, you can use nedit from your normal PATH.

32.3 Customizing and operating nedit

Keep in mind that nedit is an X/Window application and that you need an active X/Window Server on your client. In addition, you have to tell your telnet session (and nedit) how to find your PC. The environment variable DISPLAY will be checked for this information.

For more information about setting this environment variable see 50.1.4, “Setting the DISPLAY variable for X Window applications” on page 273. In 50.1.6, “Downloading an X Window server from the Internet” on page 274 you can find the URL to get a trial version of a server to your workstation.

Full documentation of this package is available on the nedit Web site. Refer to the following URL for direct HTML browser access, or to download the documentation to your workstation:

<http://www.nedit.org/documentation/doc.shtml>

32.4 The license for nedit

The nedit package is covered by the GNU General Public License which is explained in detail in 2.1, “The GNU General Public License (GPL)” on page 13.

32.5 Getting nedit from the Internet

We found the nedit package on the Web at:

<http://www.nedit.org/download/current.shtml>

After you download the package from the Web to your PC, you have to transfer it in *binary* mode to the OS/390 HFS file system.

Note: Remember that some Web browsers *omit the file suffixes* for known file types (such as .gz for gzip). Since the GNU unzip utility requires .gz as part of the filename, you’ll have to add the suffix again if it is removed by the browser.

32.6 Porting nedit to OS/390

The modifications needed to make this version of nedit work on OS/390 were written by Phil Sidler of Airborne Express and Ralf Schandl, one of the authors of this redbook.

The changes have been passed on to develop@nedit.org to consider for implementation into the mainstream of the product. Having other users also pass on their experiences and problems with nedit to this site may help to highlight the need for this tool on the OS/390 platform.

Chapter 33. OpenSSL and OpenSSH

This chapter discusses the OpenSSL and OpenSSH open source software packages.

33.1 Introduction to OpenSSL and OpenSSH

OpenSSH is a free implementation of the popular Secure Shell tools (SSH). It is based on the last free version of SSH and supports protocol versions 1.3, 1.5, and 2.0. Most of the function is available; however, neither the version 2.0 protocol nor the sftp utility are working in the OS/390 port.

The ssh protocol encrypts all data transported between two peers. A public/private key RSA authentication can be used to identify the user, but normal password authentication is also possible. Of course, the password is encrypted before it is transmitted.

OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them. The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for the following tasks:

- Creation of RSA, DH and DSA key parameters
- Creation of X.509 certificates, CSRs and CRLs
- Calculation of message digests
- Encryption and decryption with ciphers
- SSL/TLS client and server tests
- Handling of S/MIME signed or encrypted mail

33.2 Installation of OpenSSL and OpenSSH

The packages must be installed in this (required) order:

1. zlib (see Chapter 49, "Zlib library" on page 269)
2. OpenSSL
3. OpenSSH

33.2.1 Installation of OpenSSL

Here is the core of the install script openssl.install, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```

gzip -c -d openssl-0.9.6a.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s openssl-0.9.6a openssl
cd openssl
patch -p 1 <../openssl-0.9.6a-os390.patch
./Configure OS390-Unix --prefix=$installDir --openssldir=$installDir/ssl
_C89_CCMODE=1 make
make install

```

During the Configure step, some [EDC5117I File exists] messages will be issued; this is normal.

33.2.2 Installation of OpenSSH

Here is the core of the install script `openssh.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```

gzip -c -d openssh-2.9p1.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s openssh-2.9p1 openssh
cd openssh
patch -p 1 <../openssh-2.9p1-os390.patch
cp $installDir/src/autocconf/autocconf/config.sub .
cp $installDir/src/autocconf/autocconf/config.guess .
cp $installDir/src/autocconf/autocconf/install-sh .
autocconf
export _CC_CCMODE=1
export RANLIB=echo
export PATH=/usr/local/bin:$PATH
export _CC_LIBDIRS="/lib /usr/lib /usr/local/lib ./openbsd-compat ."
LDFLAGS=-Wl,EDIT=NO ./configure --prefix=$installDir
make

```

If the **make** install fails in the PRNG generation, it means that you have less than 16 valid commands in the `ssh_prng_cmds` in your installation directory `etc` subdirectory. Edit the file and rerun `make install`. Commands not valid for your system will have a `undef` sentence instead of a valid command path.

If you have `BPX.DAEMON` defined to the `RACF FACILITY` class, you *must* set the program control flag in the extended attributes of the `sshd` daemon via the `extattr +p` command, or it will be unable to succeed in password validation functions. See the chapter entitled “Observations” for an explanation of this feature.

33.3 Licenses

In this section we present the OpenSSL and Open SSH package licenses.

33.3.1 OpenSSL license

The following license was found in the file LICENSE. The majority of it is as follows:

```
LICENSE ISSUES
=====
```

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

```
OpenSSL License
-----
```

```
/* =====
 * Copyright (c) 1998-2001 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used
 * to endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
```

```

*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* =====
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
...

```

33.3.2 OpenSSH license

The license for OpenSSH is unique. It can be found in the file LICENCE:

This file is part of the OpenSSH software.

The licences which components of this software fall under are as follows. First, we will summarize and say that all components are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

```

1)
* Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
*       All rights reserved
*
* As far as I am concerned, the code I have written for this software
* can be used freely for any purpose.  Any derived versions of this
* software must be clearly marked as such, and if the derived work is
* incompatible with the protocol description in the RFC file, it must be
* called by a name other than "ssh" or "Secure Shell".

[Tatu continues]
* However, I am not implying to give any licenses to any patents or
* copyrights held by third parties, and the software includes parts that
* are not under my direct control.  As far as I know, all included
* source code is used in accordance with the relevant license agreements
* and can be used freely for any purpose (the GNU license being the most

```

* restrictive); see below for details.

[However, none of that term is relevant at this point in time. All of these restrictively licenced software components which he talks about have been removed from OpenSSH, ie.

- RSA is no longer included, found in the OpenSSL library
- IDEA is no longer included, its use is deprecated
- DES is now external, in the OpenSSL library
- GMP is no longer used, and instead we call BN code from OpenSSL
- Zlib is now external, in a library
- The make-ssh-known-hosts script is no longer included
- TSS has been removed
- MD5 is now external, in the OpenSSL library
- RC4 support has been replaced with ARC4 support from OpenSSL
- Blowfish is now external, in the OpenSSL library

[The licence continues]

Note that any information and cryptographic algorithms used in this software are publicly available on the Internet and at any major bookstore, scientific library, and patent office worldwide. More information can be found e.g. at "<http://www.cs.hut.fi/crypto>".

The legal status of this program is some combination of all these permissions and restrictions. Use only at your own responsibility. You will be responsible for any legal consequences yourself; I am not making any claims whether possessing or using this is legal or not in your country, and I am not taking any responsibility on your behalf.

2)

The 32-bit CRC implementation in crc32.c is due to Gary S. Brown. Comments in the file indicate it may be used for any purpose without restrictions:

- * COPYRIGHT (C) 1986 Gary S. Brown. You may use this program, or
- * code or tables extracted from it, as desired without restriction.

3)

The 32-bit CRC compensation attack detector in deattack.c was contributed by CORE SDI S.A. under a BSD-style license. See <http://www.core-sdi.com/english/ssh/> for details.

- * Cryptographic attack detector for ssh - source code
- *
- * Copyright (c) 1998 CORE SDI S.A., Buenos Aires, Argentina.
- *
- * All rights reserved. Redistribution and use in source and binary
- * forms, with or without modification, are permitted provided that
- * this copyright notice is retained.
- *
- *
- * Ariel Futoransky <futo@core-sdi.com>
- * <<http://www.core-sdi.com>>

4)

Remaining components of the software are provided under a standard 2-term BSD licence with the following names as copyright holders:

Markus Friedl
Theo de Raadt
Niels Provos
Dug Song
Aaron Campbell

- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.

The file also includes statements that there is no warranty with the package.

33.4 Getting OpenSSL and OpenSSH from the Internet

The OpenSSL package was found starting at:

<http://www.openssl.org/source/>

The OpenSSH package was found starting at:

<http://www.openssh.com/ftp.html>

33.5 Porting OpenSSL and OpenSSH to OS/390

A full port of OpenSSH faces the same problem as any other port to OS390 UNIX, that is, EBCDIC.

All incoming data has to be converted from ASCII to EBCDIC, and all outgoing data has to be converted from EBCDIC to ASCII. This involves an additional problem, for which we don't have a solution yet: if you use secure copy (scp), you want translation for text files, but not for binary files. The only way to solve this would be to add a new parameter, and probably take a different approach to the EBCDIC/ASCII conversion.

In this port, the conversion is done in the string reading and writing functions in the ssh/sshd programs. This works for a terminal emulation, but it isn't a good idea when ssh is used as a tunnel by other "client" applications like scp or sftp. In this case, the conversion must be made in the "client" application,

and the tunnel must remain binary-clean (no conversions). This is why scp is only useful to transfer text files, and is probably the case of the sftp failure.

The following is needed for porting OpenSSH to OS/390 UNIX:

1. OpenSSH
2. OpenSSL (OpenSSH uses the RSA code from OpenSSL)
3. libz: Compression Library
4. Entropy Gathering Daemon (EGD): An EGD is meant to be used on systems which do not have a convenient source of random bits. OpenSSH version 2, which is part of this distribution, now includes a built-in EGD.

We also looked into the possibility of using OS/390 System SSL instead of the OpenSSL library with OpenSSH. This, however, seems to be hardly possible even with extensive changes to OpenSSH, because the System SSL library does not provide access to cryptographic functions and algorithms directly, as OpenSSL does.

33.5.1 Status

With the changes described in this chapter we were able to get OpenSSH running on OS/390 UNIX. All the OpenSSH and OpenSSL changes are in the files `openssh-2.9p1-os390.patch` and `openssl-0.9.6a-os390.patch` included in this CD.

33.5.2 OpenSSH changes

First of all, a very special thanks to Manfred Porst. Due to his explanation of his SSH-port, we were able to quickly find the places to change in the OpenSSH source.

The changes described here are based on OpenSSH 2.9p1. All the changes are in bold.

configure

You must copy the files `config.sub`, `config.guess` and `install-sh` from the `autoconf` package included on this CD.

You must add the following lines to `configure.in` around line 49:

```
# Check for some target-specific stuff
case "$host" in
  *-*-os390*)
    AFS_LIBS="-lld"
    CPPFLAGS="$CPPFLAGS -I/usr/local/include"
```

```

        LDFLAGS="$LDFLAGS -I/usr/local/lib"
        AC_DEFINE(BROKEN_GETADDRINFO)
        AC_DEFINE(DISABLE_WTMP)
        AC_DEFINE(DISABLE_UTMP)
        AC_DEFINE(DISABLE_SHADOW)
        AC_DEFINE(IP_TOS_IS_BROKEN)
        ;;
    *-*-aix*)
        AFS_LIBS="-lld"
        CPPFLAGS="$CPPFLAGS -I/usr/local/include"

```

After the modification, run the “autoconf” utility to regenerate the configure script.

auth-passwd.c

The password check on OS/390 UNIX in the function `auth_password()` is quite simple. Around line 94, add the following:

```

char *salt;

#ifdef __MVS__
    int rc;
#endif

#ifdef __hpux

```

Around line 164, add the following:

```

#endif

#ifdef __MVS__
    rc = __passwd(pw->pw_name, password, NULL);
    if(rc)
        debug("__passwd: %.100s", strerror(errno));
    return rc == 0;
#else
    pw_password = pw->pw_passwd;

```

Add the following just before the closing brace of the function:

```

    /* Authentication accepted if encrypted passwords are identical. */
    return (strcmp(encrypted_password, pw_password) == 0);
#endif /* __MVS__ */
}
#endif /* !USE_PAM && !HAVE_OSF_SIA */

```

authfile.c

Around line 125, replace the line:

```
buffer_put_string(&encrypted, comment, strlen(comment));
```

with:

```
        buffer_put_bignum(&encrypted, key->rsa->e);
#ifdef __MVS__
    /* the comment has to be stored in EBCDIC */
    {
        char *tmp = xstrdup(comment);
        __atoe(tmp);
        buffer_put_string(&encrypted, tmp, strlen(tmp));
        xfree(tmp);
    }
#else
    buffer_put_string(&encrypted, comment, strlen(comment));
#endif
    /* Allocate space for the private part of the key in the buffer.
    */
```

Around line 280 in function `key_load_public_rsa1()`, add the following lines:

```
        buffer_get_bignum(&buffer, pub->rsa->e);
        if (commentp)
        {
            *commentp = buffer_get_string(&buffer, NULL);
#ifdef __MVS__
            __etoe(commentp);
#endif
        }
        /* The encrypted private part is not parsed by this function. */

        buffer_free(&buffer);
```

bufaux.c

In the function `buffer_get_string()`, the fetched string has to be translated to EBCDIC. Around line 210, behind the line:

```
value[len] = 0;
```

add:

```
        #ifdef __MVS__
            __atoe(value);
        #endif
```

In the function `buffer_put_string()` around line 220, modify the function body to:

```
void
buffer_put_string(Buffer *buffer, const void *buf, u_int len)
{
#ifdef __MVS__
    char *cp = strdup(buf);
    __etoa(cp);
    buffer_put_int(buffer, len);
    buffer_append(buffer, cp, len);
#else
    buffer_put_int(buffer, len);
    buffer_append(buffer, buf, len);
#endif
}
```

buffer.c

In the function `buffer_dump()`, replace the line:

```
u_char *ucp = (u_char *) buffer->buf;
```

With this:

```
#ifdef __MVS__
    u_char *ucp;
    ucp = xmalloc(buffer->alloc);
    memcpy(ucp, buffer->buf, buffer->alloc);
    __atoe_l(&ucp[buffer->offset], buffer->end - buffer->offset);
#else
    u_char *ucp = (u_char *) buffer->buf;
#endif
```

defines.h

Some defines are simply not present on OS/390 UNIX, or we would have to include an entire header file for only one include. Therefore, we copied some information from other header files.

Replace the line:

```
#include <sys/param.h> /* For MAXPATHLEN and roundup() */
#include <netinet/in_sysm.h> /* For typedefs */
#include <netinet/in.h> /* For IPv6 macros */
#include <netinet/ip.h> /* For IPTOS macros */
```

With this:

```
#include <sys/types.h> /* For [u]intxx_t */
```



```

#include <sys/socket.h> /* For SHUT_XXXX */
#ifdef __MVS__
# define MAXHOSTNAMELEN 64
# define IPTOS_LOWDELAY 0x10
# define TCP_NODELAY 0x01
# define IPTOS_THROUGHPUT 0x08
# define MAXPATHLEN 1024
# define IN_LOOPBACKNET 127
# define howmany(x, y) (((x)+(y)-1))/(y)
# define fd_mask __fd_mask
#else
# include <sys/param.h> /* For MAXPATHLEN and roundup() */
# include <netinet/in_systm.h> /* For typedefs */
# include <netinet/in.h> /* For IPv6 macros */
# include <netinet/ip.h> /* For IPTOS macros */
#endif
#ifdef HAVE_SYS_UN_H
# include <sys/un.h> /* For SUN_LEN */
#endif

```

includes.c

Exclude the following header file:

```

#ifdef __MVS__
#include <netinet/tcp.h>
#endif

```

log.h

In the enumeration types for SyslogFacility and LogLevel we need values for “not initialized”.

Add the following as the first entry in the definition of the type SyslogFacility:

```

/* Supported syslog facilities and levels. */
typedef enum {
    SYSLOG_FACILITY_NOT_INIT = -1,
    SYSLOG_FACILITY_DAEMON,

```

Add the following as the first entry in the definition of the type LogLevel”:

```

typedef enum {
    SYSLOG_LEVEL_NOT_INIT = -1,
    SYSLOG_LEVEL_QUIET,

```

loginrec.c

In the function `getlast_entry()` avoid the access to `wtmp`. Around line 461, add the following lines:

```

        * time, e.g. AIX */
        return 0;
# else /* DISABLE_LASTLOG */
#ifdef __MVS__
        return 0L;
#else
        /* Try to retrieve the last login time from wtmp */
# if defined(USE_WTMP) && (defined(HAVE_TIME_IN_UTMP) ||
defined(HAVE_TV_
IN_UTMP))
        /* retrieve last login time from utmp */

```

Before the last #endif in this function, add the closing #endif in this way:

```

# endif /* USE_WTMPX && (HAVE_TIME_IN_UTMPX || HAVE_TV_IN_UTMPX) */
# endif /* USE_WTMP && (HAVE_TIME_IN_UTMP || HAVE_TV_IN_UTMP) */
# endif /* DISABLE_LASTLOG */
#endif /* __MVS__ */
#endif /* USE_LASTLOG */
}

```

misc.c

Around line 472, exclude the access to `pw_passwd` and `pw_gecos` as follows:

```

        memset(copy, 0, sizeof(*copy));
        copy->pw_name = xstrdup(pw->pw_name);
#ifdef __MVS__
        copy->pw_passwd = xstrdup(pw->pw_passwd);
        copy->pw_gecos = xstrdup(pw->pw_gecos);
#endif
        copy->pw_uid = pw->pw_uid;
        copy->pw_gid = pw->pw_gid;

```

sshpty.c

OS/390 UNIX has a different naming convention for `ptys`. Fix the code beginning at line 163 as follows:

```

int i;
const char
*ptymajors="pqrstuvwxyzabcdefghijklmnopqrstuvwxyz";
const char *ptyminors = "0123456789abcdef";
#ifdef __MVS__
        int num_minors = 256;
#else
        int num_minors = strlen(ptyminors);
#endif
        int num_ptys = strlen(ptymajors) * num_minors;

```

```

        for (i = 0; i < num_ptys; i++) {
#ifdef __MVS__
            snprintf(buf, sizeof buf, "/dev/pty%c%04d", ptymajors[i /
num_minors],
                    i % 256);
            snprintf(namebuf, namebuflen, "/dev/tty%c%04d",
                    ptymajors[i / num_minors], i % 256);
#else
            snprintf(buf, sizeof buf, "/dev/pty%c%c", ptymajors[i /
num_minors],
                    ptymajors[i % num_minors]);
            snprintf(namebuf, namebuflen, "/dev/tty%c%c",
                    ptymajors[i / num_minors], ptymajors[i % num_minors]);
#endif
            *ptyfd = open(buf, O_RDWR | O_NOCTTY);
            if (*ptyfd < 0) {

```

sshconnect.c

In the function `ssh_exchange_identification()` the characters `\r` and `\n` are used. Replace `\r` with `\015` and `\n` with `\012`. After the line:

```
buf[sizeof(buf) - 1] = 0;
```

add:

```

#ifdef __MVS__
    __atoe_1 (buf, sizeof(buf));
#endif

```

In the same function, around line 416, do the following changes:

```

snprintf(buf, sizeof buf, "SSH-%d.%d-%.100s\n",
        compat20 ? PROTOCOL_MAJOR_2 : PROTOCOL_MAJOR_1,
        compat20 ? PROTOCOL_MINOR_2 : minor1,
        SSH_VERSION);
#ifdef __MVS__
    __etoe(buf);
#endif
    if (atomicio(write, connection_out, buf, strlen(buf)) !=
        strlen(buf))
        fatal("write: %.100s", strerror(errno));
#ifdef __MVS__
    __atoe(buf);
#endif
    client_version_string = xstrdup(buf);
    chop(client_version_string);
    chop(server_version_string);
    debug("Local version string %.100s", client_version_string);

```

```

}

/* defaults to 'no' */

```

In the function `check_host_key()`, completely exclude the part dealing with IPV6 as follows:

```

#ifdef __MVS__
case AF_INET6:
    local = IN6_IS_ADDR_LOOPBACK(&((struct sockaddr_in6 *)hostaddr
)->sin6_addr));
    salen = sizeof(struct sockaddr_in6);
    break;
#endif

```

ssh.c

In the function `main()`, around line 579 in the call to `log_init()` change the -1 to `SYSLOG_FACILITY_NOT_INIT` and `SYSLOG_LEVEL_NOT_INIT` respectively, as follows:

```

log_init(av0,
    options.log_level == SYSLOG_LEVEL_NOT_INIT ? SYSLOG_LEVEL_INFO :
        options.log_level,
    options.log_facility == SYSLOG_FACILITY_NOT_INIT ?
SYSLOG_FACILITY_AUTH :        options.log_facility,
    !inetd_flag);

```

sshd.c

Around line 331, change:

```

/* Send our protocol version identification. */
if (atomicio(write, sock_out, server_version_string,
strlen(server_version_string))
    != strlen(server_version_string)) {

```

by:

```

snprintf(buf, sizeof buf, "SSH-%d.%d-%.100s\n", major, minor,
SSH_VERSION);
    server_version_string = xstrdup(buf);
#ifdef __MVS__
    __etoa(buf);
#endif
    if (client_version_string == NULL) {
        /* Send our protocol version identification. */
        if (atomicio(write, sock_out, buf, strlen(buf))
            != strlen(buf)) {
            log("Could not write ident string to %s.",
get_remote_ipaddr());

```

```
fatal_cleanup();
```

Around line 320, define `macid` as:

```
char *macid="SSH-1.5-W1.0";
```

Around line 355, replace the string literal for `macid` and add the following:

```
#ifdef __MVS__
    __etoa(macid);
#endif
/* Kludge for F-Secure Macintosh < 1.0.2 */
if (i == 12 &&
    strcmp(buf, macid, 12) == 0)
    break;
continue;
}
```

Also replace `\r` with `\015` and `\n` with `\012`.

Around line 370, after the line:

```
buf[sizeof(buf) - 1] = 0;
```

add:

```
#ifdef __MVS__
    __atof(buf);
#endif
```

In the function `main()`, around line 667 in the call to `log_init()` change the `-1` to `SYSLOG_FACILITY_NOT_INIT` and `SYSLOG_LEVEL_NOT_INIT` respectively, as follows:

```
log_init(av0,
    options.log_level == SYSLOG_LEVEL_NOT_INIT ? SYSLOG_LEVEL_INFO :
    options.log_level,
    options.log_facility == SYSLOG_FACILITY_NOT_INIT ?
SYSLOG_FACILITY_AUTH : options.log_facility,
    !inetd_flag);
```

ssh_config and sshd_config

In the configuration files, uncomment the `Protocol` line as follows:

```
Protocol 1
```

ssh_prng_cmds.in

This is the command pool for the entropy daemon. In this file, more than 16 valid commands for your system must be present. In our test system, we reached this number by adding the following lines:

"ls -alni /usr/lpp"	@PROG_LS@	0.02
"ls -alni /etc"	@PROG_LS@	0.02
"ls -alni /opt"	@PROG_LS@	0.02
"ls -alni /bin"	@PROG_LS@	0.02
"ls -alni /"	@PROG_LS@	0.02
"ls -alni /etc/IBM"	@PROG_LS@	0.02
"ls -alni /var"	@PROG_LS@	0.02
"ls -alni /var/opt"	@PROG_LS@	0.02
"df -p"	@PROG_DF@	0.01
"df -v"	@PROG_DF@	0.01
"date"	/bin/date	0.01

33.5.3 OpenSSL changes

OpenSSH uses the cryptographic library (libcrypto.a) from OpenSSL for RSA, MD5, and BlowFish.

OpenSSL needs some modifications, as follows:

Configure

Around line 339, below the definitions for BS2000, add the following:

```
# OS/390 Unix an EBCDIC-based Unix system on IBM mainframe
"OS390-Unix", "c89:-g -DB_ENDIAN -DCHARSET_EBCDIC -DNO_SYS_PARAM_H
-D_ALL_SOURCE:: (unknown):: THIRTY_TWO_BIT DES_PTR DES_UNROLL MD2_CHAR
RC4_INDEX RC4_CHAR BF_PTR:::",
```

apps/s_socket.c

Around line 65, after `#include <signal.h>`, add the following:

```
#ifdef __MVS__
# define _OE_SOCKETS
# include <sys/types.h>
# include <netinet/in.h>
# include <arpa/inet.h>
#endif
```

crypto/bio/b_sock.c

Around line 64, after `#include <errno.h>`, add the following:

```
#ifdef __MVS__
# define _OE_SOCKETS
# include <sys/types.h>
# include <netinet/in.h>
# include <arpa/inet.h>
#endif
```

crypto/bio/bss_conn.c

Around line 64, after `#include <errno.h>`, add the following:

```
#ifdef __MVS__
# define _OE_SOCKETS
# include <sys/types.h>
# include <netinet/in.h>
# include <arpa/inet.h>
#endif
```

util/pod2man.pl

Around lines 730-750, change the three `/051` occurrences to: `/135`.

Around line 1048, comment out and modify the following lines:

```
sub init_noremap {
    # escape high bit characters in input stream
#    s/([\200-\377])/<E<".ord($1).">"/ge;
}

sub clear_noremap {
    my $ready_to_print = $_[0];

#    tr/\200-\377/\000-\177/;
#    s/\313/"."/ge;
#    s/B\314/""/ge;
#    s/I\314/"\n.I\n"/ge;
#    s/L\314/"\n.L\n"/ge;
#    s/\315/"^"/ge;
#    s/\315/"^"/ge;
#    s/\315L/"^"/ge;
#    s/\315R/"^"/ge;
#    s/\317//ge;
#    s/\333//ge;
#    s/\334//ge;
#    s/\335//ge;
#    s/\355/"."/ge;
#    s/\356//ge;
#    s/\357//ge;
```

33.5.4 Zlib

Zlib is a data compression library. It is used to compress the transferred data. The data format is described in RFC1950, RFC1951, and RFC1952. No modifications are needed. It builds cleanly in OS/390 UNIX/

33.5.5 Entropy gathering daemon

OpenSSH was initially written for OpenBSD. OpenBSD (like Linux) provides a device `/dev/urandom`, which provides random numbers. The numbers are not generated by a pseudo random number generator, but instead are collected from the kernel. It uses certain interrupts (network packets arriving, keyboard hits, mouse movement) to fill a entropy pool. User programs can then read a hashed portion of this pool.

In older versions of OpenSSH, for systems which didn't have such a convenient source for random bits, the implementation of a user space program to maintain an entropy pool was needed. The last version of OpenSSH includes a built-in entropy-gathering daemon. It uses the same idea of `egd.pl`, a daemon written in Perl.

This daemon uses commands like `w`, `last` and `vmstat` to collect randomness. The working commands used to fill the entropy pool must be at least 16. This is the minimum number allowed by OpenSSH. If you are interested in the original Perl daemon, you can find it at:

<http://www.lothar.com/tech/crypto>

33.5.6 OpenSSH

After the modifications, run `“autoconf”` to regenerate the configure script, run `“./configure”`, and compile with `“_CC_CCMODE=1 make”`. Install with `“make install”`.

The install sequence is:

```
$ autoconf
$ ./configure
$ _CC_CCMODE=1 make
$ make install
$ extattr +p /usr/local/sbin/sshd // only if you have BPX.DAEMON
```


Chapter 34. Patch

This chapter discusses the patch open source software package.

34.1 Introduction to patch

The patch utility takes a patch file containing a difference listing produced by *diff* and applies those changes to one or more original files, producing patched versions of those files.

This GNU version of patch includes many changes made by the Free Software Foundation. Support has been added for:

- Handling arbitrary binary data and large files
- The unified context diff format that GNU diff can produce
- Making GNU emacs-style backup files
- Improved interaction with RCS and SCCS
- The GNU conventions for option parsing and configuring and compilation
- Better POSIX.2 compliance

See the NEWS and ChangeLog in the product directory for details.

34.2 Installation of patch

The version of the package on the CD is 2.5.4. The package is a gzip file containing ASCII source code, so you'll need the GNU zip utility gunzip to install patch.

After transferring the package to the OS/390 file system, you must run these steps:

```
$ ls
patch-2_5_4_tar.gz  patch00001
$ gunzip patch-2_5_4_tar.gz
$ pax -rf patch-2_5_4_tar -o to=IBM-1047,from=ISO8859-1
$ ls -l
total 1424
drwxr-xr-x  4 AAAAAA  OMVSGRP      8192 Aug 30  1999 patch-2.5.4
-rw-r--r--  1 EGONT   OMVSGRP    706560 Mar 17 13:15 patch-2_5_4_tar
-rw-r--r--  1 EGONT   OMVSGRP     477 Mar 17 13:15 /usr/local/patch>cd
patch-2.5.4
$ cd patch-2.5.4
$ patch -b -i ../patch00001 pch.c
patching file pch.c
$ export CC='c89'
```

```

$ ./configure --host=os390
creating cache ./config.cache
checking host system type... Invalid configuration `os390': machine
`os390' not recognize
...
$ make
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g addext.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g argmatch.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g backupfile.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g basename.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g error.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g inp.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g maketime.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g partime.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g patch.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g pch.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g quotearg.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g quotesys.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g util.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g version.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g xmalloc.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g getopt1.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g getopt.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g malloc.c
c89 -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g realloc.c
c89 -o patch -g addext.o argmatch.o backupfile.o basename.o error.o
inp.o maketime.o partime.o
patch.o pch.o quotearg.o quotesys.o util.o version.o xmalloc.o
getopt1.o getopt.o malloc.o rea
lloc.o
$ make install
/bin/sh ./mkinstalldirs /usr/local/bin /usr/local/man/man1
./install-sh -c patch /usr/local/bin/`echo patch | sed 's,x,x, '`
./install-sh -c -m 644 ./patch.man /usr/local/man/man1/`echo patch | sed
's,x,x,`.`.1

```

You can ignore the messages about an invalid configuration.

34.3 Customization and operation of patch

The command `patch --version` should give you following output:

```

$ patch --version
patch 2.5.4
Copyright 1984-1988 Larry Wall
Copyright 1989-1999 Free Software Foundation, Inc.

```

This program comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of this program
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.

written by Larry Wall and Paul Eggert

You can use GNU patch in the same manner as you would use the normal patch. Refer to the man pages for more information.

An example of using GNU patch is in the groff install script. The `patch` command can use a *patch file* to modify another file. To create the patch file, the configure script to be patched is modified and saved to a new file. The `diff` command is run against the two files: the original and the modified file. The output of the `diff` command is the patch file (`../configure.patch` in the following example). Then the `patch` command with the `-i` flag is used to patch another copy of the original file. Following is an example of this command:

```
$ patch -i ../configure.patch configure
```

34.4 License

The terms of use and redistribution of patch are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify, and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

34.5 Getting patch from the Internet

The patch package was found on the mirrored GNU FTP site at:

<ftp://aeneas.mit.edu/pub/gnu/patch/>

You can try mirrored sites closer to you also. See the list of mirrored FTP sites at:

<http://www.gnu.org/order/ftp.html>

34.6 Porting patch to OS/390

There is only one simple patch required for the installation on OS/390. This small patch (named `patch00001`) is supplied on the CD; you have to apply it

to the file pch.c after downloading the package. The supplied version was written by Peter Prymmer, Forte Software. We found his notes in the perl-mvs mailing list at the following Web site:

<http://www.xray.mpe.mpg.de/mailling-lists/perl-mvs/1999-12/msg00006.html>

Chapter 35. Perl 5.005p3

This chapter discusses the Perl version 5.005p3 open source software package. See Chapter 36, “Perl 5.6.1” on page 197 for a discussion of the more recent version of Perl which is recommended.

35.1 Introduction to Perl

Perl is a general purpose, high level programming language, designed to be programmer friendly and platform neutral. As a free open software package it is available on almost all system platforms and it is the language of choice for prototypes, tools, system management, and especially in the rapidly growing area of World Wide Web programming. This makes it a mandatory tool for OS/390 systems which use a Web server on this platform to access data on the mainframe.

35.2 Installation of Perl

The version of the package on the CD is Perl 5.0 patch level 5 subversion 3. The package is a gzip file containing ASCII source code, so you'll need the GNU zip utility `gunzip` to install Perl.

Even though most of the installation steps are described in this chapter, we strongly advise you to read the `README.os390` file, which is part of the Perl distribution. It contains the latest information about the installation and customization process.

If you want to use Perl with all the default values as supplied on the CD, you don't have to follow the installation process. The `install.sh` script will do everything for you.

For an installation of the latest Perl release from the Internet, you have to execute the individual steps on your own.

35.2.1 Step 1: Prepare your system to install Perl

Check the following list and perform the necessary actions before you continue with the installation of Perl:

- The make process for Perl requires a high maintenance level of your C/C++ compiler and your language environment on OS/390.
- The Perl language will not build correctly using the Posix version of the make utility bundled with OS/390. Follow the instructions in Chapter 16,

“Gmake” on page 85, and use the GNU version of `make`. The `make` utility is automatically called during the Perl installation process, so you have to call it `make` and put in a path ahead of the normal `/bin`.

- Ensure that a copy of the parser default template file `/samples/yparse.c` is copied into your `/etc` directory. This will allow for a successful extraction of EBCDIC versions of parser files, like `perly.c`
- The configuration script will dynamically check your networking API files in the `/etc` directory. Even if you keep your active IP configuration files in MVS datasets, it's a good idea to have a copy in `/etc/resolv.conf` and/or `/etc/hosts`. This will allow the function `gethostbyname()` to return a correct value.
- Define the default loop back address `127.0.0.1` as `localhost` in your `/etc/hosts` file. Omitting this will result in errors/hangs during the Perl IVP `make test`.
- The Perl installation process will need a large virtual address region above the 16 MB boundary. If your OS/390 system limits your region above 16 MB, you should change your setup to have at least 32 MB above the line for this installation.

35.2.2 Step 2: Load the software

After you have downloaded or copied the package to your PC, transfer it in binary mode to the OS/390 HFS file system. Keep in mind that some Web browsers omit the file suffixes for known file types, like `.gz` for gzip. The GNU `unzip` utility will require the `.gz` as part of the filename, so you have to add the suffix again if it has been removed by the browser.

```
$ ls -la
-rw-r--r--  1 EGONT  OMVSGRP  3679040 Mar 15 10:21 stable_tar.gz
$ gunzip stable_tar.gz
$ ls -la
-rw-r--r--  1 EGONT  OMVSGRP  13793280 Mar 15 10:21 stable_tar
$ export $A2E="-o from=ISO8859-1,to=IBM-1047"
$ pax -rf stable_tar $A2E
$ ls -la
drwxr-xr-x 25 EGONT  OMVSGRP   16384 Mar 28  1999 perl5.005_03
-rw-r--r--  1 EGONT  OMVSGRP  13793280 Mar 15 10:21 stable_tar
```

35.2.3 Step 3: Apply patches for configuration files (Perl 5.005.03)

Even though the installation process will complete without any additional patches, it will issue some errors and abend messages. On the CD are two patches which will ensure a clean installation. The patches were written for Perl 5.005.03; they will have to be adapted to newer releases.

```

$ cd perl5.005_03
$ patch -b -i ../patch00000 Configure
Hmm... Looks like a new-style context diff to me...
The text leading up to this was:
-----
|*** Configure.orig Thu Mar 16 11:06:49 2000
|--- Configure Thu Mar 16 13:12:20 2000
-----

Patching file Configure using Plan A...
Hunk #1 succeeded at 10540.
done
$ patch -b -i ../patch01085 makedepend.SH
Hmm... Looks like a new-style context diff to me...
The text leading up to this was:
-----
|*** makedepend.SH.orig Thu Mar 16 09:17:46 2000
|--- makedepend.SH Thu Mar 16 09:22:27 2000
-----

Patching file makedepend.SH using Plan A...
Hunk #1 succeeded at 126.
done

```

35.2.4 Step 4: Configure Perl

```
$ sh Configure
```

This command will start an interactive process to guide you through the Perl configuration. Once you select os390 as your target platform, the script will automatically use defaults for most of the questions. In addition, it will automatically test your target system to customize Perl for your installation.

Unfortunately, there is no easy way to go back to a question after you hit the return key. The only option is to finish the configuration process and edit the resulting config.sh file before starting the `make`.

The configuration process can take quite a lot of time and it requires a lot of interaction with the installer. You can use `sh Configure -d` to install Perl using all default options.

At the end of the configuration process it will give you the option to stop the process and manually edit the config.sh file, or to automatically continue with the `make depend`.

35.2.5 Step 5: Build Perl

If you interrupted the configuration process, continue with:

```
$ make depend
```

Otherwise, you can continue with:

```
$ make
```

This command will now build the product in the installation directory.

35.2.6 Step 6: Installation verification process (IVP)

Perl comes with its own installation verification process (IVP) program to test the successful installation. Enter:

```
$ make test
```

At the end you should see the message “All tests successfully completed.”

35.2.7 Step 7: Copy Perl into the runtime directory

If you have installed the `groff/nroff` utility before installing Perl, the configuration script wrongly assumes that it can use the distributed man pages for Perl. There are a number of ASCII dependencies in this process, so you'll get a large number of error messages.

To prevent this, check the `config.sh` file and ensure that the following variables are set to an empty string:

```
installman1dir=''
installman3dir=''
```

You can do this using any available editor, or the `patch` utility and the `patch00003` file provided on the CD.

The command:

```
$ make install
```

will copy all runtime modules into the target directory you selected during the configuration process. The default will be `/usr/local/bin`.

35.3 Customization and operation of Perl

Here are a few steps for further customizing and enhancing Perl on the OS/390 platform.

35.3.1 Making Perl accessible from the Internet

After you have copied Perl into /usr/local/bin or an equivalent directory, this directory has to be in your PATH environment variable. If you want your Web server and CGI programs to use Perl, you have to modify PATH in your /etc/httpd.envvars file.

35.3.2 How to call Perl on OS/390

Until OS/390 R8 the *magic number* to call an interpreter from the shell using the character combination #!/usr/local/bin/perl was not working. This required that all Perl scripts had to be changed to use following syntax:

```
eval 'exec /usr/local/bin/perl -S $0 ${1+"$@"}'  
if 0;
```

Magic number support can also be installed in R7 using the new function APAR OW39467. For more information, see:

<http://www.s390.ibm.com/oe/release/apar.html>

This support allows you to start a Perl script with the more conventional:

```
#!/usr/local/bin/perl
```

35.3.3 Get an example Perl CGI script

Several easy examples of a Perl cgi script are on the CD:

- HelloWorld.pl - for pre-OS/390 R8 systems
- HelloWorldR8.pl - for OS/390 R8

To check if the Perl cgi works for you system you can call it from the UNIX shell. The result should be html code which is normally send to the browser.

35.3.4 Accessing MVS datasets from Perl

Thanks to Peter Prymmer from Forte, we have a Stdio Perl module modified for OS/390. This is available on the CD, or it can be downloaded from the Web at:

<http://www.best.com/~pvhp/os390/>
ftp://shell8.ba.best.com/pub/pvhp/public_html/os390/

Use the following steps to install it.

1. Download the package in binary.
2. gunzip OS390-Stdio-0.003.tar.gz
3. pax -rf OS390-Stdio-0.003.tar \$A2E

```
4. cd OS390-Stdoi-0.003
5. perl Makefile.PL
6. make
7. make test
8. make install
9. make -f Makefile.aperl inst_perl
```

After this you can run the test program `pds_test`, which will show you how to access an MVS PDS dataset from Perl. Refer to the README file in the installation directory for more details.

35.3.5 Useful information and Internet user groups

Although the Perl man pages are not available on OS/390, you can access the complete Perl documentation on the Web. To get a complete set of documentation in html format, point your browser to:

<http://www.cpan.org/doc/manual/html/index.html>

We also found a lot of useful information and help in the MVS-OE and in the perl-mvs user group. The current archives and sign-up informations are on:

<http://www.marist.edu/htbin/wlvindex.?mvs-oe>

and

<http://www.xray.mpe.mpg.de/mailling-lists/perl-mvs/>

Perl is a modular language and you can add functionality by installing new modules from CPAN (Comprehensive Perl Archive Network). Go to <http://www.cpan.org/> to obtain more information and to download modules.

35.3.6 Current limitations on Perl for OS/390

The Perl language has its own interface to relational databases. It consists of a language enhancement module called DBI and database-dependent (DBD) modules. These DBD modules are available for most relational databases. This includes the IBM products DB2 and UDB. At the time of this writing we are not able to use the DBI/DBD modules to access a local DB2 database on OS/390.

The current implementation of the call level interface (CLI) in DB2 for OS/390 is not compatible with the DBD for DB2. This is a real limitation because it prevents the one-to-one migration of Perl CGI code written on another platform to be ported to the mainframe Web server.

35.4 License

Perl is open source software. It's free for you to download and use as you wish. Perl's license is called the Artistic license. You can read details on the Perl Web site at:

<http://www.perl.com/language/misc/Artistic.html>

The bottom line is that the Artistic license is a kinder and gentler version of the GNU license (see 2.1, "The GNU General Public License (GPL)" on page 13 for details)—one that doesn't infect your work if you care to borrow from Perl or package up pieces of it as part of a commercial product!

It is interesting that the Perl "kit" (package) is also shipped with the GPL (in the file COPYING) as well as the Artistic license (in the file Artistic). The file README discusses these two:

```
Perl Kit, Version 5.0
```

```
Copyright 1989-1999, Larry Wall  
All rights reserved.
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of either:
```

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License" which comes with this Kit.

However, it is not stated which license this package is shipped under. It is being redistributed, but it has not been modified.

35.5 Getting Perl from the Internet

You may want to obtain the latest copy of this package. It is on the Web at:

<http://www.perl.com/CPAN/src/stable.tar.gz>.

You can still find the older release of Perl at:

<http://www.perl.com/pub/language/info/software.html#stable>

35.6 Porting Perl to OS/390

Most of the OS/390-specific changes have been integrated into the mainline of the Perl code. You can install the product as it comes from the CD or from the Web.

However, there are a few problems that cause abends and error messages during the installation. For more details or an example of the required patches, see the perl-mvs mailing list archive at these sites:

www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1999-11/msg01085.html

www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1999-12/msg00000.html

Chapter 36. Perl 5.6.1

This chapter discusses the Perl 5.6.1 software package.

Note: Chapter 35, “Perl 5.005p3” on page 189, contains a discussion of an older version of Perl; however, using that version is not recommended since it does not natively build on OS/390.

36.1 Introduction to Perl 5.6.1

Perl is a general purpose, high-level programming language, designed to be programmer-friendly and platform-neutral. As a free open software package, it is available on almost all system platforms and is the language of choice for prototypes, tools, system management, especially in the rapidly growing area of World Wide Web programming. This makes it a mandatory tool for OS/390 systems which use a Web server on this platform to access data on the mainframe.

36.2 Installation of Perl 5.6.1

The version of the package on the CD is Perl 5.6.1. The package is a gzip file containing ASCII source code, so you'll need the GNU zip utility `gunzip` to install Perl.

Even though most of the installation steps are described in this chapter, we strongly advise you to read the `README.os390` file, which is part of the Perl distribution. It contains the latest information about the installation and customization process.

Note: Do *not* try to install the older Perl 5.6.0. Since the development versions of Perl leading up to 5.6.0 incorporated an anti-ebcdic input file algorithm (basically, input with the eighth bit set treated as data that needs to undergo an ISO 8859-1 -> utf8 transform), Perl 5.6.0 will not build on OS/390.

The heart of the installation script `perl.install` is as follows:

```
gzip -c -d perl-5.6.1.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s perl-5.6.1 perl56
cd perl56
echo "creating a makefile via the command './Configure -d' ..."
./Configure -des -Dprefix=$installDir
echo "building Perl via the command 'make' ..."
make
make install
```

36.3 License

The License for Perl 5.6.1 is the same as 5.005p3; refer to 35.4, “License” on page 195.

36.4 Getting Perl 5.6 from the Internet

The version of the package on the CD is 5.6.1. You may want to obtain the latest copy of this package. It is on the Web at:

<http://www.perl.com/CPAN/src/stable.tar.gz>.

36.5 Porting Perl 5.6 to OS/390

All the OS/390-specific changes have been integrated into the mainline of the Perl code. You can install the product as it comes from the CD or from the Web.

Chapter 37. PHP

This chapter discusses the PHP software package.

37.1 Introduction to PHP

The following text was taken from the PHP Web site:

PHP is an open source programming language used to build dynamic Web pages. A dynamic Web page is a page that interacts with the user so that each visitor to the page sees customized information.

Dynamic Web applications are prevalent in commercial sites where the content displayed is generated from information accessed in a database or other external source. PHP scripts are embedded within a Web page along with HTML, similar to other scripting languages such as Microsoft's Active Server Pages (ASP) or Sun Microsystems' Java Server Pages (JSP).

Like ASP and JSP, PHP is a server-side language, meaning it runs on the Web server, rather than on the Web browser or other client. PHP can be used for any application development that uses a Web browser as the user interface. Because of user preferences for and familiarity with Web browsers, the use of browser-based solutions for mission-critical enterprise applications is increasing rapidly.

With hardly any publicity, PHP has swept the Web. According to the authoritative Netcraft survey of what technology is actually in use on the Web, PHP can now be found on more than 6 million domains, and is growing at a rate of up to 15% each month. PHP is available on over 36% of Apache Web servers - the most common server on the Web. The latest version of PHP, PHP 4, was downloaded 265,000 separate times in the first two months it was available.

Why is PHP so popular? Programmers who work with PHP say it is:

- Ideal for dynamic and interactive Web sites
- Supports a wide variety of back-end databases
- Easy to use and easy to learn
- Designed for the Web, from the ground up
- Fast
- Extremely reliable
- Cross platform - cross-OS, cross Web server and highly portable
- Compatible with other key Web technologies, including Java, COM, XML and Macromedia Flash

37.2 Installation of PHP

Attention

During the beta distribution of the open source software associated with this redbook, a number of problems were found with PHP. There was no time to address these issues. This port of PHP should be considered *alpha* quality.

In the nature of open source software, the code will remain on the Tools and Toys Web site with the hopes the quality will improve.

Do not consider this port anywhere near ready for production.

The PHP package installs an Apache http Server with a PHP module. Because you can only have one Apache server at a certain location, PHP is not part of the main installation script.

Following is the prerequisite knowledge and software:

- An ANSI C compiler
- Flex and bison (for compiling)
- A Web server (typically Apache)
- Any module-specific components (such as gd, pdf libs, etc.)

Here we describe how to build PHP as an Apache module. First build Apache (refer to 5.2, "Installation of Apache" on page 33). Then install PHP; following is the core of the PHP install script:

```
archive=$1
apachedir=$2
apacheinstalldir=$3
gzip -c -d php-4.0.5.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s php-4.0.5 php
cd php
export CFLAGS="-O -DIS_EBCDIC_HOST -D_ALL_SOURCE "
export RANLIB=echo
export LD=echo
LDFLAGS=-Wl,EDIT=NO ./configure --enable-debug --disable-pear
--disable-posix --with-regex=system --with-apache=$apachedir
--without-mysql --with-ibm-db2=/usr/local/db2/sqlllib
echo "building via the command 'make' ..."
make
```



```

rc=$?
echo "return code from make = $rc"
if [ $rc != 0 ]; then
    exit $rc
fi
make install
echo "rebuilding apache with php"
cd $apachedir
export _C89_CC_MODE=1
LD_FLAGS=-Wl,EDIT=NO ./configure --verbose --prefix=$apacheinstalldir
--activate-module=src/modules/php4/libphp4.a --enable-module=status
make
make install

```

After Apache installs successfully, and using the **php.install** script, install PHP with the following command:

```
$ php.install php-4.0.5.tar.gz /usr/local/src/apache/apache /usr/local/
```

The second parameter is the Apache source directory, and the third parameter is the Apache install directory.

After PHP builds successfully, copy the `php.ini-dist` file to the directory `/usr/local/lib/` with the new file name `php.ini`. Then edit our `httpd.conf` file and add the following:

```
AddType application/x-httpd-php .php
```

Finally, restart the Apache server.

37.3 Customization and operation of PHP

The configuration file called `php.ini` is read when PHP starts up. For the server module versions of PHP, this happens only once, when the Web server is started. For the CGI version, it happens on every invocation.

The sample ini file is correct for normal installations, but you can check the complete option list on the online php manual at the following site:

<http://www.php.net/manual/en/configuration.php>

37.4 License

Following is the PHP License, version 2.02:

“Copyright (c) 1999, 2000 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "PHP" must not be used to endorse or promote products derived from this software without prior permission from the PHP Group. This does not apply to add-on libraries or tools that work in conjunction with PHP. In such a case the PHP name may be used to indicate that the product supports PHP.
4. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
5. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes PHP, freely available from <http://www.php.net/>".
6. The software incorporates the Zend Engine, a product of Zend Technologies, Ltd. ("Zend"). The Zend Engine is licensed to the PHP Association (pursuant to a grant from Zend that can be found at <http://www.php.net/license/ZendGrant/>) for distribution to you under this license agreement, only as a part of PHP. In the event that you separate the Zend Engine (or any portion thereof) from the rest of the software, or modify the Zend Engine, or any portion thereof, your use of the separated or modified Zend Engine software shall not be governed by this license, and instead shall be governed by the license set forth at <http://www.zend.com/license/ZendLicense>.

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM
``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.

IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

37.5 Getting PHP from the Internet

The version of the package we worked on was 4.0.5, found on the Web at:

<http://www.php.net/>

37.6 Porting PHP to OS/390

Before starting, we checked to see if any attempt to port PHP to an EBCDIC world had already been tried. However, although many people have been asking for such a port, it appeared that nobody had succeeded in porting it.

This kind of information can be found in the PHP developer’s mailing list archive:

<http://marc.theaimsgroup.com/?l=php-dev>

We decided to attempt the port (while realizing that it wouldn’t be easy—and without much time in our budget) to give at least a starting point to future enhancements.

Our work was divided into three steps:

- Configure and compile PHP as an Apache module
- Build and run this Apache/PHP server
- Make PHP correctly interpret the language

We successfully completed the first two steps, but PHP offers too much functionality to test every single command. We were able to correct all the problems we encountered, but there might be more out there. One cause for trouble might be the zend interpreter, which needs bison 1.28 to be built, while we only had access to bison 1.25 (included on the CD).

In the following sections we describe the problems we encountered, as well as the actions we took to resolve them.

First we copied into our PHP directory some files from autoconf to correctly detect OS/390:

```
$ cd php-4.0.5
$ cp /usr/local/autoconf/autoconf/config.guess .
$ cp /usr/local/autoconf/autoconf/config.sub .
$ cp /usr/local/autoconf/autoconf/install-sh .
```

Then, to avoid problems with CR/NL/SP in libtool, we changed ltmain.sh to do OS/390-specific translations to EBCDIC codes:

```
71,80c71,72
< case "$host" in
<   *os390*)
<     SP2NL='tr \100 \025'
<     NL2SP='tr \025 \100'
<     ;;
<   *)
<     SP2NL='tr \040 \012'
<     NL2SP='tr \015\012 \040\040'
<     ;;
<   esac
---
> SP2NL='tr \040 \012'
> NL2SP='tr \015\012 \040\040'
```

After this patch, we were able to configure the package by running the following:

```
export CFLAGS="-O -DIS_EBCDIC_HOST -D_ALL_SOURCE " \
export RANLIB=echo \
export LD=echo \
./configure \
  --enable-debug \
  --disable-pear \
  --disable-posix \
  --with-regex=system \
  --with-apache=/usr/local/apache-php/src/apache_1.3.19 \
  --without-mysql
```

After the configure runs successfully, we had to manually correct a few files. In ./main/php_config.h file, we made the following changes:

```
< #define HAVE_GETRUSAGE 1
---
> #undef HAVE_GETRUSAGE
469c469
< #define HAVE_STRtok_R 1
---
```

```

> #undef HAVE_STRtok_R
874c874
< #define HAVE_GETADDRINFO 1
---
> #undef HAVE_GETADDRINFO

```

However, since this file is regenerated upon configuring PHP, we #ifdef'ed the above changes. In ./ext/standard/mail.c , after line 45, we added the following:

```

#ifdef __MVS__
#define EX_OK 0
#define EX_TEMPFAIL 1
#endif

```

In the file ./ext/standard/syslog.c, we changed line 58:

```
REGISTER_LONG_CONSTANT("LOG_SYSLOG", LOG_SYSLOG, CONST_CS | CONST_PERSISTENT);
```

to:

```

#ifdef __MVS__
REGISTER_LONG_CONSTANT("LOG_SYSLOG", LOG_SYSLOG, CONST_CS | CONST_PERSISTENT);
#endif

```

We also changed line 147 from:

```
SET_VAR_LONG("LOG_SYSLOG", LOG_SYSLOG);
```

to:

```

#ifdef __MVS__
SET_VAR_LONG("LOG_SYSLOG", LOG_SYSLOG);
#endif

```

In the ./ext/standard/url.c file, after line 23, we added:

```

#ifdef IS_EBCDIC_HOST
#include "ebcdic.h"
#endif

```

We also copied the file /path_to_apache_dir/src/os/os390/ebcdic.h to ./ext/standard/ebcdic.h.

In the ./ext/standard/microtime.c file, we made the following change:

```

< #ifdef HAVE_GETRUSAGE
---
> #if HAVE_GETRUSAGE && !__MVS__

```

In the `./Zend/zend_ini_scanner.c` file, in line 696, we made the following change:

```
> (c = getc( yyin )) != EOF && c != '\n'; ++n ) \  
< (c = __etoa(getc( yyin ))) != EOF && c != '\n'; ++n ) \  

```

This patch permits PHP to read its `php.ini` file in EBCDIC format.

Also, we needed to touch all the `.y` and `.l` files in `./Zend/` to update the modification date, so during `make` they will be processed to create the new `.c` files during build.

The last patch fixes the problem with displaying the date. In `./main/php_reentrancy.h`, we changed lines 40 to 45 to the following:

```
#if PHP_HPUX_TIME_R || __MVS__  
#undef HAVE_LOCALTIME_R  
#undef HAVE_ASCTIME_R  
#undef HAVE_CTIME_R  
#undef HAVE_GMTIME_R  
#endif
```

Functions named `*_r` are thread-safe equivalents to the standard functions. On OS/390 UNIX, the normal versions are thread-safe so one can use them directly.

Finally, we ran **make**:

```
$ export _CC_CCMODE=1  
$ make  
$ make install
```

This copied PHP to the Apache directory tree. The last step was to rebuild Apache with the PHP module:

```
$ cd ../apache_1.3.19  
$ export _C89_CCMODE=1  
$ ./configure --verbose --prefix=/apache/install/directory \  
  --activate-module=src/modules/php4/libphp4.a --enable-module=status  
$ make
```

Then the server must be stopped (if it is already running), as follows:

```
$ apachectl stop  
$ make install
```

Before restarting Apache, we had to install the `php.ini` files and change `httpd.conf`:

```
$ apachectl start
```

At this point in the porting effort, PHP was running and correctly interpreting many of its functions. As previously mentioned, while we could not check all the functions offered from a complete PHP build, due to the huge number of them, we did run some simple tests and noted that generally the internal functions worked well.

Chapter 38. RCS

This chapter discusses the Revision Control System (RCS) open source software package.

38.1 Introduction to RCS

The following description was obtained from the GNU Web site:

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

38.2 Installation of RCS

Here is the core of the install script named `rsc.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
pax -rzf rcs-5.7.tar.Z
ln -s rcs-5.7 rcs
cd rcs-5.7/src
make
for i in ci rcs rcsfreeze.sh rlog co ident rsclean rcsmerge merge \
    rcsdiff
do
    cp $i $installDir/bin/$i
done
cd ../man
for i in *.1
do
    cp $i $2/man/$LANG/cat1/$i
done
```

38.3 Customization and operation of RCS

To use the RCS package, you normally create a directory named `RCS` in the directory of source code your team wants to maintain.

RCS consists of the four frequently used shell commands `ci`, `co`, `rlog`, and `rcsdiff`, as well as the less common commands `srcs`, `ident`, `rsclean`, and `rcsmerge`. Following are the common commands with the flags typically used for various operations:

- `ci -u filename` - Check the latest changes into the database, release the lock, and keep an unlocked read-only working file. (Without `-u`, `ci` would remove your working file and you would have to use `co` to get it back.)
- `ci -l filename` - Check the latest changes into the database, but do not release the lock and keep a locked, writable working file.
- `co filename` - Check out an unlocked latest-version read-only working file from the database.
- `co -l filename` - Retrieve a locked latest-version writable working file from the database. Only one user can have such a writable locked version at a time.
- `rlog filename` - Show the revision history of a file.
- `rcsdiff -u filename` - Show the differences between your working file and the latest revision in the database.

To learn more about RCS (and CVS), see the HOW-TO on the linuxdoc.org Web site at:

http://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/html_single/

From there, search for the document *CVS-RCS-HOWTO.html*.

38.4 License

The terms of use and redistribution of RCS are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

38.5 Getting RCS from the Internet

The RCS package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 5.7. The package was found on the MKS ftp server at:

<ftp://ftp.mks.com/pub/s390/gnu/>

The package is a compressed tar file in EBCDIC, so it was unwound with the following `pax` command:

```
$ ls -l
```

```

total 2560
-rw-r----- 1 MIKEM   SYS1      1157120 Feb 22 09:08 rcs-5_7_os390_tar.Z
$ pax -rzf *.Z
$ ls -l
total 2664
-rw-r----- 1 MIKEM   SYS1      1157120 Feb 22 09:08 rcs-5_7_os390_tar.Z
drwxr-xr-x  2 MIKEM   SYS1        8192 May 13 1997 src/
-rwxr-xr-x  1 MIKEM   SYS1        2452 May 13 1997 README*
drwxr-xr-x  2 MIKEM   SYS1        8192 May 13 1997 programs/
drwxr-xr-x  2 MIKEM   SYS1        8192 May 12 1997 manpages/

```

38.6 Porting RCS to OS/390

A README file is included in the package, with some high-level notes about the port. The OS/390 changes to the code can be gauged with the command:

```
grep __MVS__ *. [ch]
```

The README file discusses how OS/390 UNIX `diff` and the `diff3` command included don't support comparison of binary files. Because the `diff` and `diff3` commands included in the `diffutils` package (See Chapter 10, "Diffutils" on page 61) do support binary comparisons, RCS should now support binary files; however, this was not tested.

Some changes were made to the structure of the directories. Originally, there was a subdirectory named `programs`. Since the philosophy of this redbook is to keep all the binaries in one tar file and all the source in another, the directory and the executables were removed. The `diff3` command is no longer needed, as mentioned earlier. Also, a couple of scripts were moved to the `src` directory.

```
rcstest      Test the RCS commands.
rcsfreeze.sh Freeze a configuration of sources checked in under RCS.
```

There is no `configure` script included in the RCS package, but there is a Makefile and a `conf.h` header file. Therefore, it is tricky to install RCS in a location other than `/usr/bin`, which appears to be the default. In order to modify the install location, the Makefile and `conf.h` header file have to be modified before the code is built. This is accomplished in the `rcs.install` script with the following `sed` commands:

```

cat Makefile | sed "s+prefix = /usr+prefix = $installDir+g" \
              | sed "s+/usr/bin/diff3+$installDir/bin/diff3+g" \
              | sed "s+ /bin/diff+ $installDir/bin/diff+g" > Makefile.new
mv Makefile Makefile.orig
mv Makefile.new Makefile

```

```
cat conf.h | sed "s+/usr/bin+$installDir/bin+g" > conf.h.new
mv conf.h conf.h.orig
mv conf.h.new conf.h
```

We made one change to the Makefile. We set LDFLAGS to -Wl,EDIT=NO so the executable sizes are much smaller.

Chapter 39. Rlogin client

This chapter discusses the rlogin client open source software package.

39.1 Introduction to the rlogin client

The rlogin client performs a function similar to that of the telnet client: it allows you to remotely log in to a UNIX server over the network.

39.2 Installation of the rlogin client

In order to run, rlogin must call `setuid()` to set the UID. In order for this call to succeed, the set-user-ID bit must be set. This is illustrated in the following session:

```
$ ./rlogin wtsc55oe
rcmd: socket: EDC5111I Permission denied.
$ chmod u+s rlogin
$ ls -l rlogin
-rwsr-xr-x  1 MIKEM   SYS1      200704 Dec 16  1998 rlogin*
```

Here is the core of the install script named `rlogin.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
pax -rzf rlogin.tar.Z
cd rlogin
make
cp rlogin $installDir/bin/rlogin
cp rlogin.1 $installDir/man/$LANG/cat1/rlogin.1
chmod u+s $installDir/bin/rlogin
if [ ! -u $installDir/bin/rlogin ]; then
    echo "Error: chmod u+s $installDir/bin/rlogin failed - set-user-ID \
    bit must be set for rlogin to work!"
fi
```

39.3 Customization and operation of rlogin client

After proper installation, the rlogin client enables users to remotely log in to a UNIX server (which has the rlogin server running) over a TCP/IP network.

The syntax for using rlogin is as follows:

```
rlogin [-8EKldx] [-e char] [-k realm] [-l username] host
```

An example of using rlogin with the `-l` flag is as follows:

```
rlogin -l mikem wtsc.itso.ibm.com
```

The user is then logged into the remote machine, which appears to take over his current session.

39.4 License

The terms of use and redistribution of the rlogin client are governed by the Regents of the University of California copyright. It was found as comments in the source files as follows:

```
/*
 * Copyright (c) 1983 The Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by the University of California, Berkeley. The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED 'AS IS' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 */
```

39.5 Getting the rlogin client from the Internet

The rlogin client package is available on the CD that accompanies the hard copy of this redbook. The version of the package on the CD was found on the IBM OS/390 Tools and Toys Web site at:

<http://www.s390.ibm.com/ftp/os390/oe/port/RLOGIN/rlogin.tar.Z>

The package was downloaded to the PC and then moved to OS/390 in binary via FTP. The text data is already in EBCDIC, so no translation was necessary. The package was uncompressed and unwound with a single `pax` command:

```
$ ls -l
total 512
-rw-r--r--  1 MIKEM   SYS1      92160 Feb 15 11:01 rlogin.tar.Z
$ pax -rzf *
$ ls -l
total 544
```

```

-rw-r--r--  1 MIKEM   SYS1      92160 Feb 15 11:01 rlogin.tar.Z
drwxr-xr-x  2 MIKEM   SYS1      8192 Dec 16  1998 rlogin/

```

39.6 Porting the rlogin client to OS/390

This package was ported to OS/390 by Jean Yves Girard of IBM France. When unwound on OS/390, the package contains an executable, rlogin, and the source code:

```

$ ls -l
total 488
-rw-r--r--  1 MIKEM   SYS1        394 Dec 16  1998 README
-rwxr-xr-x  1 MIKEM   SYS1    200704 Dec 16  1998 rlogin*
-rwxr-xr-x  1 MIKEM   SYS1    27580 Dec 16  1998 rlogin.c*
-rw-r--r--  1 MIKEM   SYS1        292 Dec 16  1998 Makefile
-rwxr-xr-x  1 MIKEM   SYS1       7327 Dec 16  1998 rcmd.c*
-rwxr-xr-x  1 MIKEM   SYS1       2062 Dec 16  1998 rresvport.c*

```

Building the package was as simple as typing `make`:

```

$ make
cc -g -c -D_ALL_SOURCE rlogin.c
cc -g -c -D_ALL_SOURCE rcmd.c
cc -g -c -D_ALL_SOURCE rresvport.c
cc -g -o rlogin -L. -D_ALL_SOURCE rlogin.o rcmd.o rresvport.o

```

To keep with the philosophy of one source tar file and one binary tar file, the rlogin executable and object files were removed from the package on the CD. Also, to keep with the philosophy of the smallest possible executable, the `-g` flag was removed from the compile steps and the link flag `-Wl,EDIT=NO` was added to the Makefile.

Chapter 40. Rsh

This chapter discusses the rsh open source software package.

40.1 Introduction to rsh

The rsh program executes a command on a remote UNIX server. It does this by copying standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. In this fashion, it appears that the remote command is being executed locally. The interrupt, quit and terminate signals are propagated to the remote command; rsh normally terminates when the remote command terminates.

40.2 Installation of rsh

Here is the core of the install script named rsh.install, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
pax -rzf rsh.tar.Z
cd rsh
make
rm *.o
cp rsh $installDir/bin/rsh
cp rsh.1 $installDir/man/$LANG/cat1/rsh.1
```

40.3 Customization and operation of rsh

To use rsh, you must first establish authorization. This can either be done via the `/etc/hosts.equiv` file on the local machine, or the `$HOME/.rhosts` file on the remote machine.

Then there is an issue with low port numbers. The rsh protocol requires that the rsh client obtain a low port. Traditionally, low ports are reserved for superusers (users with UID zero). If you have `TCPCONFIG RESTRICTLOWPORTS` in your TCP/IP profile, users with non-zero UID will be unable to use this rsh client unless an additional configuration step is performed. A typical symptom is shown below:

```
$ rsh -l user1/group1 10.1.1.1 ls
rcmd: rresvport: EDC5111I Permission denied.
Reason code: 744C7246
```

One of the following steps should be performed if users with non-zero UID are allowed to use rsh:

1. Make the rsh executable suid to a UID zero user ID. The owner of the rsh executable file should have UID zero. Use the following command to let rsh run with UID zero, replacing `mypath` with the name of the directory containing rsh:

```
$ chmod u+s /mypath/rsh
```

2. Make the rsh executable run with APF authorization. The rsh executable file was link-edited with AC=1. All that is needed to allow it to run with APF authorization is to set the HFS APF attribute. Use the following command to let rsh run with APF authorization, replacing `mypath` with the name of the directory containing rsh:

```
$ extattr +a /mypath/rsh
```

Once properly authorized, the `rsh` command syntax is of the form:

```
rsh [-Kdnx] [-k realm] [-l username] host [command]
```

For example:

```
$ rsh otherhost cat remotefile >> localfile
```

40.4 License

The terms of use and redistribution of the rsh client are governed by Regents of the University of California copyright. It was found as comments in the source files as follows:

```
/*-
 * Copyright (c) 1983, 1990, 1993, 1994
 *   The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *   must display the following acknowledgement:
 *     This product includes software developed by the University of
 *     California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *   may be used to endorse or promote products derived from this software
 *   without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
```

```
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
```

40.5 Getting rsh from the Internet

The rsh package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is FreeBSD 1.4.4.2. The package was obtained from the person who ported it. The binary is on the OS/390 Tools and Toys Web page (ported tools) at:

<http://www.s390.ibm.com/oe/bpxalty1.html>

The latest version of the rsh source (without OS/390 changes) is available at:

<ftp://ftp.FreeBSD.org/pub/FreeBSD/>

Click the **FreeBSD-stable** directory; then **src -> usr.bin -> rsh**.

The package on the CD is in ASCII, so it can be unwound as follows:

```
$ ls -l
total 160
-rw-r----- 1 MIKEM  SYS1      64512 Feb 29 11:16 rsh_ascii.tar
$ pax -rf *.tar $A2E
$ ls -l
total 160
drwxr-xr-x  3 MIKEM  SYS1      8192 Feb 29 13:01 rsh/
-rw-r----- 1 MIKEM  SYS1      64512 Feb 29 11:16 rsh_ascii.tar
```

40.6 Porting rsh to OS/390

As the porting of rsh was done at another time, no details of the port are available.

Chapter 41. Samba

This chapter discusses the Samba version 2.2.1a open source software package. For information about the older, 1.9.18p1 version, refer to Chapter 42, “Samba 1.9.18p1” on page 231.

Attention

During the beta distribution of the open source software associated with this redbook, a number of problems were found with Samba. There was no time to address these issues. This port of Samba 2.2.1a should be considered *alpha* quality. In the nature of open source software, the code is being made available with the hope that the quality will improve.

It is recommended that you use the Samba 1.9.18p1 version, which has proven to be more stable.

41.1 Introduction to Samba

Samba is a Windows file server for UNIX. Samba is actually a suite of programs that implements the Server Message Block (SMB) protocol. See the IBM Redbook *S/390 File and Print Serving*, SG24-5330, for details of Samba for OS/390.

The port that is used is Samba version 2.2.1a. This is not considered a complete port, since we did not have the time and resources to implement all features. It works well as a file server for Windows clients.

41.2 Installation of Samba

Here is the core of the install script named `samba22.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`). Also, after the executables are installed via `make install`, the code pages are removed (per the `README.OS390` file). Here is the core of the install script:

```
gzip -c -d samba-2.2.1a.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s samba-2.2.1a samba
cd ./samba/source
```

```
export LD=echo
export CFLAGS="-D_ALL_SOURCE"
CC=c89 ./configure --prefix=$installDir
make
make install
```

41.3 Customization and operation of Samba V2

Again, refer to the Samba chapter in *S/390 File and Print Serving*, for complete details of customizing Samba.

The important steps to remember are:

- Get proper authorization to smbd
- Define shares in the smb.conf file
- Get smbd running
- Get clients configured properly (server name, password encryption)

The Samba Web Administration Tool (SWAT) is a new tool that is part of Samba 2.x. It enables a user-friendly browser interface to administer all aspects of Samba. We started porting SWAT to OS/390. Unfortunately, this port is not yet ready for production. If you do any changes to the Samba configuration via SWAT, we recommend that you check the changes afterwards by looking at the smb.conf file.

A nice feature of SWAT is the existence of help links for all configured options. This is a good start if you are looking for documentation on Samba. For further discussion about getting SWAT onto OS/390, see 41.6.3, “Porting SWAT” on page 228.

Refer to the Samba chapter of *S/390 File and Print Serving*, SG24-5330, for details on customizing Samba.

41.4 License

The terms of use and redistribution of Samba are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

41.5 Getting Samba V2 from the Internet

The Samba package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 2.2.1a. It was found on the Web at:

<ftp://us1.samba.org/pub/samba/samba-latest.tar.gz>

41.6 Porting Samba V2 to OS/390

In order to configure Samba, we made the following changes to the configure script in order to support wide characters:

```
#ifdef __MVS__
#include "wchar.h"
#endif
```

and:

```
#ifndef __MVS__
#define ISLOWER(c) ('a' <= (c) && (c) <= 'z')
#define TOUPPER(c) (ISLOWER(c) ? 'A' + ((c) - 'a') : (c))
#else
#define ISLOWER(c) islower(c)
#define TOUPPER(c) toupper(c)
#endif
```

in order to pass the test for ANSI C header files.

41.6.1 Porting `smbd` and `smbclient`

OS/390 UNIX does not have a `/etc/passwd` file, which is generally used by the Samba code to check for passwords. On Linux and most UNIXes, the `passwd` structure is as follows:

```
struct passwd
{
    char *pw_name;           /* Username. */
    char *pw_passwd;         /* Password. */
    __uid_t pw_uid;          /* User ID. */
    __gid_t pw_gid;          /* Group ID. */
    char *pw_gecos;          /* Real name. */
    char *pw_dir;            /* Home directory. */
    char *pw_shell;          /* Shell program. */
};
```

OS/390 has neither the `pw_passwd` field nor the `pw_gecos` field. These are used to store the encrypted password and the user's full name. All the

occurrences of `pw_passwd` of the struct `passwd(pwd.h)` were `#ifdef`'ed out of the code by us, and all occurrences of `pw_gecos` were changed to `pw_name`.

OS/390 has a special function, `__passwd()` to validate passwords. So we made the following change to `pass_check.c`:

```
#if __MVS__
    DEBUG(11, ("Checking user \"%s\" with password \"%s\"\n",
               this_user, password));
    return(__passwd(this_user, password, NULL) == 0);
#endif
```

Group passwords are not available in OS/390 (variable `gr_passwd` is not a member of struct `passwd`). This functionality is not required, so we `#ifdef`'ed all occurrences of the same.

The incoming packet is checked for the characters `"\377SMB"` in order to determine whether it is a SMB packet or not. We needed to convert the string `"SMB"` to ASCII and compare it with the incoming packet (which is in ASCII, too) in `source/smbd/process.c` as follows:

```
#ifndef __MVS__
    if (strcmp(smb_base(inbuf), "\377SMB", 4) != 0)
#else
    /* ASCII values for SMB */
    if (strcmp(smb_base(inbuf), "\377\123\115\102", 4) != 0)
#endif
```

The string `"SMB"` is reconverted to EBCDIC, otherwise it gets reconverted to ASCII every time, even though it is in ASCII.

In order to make the SMB protocol to successfully work between ASCII clients and EBCDIC servers, we needed to make the requisite ASCII and EBCDIC changes to the incoming and outgoing packets.

Note: These changes were required only to *strings* and not to integers in the SMB headers, for example.

Another minor problem was that the `islower()`, `isupper()`, `tolower()`, `toupper()` functions worked selectively on certain letters only. This is because the Samba source code redefined these functions in `lib/charset.c`. We `#ifdef`'ed these redefinitions to use the standard functions of OS/390.

The following are some of the diffs to the changes made to the Samba source code:

The file source/include/proto.h

```
diff -r /u/samba22/samba/source/include/proto.h
89,92d88
< /*The following definitions come from lib/ebcdic.c */
<
< int translate_mode(char *name,int snum);
<
1881d1876
< BOOL lp_atoe(int );
```

The file source/lib/util.c

```
diff -r /u/samba22/samba/source/lib/util.c ./lib/util.c
328d327
< #ifndef __MVS__
332,337d330
< #else
< /* 'SMB' in EBCDIC */
< CVAL(buf,5) = 0x53;
< CVAL(buf,6) = 0x4D;
< CVAL(buf,7) = 0x42;
< #endif
1486d1478
< /* __atoe_l(buf, len);*/
```

The file source/nmbd/nmbd_subnetdb.c

```
diff -r /u/samba22/samba/source/nmbd/nmbd_subnetdb.c
./nmbd/nmbd_subnetdb.c
133d132
< #ifdef __MVS__
135,139d133
< nmb_sock = ClientNMB;
< dgram_sock = ClientDGRAM;
< }
< #else
< {
175d168
< #endif
```

The file source/smbd/ipc.c

```
diff -r /u/samba22/samba/source/smbd/ipc.c ./smbd/ipc.c
71,73d70
< #ifdef __MVS__
< __etoa_l( &rdata[data_offset],data_len);
< #endif
486,488d482
< #ifdef __MVS__
< __atoe(name);
```

```
< #endif
```

The file source/smbd/mangle.c

```
diff -r /u/samba22/samba/source/smbd/mangle.c ./smbd/mangle.c
1023,1025c1023
< #ifdef __MVS__
< __etoa(OutName);
< #endif
---
>
```

The file source/smbd/process.c

```
diff -r /u/samba22/samba/source/smbd/process.c ./smbd/process.c
617,621d616
<         char *smb = "SMB";
< #ifdef __MVS__
< __etoa(smb);
< #endif
<
631c626
< if (strcmp(smb_base(inbuf)+1, smb, 3) != 0)
---
> if (strcmp(smb_base(inbuf), "\377SMB", 4) != 0)
638,640d631
< #ifdef __MVS__
< __atoe(smb);
< #endif
```

41.6.2 Text character translations

Typically, text data is represented in EBCDIC on OS/390, so if it is to be viewed on an ASCII-based computer, it must be translated between the two formats.

However, this does not always have to be the case. If text data is to be viewed only by ASCII computers, it never has to be converted to EBCDIC, regardless of where it is stored. This is the default behavior with Samba, since no other operating system that it runs on uses EBCDIC to store text data.

If data is not converted to EBCDIC, any system expecting text data in that format will not be able to read it. For example, if text data were not converted to EBCDIC, an rlogin session to OS/390 UNIX would not be able to view the data, since the rlogin daemon (server) running on OS/390 UNIX translates from EBCDIC to ASCII.

Also, ASCII data could not be accessed from OS/390 unless the appropriate translation were first implemented. If it is necessary that text data be stored in EBCDIC on OS/390, there is an issue of *when* to convert between ASCII and EBCDIC. In the port of Samba to OS/390, we added two extensions to the `smbd.conf` file to make this process easier.

These extensions use the Windows PC concept of a file type, which is the suffix following the period in the name of the file. The new configuration parameters are:

```
translate map = pattern,text or translate map = pattern,binary
```

In this case, *pattern* is of the form `*.type`, which is a global parameter to specify which file types are to be converted.

```
text conv = filetype,text or text conv = filetype,binary
```

This is a parameter set on a per resource basis that works in conjunction with the `translate map` parameter. It specifies the conversion for all files that are not matched by the `translate map` parameter.

The following parameter examples in the `smb.conf` file specify that files of type `.txt`, `.pl`, `.c`, `.html` and `.htm` are text by their file name convention and should be converted between ASCII and EBCDIC. Also, the `text conv` parameter specifies that for the share named `smbdir`, all other file types are binary and are not to be converted.

```
[global]
...
translate map = *.txt,text
translate map = *.html,text
translate map = *.htm,text
translate map = *.pl,text
translate map = *.c,text
...
[smbdir]
text conv = filetype,binary
...
```

One of the limitations of `translate map` is that when the `smbclient` is run on an OS/390 UNIX, all the file types specified in the `text conv` will also be converted to ASCII, thus making them unreadable. A possible solution to this would be to reconvert these files on the client side back to EBCDIC.

41.6.3 Porting SWAT

SWAT is run from inetd. That means that SWAT can output the HTML pages to its standard out and inetd will send them over the 'net. SWAT uses many printf's to output HTML code. Converting that output to ASCII cannot be done the usual way because the actual output is generated by the printf function. Translating its arguments to ASCII does not help either, since you can also have integers as parameters to printf.

We solved this problem by writing a wrapper function which is called on EBCDIC hosts instead of printf:

```
int mvs_printf(const char *format, ...) {
    int ret;
    FILE *bucket;
    char *buf;
    va_list args;

    bucket = fopen("/dev/null", "a");
    va_start(args, format);
    ret = vfprintf(bucket, format, args);
    va_end(args);

    buf = (char*) malloc(ret+1);
    va_start(args, format);
    vsprintf(buf, format, args);
    buf[ret]=0;
    __etoa_l(buf, ret);
    ret = printf("%s", buf);
    fflush(stdout);
    free(buf);
    return ret;
}
```

The next problem was converting the input. SWAT uses fgets(), which reads from a stream until it encounters EOF or a newline character. But the incoming stream is in ASCII, and an ASCII newline is different than one in EBCDIC.

We wrote a function that reads and converts the input stream byte by byte:

```
char *mvs_fgets (s, n, stream)
    char *s;
    int n;
    FILE *stream;
{
    register char *p = s;
    register int c = 0;
```

```

if (s == NULL || n <= 0)
{
    return NULL;
}

while (--n > 0 && (c = getc (stream)) != EOF) {
    *p = c;
    __atoc_l(p, 1);
    if (*p++ == '\\n')
        break;
}
if (c == EOF && p == s )
return NULL;
*p = '\\0';
return s;
}

```

The next problem was caused by escaped characters in the data sent by the browser. Certain characters (the slash, for example), must be escaped using the % HEX HEX encoding before sent to the HTTP server (SWAT, in our case). The character is sent as a string representing its ASCII value in hex, preceded by a percent sign. A slash would be sent as %2F, but a slash in EBCDIC is 0x61. After the % HEX HEX decoding, we had to convert that byte value from ASCII to EBCDIC.

The next issue was the authentication. The user name and password one enters in a browser are Base64 encoded by the browser. Base64 encoded data is represented as letters, digits, the plus sign, and the slash. SWAT decodes the string to get the original data, which is in ASCII. Hence, after the Base64 decoding, we had to convert the string to EBCDIC a second time.

SWAT still has the following problems:

- It cannot handle variables that occur more than once in smb.conf. This causes problems with the translate map parameter, which is used to specify what file types should be converted to or from ASCII.
- It does not display the smb.conf file in ASCII in the browser.

Chapter 42. Samba 1.9.18p1

This chapter discusses the Samba v1.9.18p1 open source software package. To read about the newer 2.2.1a version which is recommended, see Chapter 41, “Samba” on page 221.

42.1 Introduction to Samba

Samba is a Windows file server for UNIX. Samba is actually a suite of programs that implements the Server Message Block (SMB) protocol. See the redbook *S/390 File and Print Serving*, SG24-5330, for details of Samba for OS/390.

The port that is used is Samba version 1.9.18p1 from MKS.

42.2 Installation of Samba

Here is the core of the install script named `samba.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`). The Makefile is modified via `sed` commands to add the `-Wl,EDIT=NO` link directive. This results in much smaller executables. Also, after the executables are installed via `make install`, the code pages are removed (per the README.OS390 file).

```
pax -rzf samba-1.9.18p1.os390.tar.Z -o from=ISO8859-1,to=IBM-1047
ln -s samba-1.9.18p1 samba
cd samba/source
sed -e 's/FLAGS1 = -O/FLAGS1 = -O -Wl,EDIT=NO/g' \
    -e "s\BASEDIR = /usr/local/samba\BASEDIR = $installDir\g" Makefile \
    > Makefile.new
mv Makefile Makefile.orig
mv Makefile.new Makefile
make
make install
make uninstallcp
extattr +p $installDir/bin/smbd
```

Refer to Chapter 6 of *S/390 File and Print Serving*, SG24-5330, for the details of installing Samba.

42.3 Customization and operation of Samba

Again, refer to the Samba chapter in *S/390 File and Print Serving*, for the complete details of customizing Samba.

The important steps to remember are:

- Get proper authorization to smbd
- Define shares in the smb.conf file
- Get nmbd and smbd running
- Get clients configured properly (server name, password encryption)

42.4 License

The terms of use and redistribution of Samba are governed by the GPL. See Appendix A, “The GNU General Public License” on page 289 for the complete text.

In general, you are free to copy, modify and redistribute the package, but you must make the source available and you cannot put any restrictions on further redistribution.

42.5 Getting Samba from the Internet

The Samba package is available on the CD that accompanies the hardcopy of this redbook. The version of the package on the CD is 1.9.18p1. It was found on the MKS ftp server at:

<ftp://ftp.mks.com/pub/s390/gnu/>

42.6 Porting Samba to OS/390

MKS did the port of Samba to OS/390. The file README.os390 describes some of the aspects of the work that was done.

Chapter 43. MSNTP

This chapter discusses the MSNTP software package.

43.1 Introduction to MSNTP

MSNTP is the name of this package, which implements the SNTP protocol.

SNTP is a simple network time protocol (SNTP) utility. It was developed by Mr. N.M. Maclaren at the University of Cambridge. NTP is a protocol which provides globally consistent time stamps in a hostile environment. SNTP(RFC 2030-which supersedes RFC 1769) is a subset of NTP (RFC 1305) which gives excellent accuracy in most environments.

SNTP is also designed to operate in a dedicated server configuration including an integrated radio clock. With careful design and control of the various latencies in the system, it is possible to have accuracy to the order of microseconds.

43.2 Installation of sntp

Here is the core of the install script named msntp.install, which is included on the CD that accompanies the hardcopy of this redbook (where \$installDir is typically /usr/local)

```
gzip -c -d msntp-1.6.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s msntp-1.6 msntp
cd msntp
export CFLAG='-O2 -D_ALL_SOURCE'
CC=cc CFLAGS="$CFLAG" CCC=c++ CXXFLAGS="-+ $CFLAG"
make
```

43.3 Customization and operation of sntp

In order to test the sntp server, we used the following command.

```
$ ./sntp -S -v
```

It can also be started as a daemon via **inetd** and the inetd.conf configuration file. See the README file for details:

We downloaded and tested a shareware sntp client from:

<http://www.tucows.com>

It successfully connected to the server and adjusted time at specified intervals.

43.4 License

The following was found in the COPYRIGHT file of the package:

“(c) Copyright, N.M. Maclaren, 1996, 1997, 2000

(c) Copyright, University of Cambridge, 1996, 1997, 2000

Free use of MSNTP in source and binary forms is permitted, provided that this entire licence is duplicated in all copies, and that any documentation, announcements, and other materials related to use acknowledge that the software was developed by N.M. Maclaren (hereafter referred to as the Author) at the University of Cambridge. Neither the name of the Author nor the University of Cambridge may be used to endorse or promote products derived from this material without specific prior written permission.

The Author and the University of Cambridge retain the copyright and all other legal rights to the software and make it available non-exclusively. All users must ensure that the software in all its derivations carries a copyright notice in the form:

(c) Copyright N.M. Maclaren,

(c) Copyright University of Cambridge.

NO WARRANTY

Because the MSNTP software is licensed free of charge, the Author and the University of Cambridge provide absolutely no warranty, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the MSNTP software is with you. Should MSNTP prove defective, you assume the cost of all necessary servicing or repair. In no event, unless required by law, will the Author or the University of Cambridge, or any other party who may modify and redistribute this software as permitted in accordance with the provisions below, be liable for damages for any losses whatsoever, including but not limited to lost profits, lost monies, lost or corrupted data, or other special, incidental or consequential losses that may arise out of the use or inability to use the MSNTP software.

COPYING POLICY

Permission is hereby granted for copying and distribution of copies of the MSNTP source and binary files, and of any part thereof, subject to the following licence conditions:

1. You may distribute MSNTP or components of MSNTP, with or without additions developed by you or by others. No charge, other than an "at-cost" distribution fee, may be charged for copies, derivations, or distributions of this material without the express written consent of the copyright holders.
2. You may also distribute MSNTP along with any other product for sale, provided that the cost of the bundled package is the same regardless of whether MSNTP is included or not, and provided that those interested only in MSNTP must be notified that it is a product freely available from the University of Cambridge.
3. If you distribute MSNTP software or parts of MSNTP, with or without additions developed by you or others, then you must either make available the source to all portions of the MSNTP system (exclusive of any additions made by you or by others) upon request, or instead you may notify anyone requesting
4. You may not omit any of the copyright notices on either the source files, the executable files, or the documentation.
5. You may not omit transmission of this License agreement with whatever portions of MSNTP that are distributed.
6. Any users of this software must be notified that it is without warranty or guarantee of any nature, express or implied, nor is there any fitness for use represented.

43.5 Getting sntp from the Internet

The version of the package on the CD is available at:

<http://www.hpcf.cam.ac.uk/export/>

43.6 Porting MSNTP to OS/390

While building MSNTP, we uncommented and modified the following in the make file (as suggested in the README file):

```
# These options work on most modern systems. Start with them, and add
# any necessary options.
CC = cc -D_ALL_SOURCE
CFLAGS = -D_ALL_SOURCE
```

```
LDFLAGS =  
LIBS = -lm
```

Also, `settimeofday()` and `adjtime()` are not available in OS/390. Thus, as these functions are only used to test for error conditions, they were commented out (as also suggested in the README file).

Chapter 44. Squid

This chapter discusses the Squid open source software package. Due to legal concerns, however, this package cannot be distributed either with the redbook or on the z/OS Tools and Toys Web site.

44.1 Introduction to Squid

The following description was obtained from the Squid Web site:

Squid is a high-performance proxy caching server for web clients, supporting FTP, gopher, and HTTP data objects. Unlike traditional caching software, Squid handles all requests in a single, non-blocking, I/O-driven process.

Squid keeps meta data and especially hot objects cached in RAM, caches DNS lookups, supports non-blocking DNS lookups, and implements negative caching of failed requests.

Squid supports SSL, extensive access controls, and full request logging. By using the lightweight Internet Cache Protocol, Squid caches can be arranged in a hierarchy or mesh for additional bandwidth savings.

Squid consists of a main server program squid, some optional programs for rewriting requests and performing authentication, and some management and client tools.

Squid is derived from the ARPA-funded Harvest project.

44.2 Installation of Squid

To install Squid, you must unwind the tar file with gzip as the first step. Then it is recommended that you use the **ext** command to convert all the text files to EBCDIC and preserve the GIF icons in binary format.

Run the configure script with the following parameters

```
export _CC_CCMODE=1
./configure --prefix=$installDir --disable-wccp \
--disable-htcp --disable-unlinkd
```

Using **--disable-wccp** and **--disable-htcp** is optional, but they disable component not ported or not supported, so including them has no meaning. Use of **--disable-unlinkd** is mandatory, because the unlinkd support is broken in OS390. If you use it, the process never dies during a normal shutdown, or after a fork.

One additional parameter is `--disable-setuid`. If you are using a system with `BPX.DAEMON` defined (see Chapter 50, “Observations” on page 271) and you don’t have authorization to set the program control flag, this option disables all the `setuid/setgid` functions. In this case, Squid will run as your user ID, ignoring the `effective_user` setting.

To compile Squid, run the `make` command. To install it, run the `make install` command.

44.3 Customization and operation of Squid

Squid is a very complex piece of software, so customization is very dependent on the planned functionality. It can be used as a HTTP accelerator, as a proxy/cache, as a redirector from intranet servers, and even as a central access content control point.

Usually, a good starting point for customization information is the configuration file itself. All the options are well documented.

You cannot use Squid with the default configuration file. For security reasons, the HTTP access is disabled to all clients by default. To enable Squid to accept requests without any restriction, you can add the following line in your configuration file:

```
http_access allow all
```

As an alternative, you can enable only your network. Assuming your local network is 192.168.1.0/24, you can add:

```
acl mynet src 192.168.1.0/255.255.255.0
http_access allow mynet
http_access deny all
```

Further information can be found in the online Squid Configuration Manual at:

<http://squid.visolve.com/squidconf.html>

A block diagram of Squid is shown in Figure 4 on page 239.

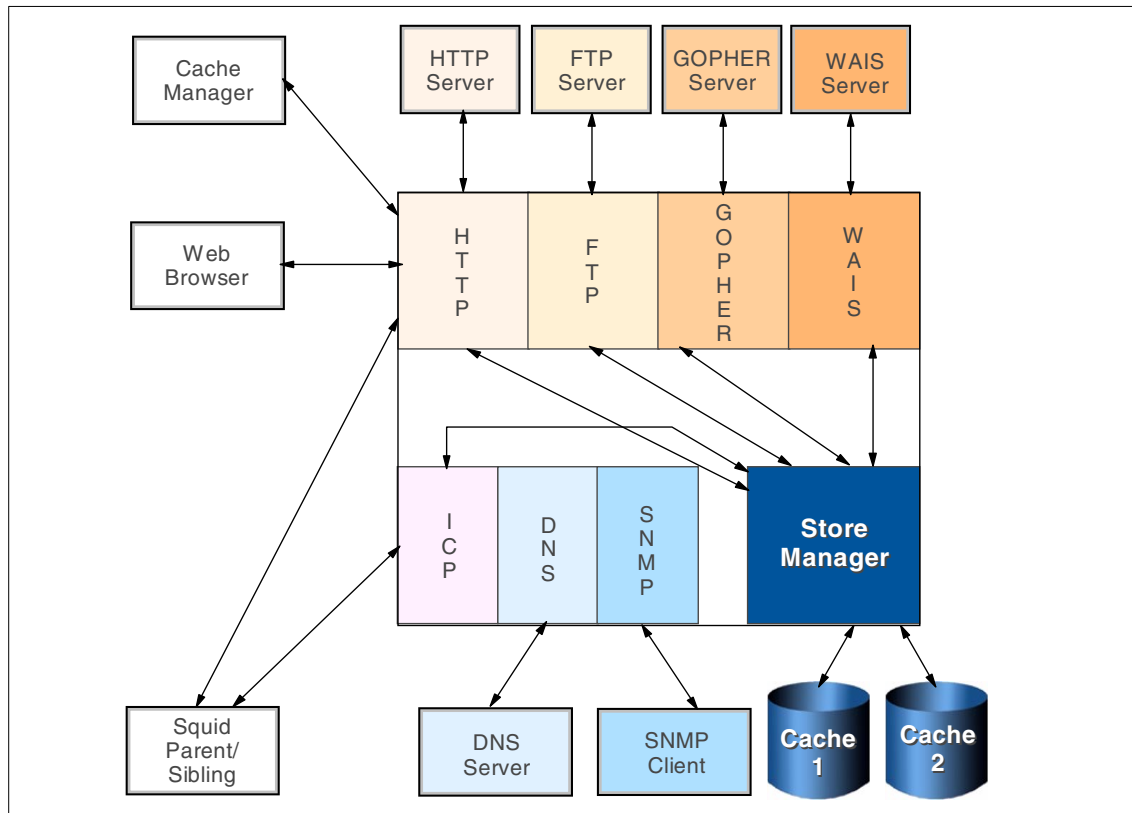


Figure 4. Squid block diagram

44.4 License

Squid is covered by the GNU GPL, which is explained in detail in Appendix A, “The GNU General Public License” on page 289.

44.5 Getting Squid from the Internet

The version of the package on the CD that accompanies the hardcopy of this redbook is 2.4.STABLE1 with all the patches available to May 7, 2001. The original package was found at:

<http://www.squid-cache.org/>

An OS/390 port was found at:

<http://linux390.unlp.edu.ar/squid/>

44.6 Porting Squid to OS/390

This application has two porting issues: code porting (undefined symbols, incorrect use of some structures, etc), and ASCII/EBCDIC conversions.

Code porting

In the case of code porting, these changes were made:

- There was a warning of a missing `sys/param.h` in `util.h`. The corrected code bypasses this include file, using the `_SQUID_SYS_PARAM_H` variable .
- In standard UNIX, the `strerror()` function has an array of strings in `sys_errlist` which is dimensioned to be `sys_nerr`. This is not the case with OS/390 UNIX, so the `sys_nerr` definition and comparisons in `lib/util.c` are not used if `__MVS__` symbol exists.
- The `configure.in` is patched for OS/390 support, adding `-DSTDC_HEADERS`, `DIS_EBCDIC` and `-D__rusage` as compilation flags. (The OS/390 `rusage` struct is incomplete, and is bypassed and replaced by the Squid `rusage` struct.)
- We added the option `--disable-setuid` in `configure.in`, because due to the “non always superuser” nature of UID 0 (its privileges are handled by RAC/F), the `setuid/setgid` from 0/0 to another user isn’t possible in some configurations. With this option, all the `setuid/setgid` functions are bypassed.
- Some file bits test bypassed if `__MVS__` symbol is defined. It generates assert fails.

ASCII/EBCDIC conversions

In the case of ASCII/EBCDIC conversions, two new files (`ebcdic.h/ebcdic.c`) are added. In all cases, the code is preprocessed in by the `IS_EBCDIC` symbol, so the modifications are not OS/390-specific. The conversion tables and functions used are the same as Apache 2. Conversion calls are made in the following cases:

- In the Cache Manager interface (`cachemgr.c`), in `read_reply()`, the `req->action` variable is converted from ASCII to EBCDIC. In the `process_request` function, the `but` variable is converted from EBCDIC to ASCII. Also, the `ebcdic.o` requirement to the `makefile` is added.
- In the client side front end (`client_side.c`), a ASCII/EBCDIC conversion of the incoming HTTP headers is made before processing, and in the HTTP protocol case, a reconversion to ASCII is made before the forwarding is done to the server.

- An ASCII/EBCDIC local conversion of the HTTP reply headers is made, before parsing them, in `http.c`.
- In the SSL CONNECT method (`ssl.c`), a couple of EBCDIC/ASCII conversions are inserted in `sslConnected` and `sslProxyConnected` functions.
- In the ICP protocol, URL conversion to ASCII in the `icpCreateMessage` function is issued. Also, an ASCII/EBCDIC conversion is provided in `icpHandleIcpV2`. The same conversion is ported to `icpHandleIcpV3` for Harvest 2 compatibility.
- In the FTP protocol, the ASCII/EBCDIC/ASCII conversion in the command **port** is done in `ftpWriteCommand()`, `ftpParseControlReply()` and `ftpParseListing()` and worked well.

There are still problems in the dynamic creation of HTTP headers/HTML pages to serve back to the browser, so the data port conversions were not fully tested. The “ASCII MODE” transfers apparently work, but due to header corruptions, we did not test the “BINARY MODE” transfers.

- In `access_log.c`, a mime headers conversion table is converted to EBCDIC special characters.
- In `errorpage.c`, an EBCDIC/ASCII error text conversion is added.

Summary

In summary, the working status of each protocol was as follows:

- HTTP: Fully working.
- SSL: Fully working.
- ICP: Fully working (only V2 tested).
- HTCP: Not ported yet, probably ported later. Use ICP.
- WCCP: Not ported, probably never ported (no OS support).
- DNS: Fully working.
- SNMP: Not ported yet.
- CACHE_OBJECT (Cache Manager protocol): Fully working locally.
- Store Manager: Working, but not caching files; header problems.
- FTP: Partially working, not usable yet; header problems.
- Gopher: Not ported, probably never ported.
- WAIS: Not ported, probably never ported.
- Error Pages: Partially working, header problems.

Further considerations

If someone were to continue work on this port, we suggest that the remaining work items, in priority order, would be the following:

- Code the ASCII/EBCDIC conversions in Store Manager and validate the cache header fields in disk and memory. This may solve the Store Manager inability to reuse cached files.
- Code the EBCDIC/ASCII conversions in HTTP Headers functions. This may solve the Error Pages, FTP protocol, and icon objects problems
- Solve the FTP problems.
- Solve the Error Pages problems.
- Solve the icon objects corruption problem.
- Solve the incompatibility between ASCII cachemgr.cgi and Squid.
- Port the SNMP protocol.
- Port the HTCP protocol.
- Port the Gopher and WAIS protocols.

Chapter 45. The Hessling Editor (THE)

This chapter discusses The Hessling Editor (THE) software package.

THE is a text editor that uses both command line commands and key bindings to operate. It is intended to be similar to the VM/CMS System Product Editor named XEDIT, and to KEDIT from Mansfield Software.

45.1 Introduction to THE

The default screen displayed when THE starts contains several areas:

- A window, which displays the contents of the file being edited. This is called the *file area*.
- A command line, from which commands may be issued.
- A prefix area, which shows line numbers and from which prefix commands can be issued.
- An ID line, which displays the file name, row, col and so on for the current file.
- A status line, which indicates global status information such as number of files being edited, time, and so on.

When THE starts, the cursor is positioned on the command line. To move between the command line and the file area, the CURSOR HOME command is used. By default this command is bound to the HOME key (under DOS and OS/2), the DO key (on vt220s), and HOME (on xterms).

To execute commands from the command line, simply type the command and press Enter (or the Return key).

To determine what keys are bound to what commands, execute the SHOWKEY command from the command line. As you press each key, THE will respond with the name of the key and any commands bound to that key. To exit from the SHOWKEY command, press the space bar.

Key bindings may be changed for the current session by using the DEFINE command. To keep key bindings between sessions, the DEFINE commands can be placed in a profile file, which is executed each time THE starts; for more information, refer to 45.3, "Customization and operation of THE" on page 244.

You can make THE look and behave more like XEDIT or KEDIT; the THE SET COMPAT command controls this function.

Like its relatives, THE uses REXX as its macro language, which means that THE is a highly configurable and versatile editor.

45.2 Installation of THE

Here is the core of the install script named fgres.install, which is included on the CD that accompanies the hardcopy of this redbook (where \$installDir is typically /usr/local):

```
gzip -c -d THE-3.0.os390.tar.gz> temp.tar
ext -v temp.tar
rm temp.tar
ln -s THE-3.0 the
cd the
export _CC_CCMODE=1
export RANLIB=echo
export LD=echo
CC=cc CFLAGS="-O2 -D_ALL_SOURCE -DIS_EBCDIC_HOST" CCC=c++ CCFLAGS="-+"
./configure --prefix=$installDir
echo "#define REGEX_MALLOC 1" >> config.h
gmake the
gmake install
cp therec.sample $installDir/THE/.therec
```

45.3 Customization and operation of THE

Various session defaults may be changed on startup for an individual by using a profile file. This file contains various commands that set the current environment, including key bindings. This profile file can also be used to process commands in batch mode.

The installation process copies a sample .therec profile into /usr/local/THE (or whatever directory you chose for installation). This is the system-wide profile for the editor. It is also possible to have an user-defined profile; copy this file into your home directory and change it as you prefer.

The usability of a Telnet-oriented editor depends mainly on the quality of the Telnet client and the server definitions for emulated terminals (terminfo).

During our test, we found it worked well with the putty Telnet client (available at the following site):

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

We used the following terminal definition:

```
$ export TERM=rxvt-color
```

The correct terminfo definition for this terminal was found at:

<http://www.tuxedo.org/terminfo>

We included this file (termtypes.ti) in our package, in the directory ./terminfo.

If you want to create all the terminal definitions contained in this file, use the following commands:

```
$ cd ./terminfo
$ ls -la
-rw-r--r--  1 TOT102  SYS1      802841 May 21 14:36 termtypes.ti
$ export TERMINFO=/usr/local/terminfo
```

Note: To avoid overwriting the existing definitions, write the path where you will install your new terminal definitions, instead of using the actual TERMINFO path (if you have one).

```
$ tic termttype.ti
```

After this command, you can use new definitions by putting in your .profile the new TERMINFO path and selecting the terminal type you want:

```
export TERMINFO=/u/tot102/terminfo
if[ -f /u/tot102/terminfo/r/rxvt ]; then
    export TERM=rxvt-color
fi
```

45.4 License

THE is covered by the GNU GPL, which is explained in detail in Appendix A, “The GNU General Public License” on page 289.

The Hessling Editor is Copyright © Mark Hessling, 1990-1999
<M.Hessling@qut.edu.au>

45.5 Getting THE from the Internet

The version of the package on the CD is 3.0. It was found on:

<http://www.lightlink.com/hessling/>

45.6 Porting THE to OS/390

We copied the following config.* files from autoconf to detect OS/390:

```
cp /usr/local/autoconf/autoconf/config.guess .
cp /usr/local/autoconf/autoconf/config.sub .
cp /usr/local/autoconf/autoconf/install-sh .
```

We ran **configure** as follows:

```
_CC_CCMODE=1 CC=cc CCC=c++ CCFLAGS="-+" CFLAGS="-O -DIS_EBCDIC_HOST" \
RANLIB=echo LD=echo ./configure
```

We commented the #define in getch.h at line 330 from:

```
#define KEY_DELETE 127
```

to:

```
/* #define KEY_DELETE 127 */
```

```
/* for OS/390 */  
#define KEY_DELETE 007
```

We added the following line at the end of the file config.h:

```
#define REGEX_MALLOC 1
```

Some functions did not work because of the lack of ncurses on OS/390. The current port uses standard curses. In particular, the editor screen area is limited to 80 x 24, regardless of the dimension of the terminal window.

When resizing windows after the editor is run, the editor area normally remains at 80 x 24, but then the content is visually corrupted.

Chapter 46. Vim

This chapter discusses the Vim open source software package.

46.1 Introduction to Vim

Vim is a professional text editor. It can be used to edit plain text files in a highly productive way. Although it is aimed at programmers, it can be used for many purposes.

Vim is a improved version of the vi editor shipped with every UNIX system today. Many features have been added to Vim, including:

- Multi-level undo
- Compile support (parsing compiler output and positioning the cursor)
- Syntax highlighting (around 190 different file types)
- Command line history
- Online help
- Filename completion
- Script language
- Optional GUI support

Vim has been ported to a lot of operating systems, and can be used on AmigaOS, AtariMiNT, BeOS, DOS, MacOS, OS/2, RISCOS, VMS, and Windows (95/98/NT4/NT5), on many flavors of UNIX (including AIX and Linux), and now finally on OS/390 UNIX.

For more information on Vim visit the Vim homepage:

<http://www.vim.org>

The main author of Vim is Bram Moolenaar (bram@vim.org). The port of version 5.6 to OS/390 was done by Ralf Schandl (schandl@de.ibm.com).

46.2 Installation of Vim

If you want to install the binary distribution from the CD, refer to Chapter 3, “What’s on the CD-ROM” on page 17.

If you want to compile it yourself, you need the source code from the CD or you can download it as explained in Chapter 46.5, “Getting Vim from the Internet” on page 257. You may find a newer version on the Internet.

Normally, vim is always built with debugging information included. It is highly recommend that you build Vim without this since it is much smaller then. See 50.1.2, “Configure and the debugging option” on page 272 on how to avoid this.

46.2.1 Version 5.6

46.2.1.1 Building without X11

If you build Vim without X11 support, compiling and building is nearly straightforward. The only restriction is that you can't call make from the Vim toplevel directory. To overcome this restriction, change to the src directory first and call make from there. The following log illustrates what we did:

Allow intermixing of compiler options and files:

```
$ export _CC_CMODE=1
$ ./configure --help
Usage: configure [options] [host]
Options: [defaults in brackets after descriptions]
...
    --prefix=PREFIX          install architecture-independent files in
PREFIX                        [/usr/local]
...
    --enable-max-features    Use maximum number of features (but not
*interp).
    --with-x                 use the X Window System
    --enable-gui [=opts]     X11 GUI [default=auto]
[opts=auto/no/gtk/motif/athena/
beos]
...
```

The help for the `configure` command shows you the different things you can include or exclude from Vim. We recommend using `--enable-max-features`, as you then get the most functionality out of Vim. Since `--enable-max-features` also enables the GUI support, we have to switch it off again using `--without-x` `--enable-gui=no`, as follows:

```
$ ./configure --enable-max-features --without-x --enable-gui=no
creating cache ./config.cache
checking whether make sets ${MAKE}... yes
checking for gcc... cc
checking whether the C compiler (cc -D_ALL_SOURCE ) works... yes
...
checking if character set is EBCDIC... yes
...
defaulting to: don't HAVE_X11
```

```

...
creating ./config.status
creating config.mk
creating config.h
$ cd src
$ make
CC="cc -Iproto -DHAVE_CONFIG_H -D_ALL_SOURCE " srcdir=. sh ./osdef.sh
cc -c -I. -Iproto -DHAVE_CONFIG_H -D_ALL_SOURCE buffer.c
...
cc -o vim buffer.o charset.o digraph.o edit.o eval.o ex_cmds.o
ex_docmd.o ex_getln.o fileio.o getchar.o if_cscope.o main.o mark.o
memfile.o memline.o menu.o message.o misc1.o misc2.o multbyte.o normal.o
ops.o option.o os_unix.o pathdef.o quickfix.o regexp.o screen.o search.o
syntax.o tag.o term.o ui.o undo.o window.o version.o -lcurses
link.sh: Linked fine, no libraries can be removed
cd ctags; SUFFIX="" CC="cc" CFLAGS="-D_ALL_SOURCE" \
    LDFLAGS="" \
    ./configure --mandir="/usr/local/man" --exec_prefix="/usr/local"
creating cache ./config.cache
Exuberant Ctags, version 3.4
OS/390 08.00 02 9672
...
cd ctags; CC="cc" CFLAGS="-D_ALL_SOURCE" \
    make -f Makefile SUFFIX=""
make[1]: Entering directory `/u/ralfs/src/vim-5.6-390c/src/ctags'
cc -I. -I. -DHAVE_CONFIG_H -D_ALL_SOURCE -c args.c
...
cc -o ctags args.o eiffel.o entry.o fortran.o get.o keyword.o main.o
options.o parse.o read.o sort.o strlist.o vstring.o
make[1]: Leaving directory `/u/ralfs/src/vim-5.6-390c/src/ctags'
cd xxd; CC="cc" CFLAGS="-D_ALL_SOURCE" make -f Makefile.unix
make[1]: Entering directory `/u/ralfs/src/vim-5.6-390c/src/xxd'
cc -D_ALL_SOURCE -DUNIX -o xxd xxd.c
make[1]: Leaving directory `/u/ralfs/src/vim-5.6-390c/src/xxd'

```

46.2.1.2 Building with X11

To build Vim with X11 support, you need at least version 5.6-390b.

There are two ways to build Vim with X11 support. Vim can be linked statically with the X11 libraries, which results in a very big executable (nearly 14 Mb), or it can be bound with the X11 DLLs, which shrinks the size to around 4.5 Mb.

1. To link statically, do the following:

```

$ export CC=cc
$ export CFLAGS="-D_ALL_SOURCE -W c,dll"

```

```

$ export LDFLAGS="-W l,dll"
$ ./configure --enable-max-features --enable-gui=motif
...
checking if character set is EBCDIC... yes
...
checking for X... libraries /usr/lib, headers
...
creating ./config.status
creating config.mk
creating config.h
$ cd src
$ make
CC="cc -Iproto -DHAVE_CONFIG_H -DUSE_GUI_MOTIF -I. -D_ALL_SOURCE -W
c,dll -D_ALL_SOURCE -W c,dll" srcdir=. sh ./osdef.sh
...
link.sh: Using link.sed file to remove a few libraries
cc -W l,dll -W l,dll -L/lib -L/usr/lib -o vim buffer.o charset.o
digraph.o edit.o eval.o ex_cmds.o ex_docmd.o ex_getln.o fileio.o
getchar.o if_cscope.o main.o mark.o memfile.o memline.o menu.o
message.o misc1.o misc2.o multbyte.o normal.o ops.o option.o os_unix.o
pathdef.o quickfix.o regexp.o screen.o search.o syntax.o tag.o term.o
ui.o undo.o window.o gui.o gui_motif.o gui_x11.o pty.o version.o -lXext
-lXm -lXt -lSM -lICE -lXt -lX11 -lcurses
link.sh: Linked fine with a few libraries removed
...
$ ls -l vim
-rwxr-xr-x 1 RALFS SYS1 13869056 Mar 1 13:30 vim*

```

Vim is now linked statically with the X11 libraries.

Don't be concerned about the error messages you receive during the link process. These messages occur because the script link.sh tries to find a minimal set of runtime libraries needed to link Vim. This is an iterative process and needs several link edit steps. This script is handy when we link Vim dynamically with X11.

2. To link dynamically, do the following:

Build Vim like above, then delete the executable. Now create a **sed** commandfile named link.sed, which removes several libraries and replaces others by definition side files.

Remove:

```

-lXext
-lXmu
-lXt

```

Replace:

```

-lXm      with /usr/lib/Xm.x
-lX11     with /usr/lib/X11.x
-lSM      with /usr/lib/SM.x
-lICE     with /usr/lib/ICE.x

```

The file should look like this:

```

s%-lXext  *%%g
s%-lXmu   *%%g
s%-lXt    *%%g
s%-lXm    *%/usr/lib/Xm.x %g
s%-lX11   *%/usr/lib/X11.x %g
s%-lSM    *%/usr/lib/SM.x %g
s%-lICE   *%/usr/lib/ICE.x %g

```

With this link.sed file the script link.sh will do the work for you:

```

$ rm vim
$ make
cc -c -Iproto -DHAVE_CONFIG_H -DUSE_GUI_MOTIF -I. -D_ALL_SOURCE -W
c,dll -D_ALL_SOURCE -W c,dll      version.c
link.sh: The file 'link.sed' exists, which is going to be used now.
link.sh: If linking fails, try deleting the link.sed file.
link.sh: If this fails too, try creating an empty link.sed file.
link.sh: Using link.sed file to remove a few libraries
      cc -W l,dll -W l,dll -L/lib -L/usr/lib -o vim buffer.o charset.o
digraph.o edit.o eval.o ex_cmds.o ex_docmd.o ex_getln.o fileio.o
getchar.o if_cscope.o main.o mark.o memfile.o memline.o menu.o
message.o misc1.o misc2.o multbyte.o normal.o ops.o option.o os_unix.o
pathdef.o quickfix.o regexp.o screen.o search.o syntax.o tag.o term.o
ui.o undo.o window.o gui.o gui_motif.o gui_x11.o pty.o      version.o
/usr/lib/Xm.x /usr/lib/SM.x /usr/lib/ICE.x /usr/lib/X11.x -lcurses
link.sh: Linked fine with a few libraries removed
$ ls -l vim
-rwxr-xr-x  1 RALFS  SYS1      4444160 Mar  1 13:40 vim*

```

Now Vim is linked with the X11-DLLs; because of this it is much smaller.

46.2.1.3 Test and install

After building, you can test the recently compiled Vim by calling `make test`.

```
$ make test
```

Test 28 will be reported as failed. This is because the standard OS/390 UNIX diff program can't compare files containing '\0' characters. Test 11 will fail if you don't have **gzip** installed.

Finally, you can install Vim by:

```
$ make install
```

This will install all needed files.

46.3 Customization and operation of Vim

If you are not familiar with `vi` or Vim, start the program **vimtutor** first. This is a tutorial on how to do basic editing tasks in Vim.

If you know `vi`, you can easily work with Vim. The most important feature for a new user is the online help. Use the help command like this:

```
:help tips
```

The help files contain links that are between two pipes (|). Position the cursor on one and hit <Ctrl-]> to follow the link; use <Ctrl-T> to return.

On the command line, you can use TAB for help item completion and <Ctrl-D> to show what items would match your current input. If you do:

```
:help move<Ctrl-D>
```

Vim shows you all help items containing the word “move”.

Table 1 shows several interesting help items.

Table 1. Help items

Help item	Description
help	How to use the help system.
os390	OS/390 UNIX related topics.
howto	General information of how to do things in Vim. Even if you are an experienced Vi user, you may find interesting links here.
tips	Tips for editing C files, HTML, and a lot more.
quickfix, make	How to start <code>make</code> from inside of Vim, so that Vim parses the error messages and you could easily navigate between them.
grep	How Vim can interface with <code>grep</code> or <code>grep</code> -like utilities.
syn-qstart	Quick start on syntax highlighting.
tagsearch	How to easily find identifier functions in your code.
expression	Expression evaluation and the internal script language.

Vim can be customized with a resource file, located in the user's home directory. The file is named `.vimrc`, and it is automatically read when Vim starts. A simple customization file may look like this:

```
" FILE: vimrc
"
" Define "del" char to be the same as backspace (saves a LOT of
trouble!)
if(has("ebcdic"))
    map <C-V>x07 <C-H>
else
    map <C-V>127 <C-H>
endif
"
" if in insert mode and pasting with the mouse (X11) hit F9 first to
" prevent unwanted indenting
if v:version >= 504
set pastetoggle=<F9>" toggle paste option
map <F9> :set invpaste paste?<CR>
imap <F9> <ESC>:set invpaste<CR>:echo ""<CR>$a
endif
"
let c_syntax_for_h = 1" c syntax vor *.h files, instead of C++
set background=dark" effects colors set with next command
syntax on" switch on syntax highlighting
"
set backspace=2" allows BS over the start point of insert
set scrolloff=4" keep at least 4 lines of text above/below cursor
set shortmess=atToO" a short message is OK
"
" startofline: no: do not jump to first character with page
" commands, ie keep the cursor in the current column.
set nostartofline
set autowrite" saves a lot of trouble
set backup" backup saved my life several times
set writebackup
set esckeys" use function keys in insert mode
set hidden" hide files if we are going to next one
set ruler" show cursor position
set showcmd" show typed command chars
set showmatch" when inserting bracket show matching one
set visualbell" no beep on errors but flash screen
set incsearch" incremental search is FUN!
"
" short date Example: 29Jun98
iab _Ydate <C-R>=strftime("%d%b%y")<CR>
"
```

```
" long date Example: Fri 16 April 1999 21:55:53 CEST
iab _YDate <C-R>=strftime("%a %d %B %Y %T %Z")<CR>
"
" put parentheses around "visual text"
vmap ,) v`>a)<ESC>`<i(<ESC>
```

It is a little bit confusing that the double quote is used as a comment header and is also used to enclose strings.

For an example of another customization file (.vimrc), have a look at:

<http://www.vim.org/rc>.

46.4 Licence

Vim isn't released under the GPL; rather, it uses a unique licence. Here is a quote from Vim's online help (:help copyright):

SUMMARY

Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation to orphans in Uganda. See |iccf|¹ below. If you include Vim on a CD-ROM, I would like to receive a copy. Just so I know which Vim distributions exists in the world (and to show off to my friends :-)).

DETAILS

There are no restrictions on distributing an unmodified copy of Vim. Parts of Vim may also be distributed, but this text must always be included. You are allowed to include executables that you made from the unmodified Vim sources, your own usage examples and Vim scripts.

If you distribute a modified version of Vim, you are encouraged to send the maintainer a copy, including the source code. Or make it available to the maintainer through ftp; let him know where it can be found. If the number of changes is small (e.g., a modified Makefile) e-mailing the diffs will do. When the maintainer asks for it (in any way) you must make your changes, including source code, available to him.

The maintainer reserves the right to include any changes in the official version of Vim. This is negotiable. You are not allowed to distribute a modified version of Vim when you are not willing to make the source code available to the maintainer.

The current maintainer is Bram Moolenaar <Bram@vim.org>. If this changes, it will be announced in appropriate places (most likely www.vim.org and comp.editors). When it is completely impossible to contact the maintainer, the obligation to send him modified source code is dropped.

It is not allowed to remove these restrictions from the distribution of the Vim sources or parts of it. These restrictions may also be used for previous Vim releases instead of the text that was included with it.

¹ In the online help system, you can follow this link to read more about iccf.

46.5 Getting Vim from the Internet

The Vim package is available on the CD that accompanies the hardcopy of this redbook. The package is a port of version 6.0. A well-mirrored download page was found on the Web at:

<http://www.vim.org/download.html>

On the ftp servers, you will find the OS/390 UNIX package in the directory `unreleased/os390`, as it is not part of the main distribution yet.

If you want to compile Vim by yourself, you need these files:

- `vim-6.0-rt2.tar.gz`
- `vim-6.0-src2.tar.gz`
- `vim-6.0-rt1.tar.gz`
- `vim-6.0-src1.tar.gz`

For convenience, these four files were unwound, patch6.0.101 was applied, and they were incorporated into a single tar file, `vim-6.0-src.tar.gz`. This was also necessary for vim to fit into the automated build script, `5944-01.install`, which expects packages in a single file.

46.6 Porting Vim to OS/390

As expected, the main problem porting Vim was the heavy ASCII dependency in the code. To detect that the current machine uses EBCDIC, we added a test to the configure script as described in 50.1.4, “Setting the DISPLAY variable for X Window applications” on page 273. With this change, the macro named `EBCDIC` is defined in `config.h` and can be used for `#ifdef`’s in the code.

A problem was encountered with the way control characters are calculated:

```
#define Ctrl(x) ((x) & 0x1f)
```

For a constant `x` this evaluates to a constant expression during compile time, and can therefore be used in case labels. In EBCDIC there is no way to calculate the control character from a given character, so a lookup table was needed:

```
#define Ctrl(x) (CtrlTable[x])
```

Since this doesn’t evaluate to a constant expression, it can’t be used in case labels. For this we had to define `Ctrl_A`, `Ctrl_B`, and so on. These constants are also defined for ASCII, to minimize `#ifdef`’s in the code.

We also added lookup tables to look up meta characters and to translate a control character back to the corresponding character:

```
#define CtrlChar(x) ((x<' ')?CtrlCharTable[x]:0)
#define MetaChar(x) ((x<' ')?MetaCharTable[(x)]:0)
```

For example:

```
Ctrl('A')      == 0x01
CtrlChar(0x01) == 'A'
```

To calculate the position of a character in the alphabet we added the following macro:

```
#ifdef EBCDIC
#define CharOrd__(c) ((c)<('j'-'a')?(c):((c)<('s'-'a')?(c)-7:(c)-7-8))
#define CharOrd(x) (isupper(x)?CharOrd__((x-'A')):CharOrd__((x-'a'))))
#else
# define CharOrd(x) (isupper(x)?(x)-'A':(x)-'a')
#endif
```

Printable and non-printable characters

In EBCDIC, all characters above and including space (code 64) are printable for Vim (they can be displayed as one character on the screen).

Digraphs

A new digraph table has been added for EBCDIC. This table is codepage IBM-1047 dependent. Use the command `:digraphs` to see the actual characters.

Hardcoded character values

At a lot of places hardcoded character values were found in strings. Like:

```
char *str = "\026x"
```

This are the chars Ctrl-V and 'x'. We replaced this kind of strings with:

```
/* include file */
#ifdef EBCDIC
# define CTRL_V_STR "\062"
#else
# define CTRL_V_STR "\026"
#endif

/* source file */
char *str = CTRL_V_STR "x"
```

Remember that consecutive strings in the code are concatenated by the compiler. (The usage of compile-time string concatenation will be removed if

the OS/390 UNIX port is merged with the mainstream. Some older compilers may not support this.)

The same procedure was used for the often-used character ESCAPE.

Eval

For the internal script language we added the feature “ebcdic”. With it you can fix ASCII-dependent scripts like this:

```
if(has("ebcdic"))
    let space = 64
else
    let space = 32
endif
```

Tag search

Vim has the ability to do a binary search on a sorted tag file. As `/bin/sort` on OS/390 UNIX simulates an ASCII sort program, the lines are *not* sorted by the EBCDIC character values, but by the ASCII character values. With this kind of sorting the binary search doesn't work.

For this reason, binary search is turned off if EBCDIC is detected. The tag files are only searched linear, which is a bit slower.

Options

The default values for several options had to be changed for EBCDIC.

The format for c89 error messages has been added to the `errorformat` option.

Regular expressions

Since the ranges a-z and A-Z in EBCDIC contain non-alphabetic characters, the code for these ranges was fixed to ignore these non-alphabetic characters.

Tests

The following tests had to be modified because they contained ASCII character values:

- Test2
- Test14
- Test17
- Test26

Chapter 47. Webalizer

This chapter discusses the Webalizer software package. According to the Webalizer Web site, it is a fast, free Web server log file analysis program. It produces highly detailed, easily configurable usage reports in HTML format, for viewing with a standard Web browser.

47.1 Introduction to Webalizer

Webalizer is written in C to be extremely fast and highly portable. It supports standard Common Log file Format server logs. In addition, several variations of the Combined Log file Format are supported, allowing statistics to be generated for referring sites and browser types, as well.

Generated reports can be configured from the command line, or by use of one or more configuration files.

Webalizer supports multiple languages including Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Galician, Hungarian, Icelandic, Indonesian, Italian, Korean, Latvian, Malay, Norwegian, Polish, Portuguese, Romanian, Serbian, Slovak, Slovene, Spanish, Swedish and Turkish.

Unlimited log file sizes and partial logs are supported, allowing logs to be rotated as often as needed, and eliminating the need to keep huge monthly files on the system.

47.2 Installation of Webalizer

Webalizer requires the GD graphics library by Tom Boutell. To install GD, the following packages are required:

- Zlib
- LibPNG
- Jpeg-6b (optional, but recommended)

The Webalizer package is available on the CD that accompanies the hardcopy of this redbook. The prerequisite libraries and tools are also available there; ensure you have them installed before you install Webalizer.

The version of Webalizer provided on the CD is 2.01-06.

Here is the core of the install script named `webalizer.install`, which is included on the CD that accompanies the hardcopy of this redbook.

```
gzip -c -d webalizer.os390.2.01-06.tar.gz | \
  pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s webalizer-2.01-06 webalizer
cd webalizer
chmod u+w *
export _CC_CCMODE=1
CC=cc ./configure --prefix=$installDir \
  --with-z-inc=/usr/local/include \
  --with-zlib=/usr/local/lib \
  --with-gd=/usr/local/include \
  --with-gdlib=/usr/local/lib
make
make install
```

47.3 Customization and operation of Webalizer

Generated reports can be configured either from the command line, or by use of one or more configuration files. See the README file for a complete description of Webalizer configuration files and the command line options.

To test Webalizer, copy the file `sample.conf` into a working directory, rename it `webalizer.conf`, edit it to specify (at a minimum) where the log file is, and then invoke:

```
$ webalizer
```

It should create the HTML report and associated images in this working directory.

Due to errors in GD porting, we were unable to create pie chart graphics, so we defined the following statement in our `webalizer.config` file to suppress the generation of pie charts:

```
CountryGraph    no
```

Figure 5 on page 263, however, contains a sample chart showing the Webalizer report for a month.

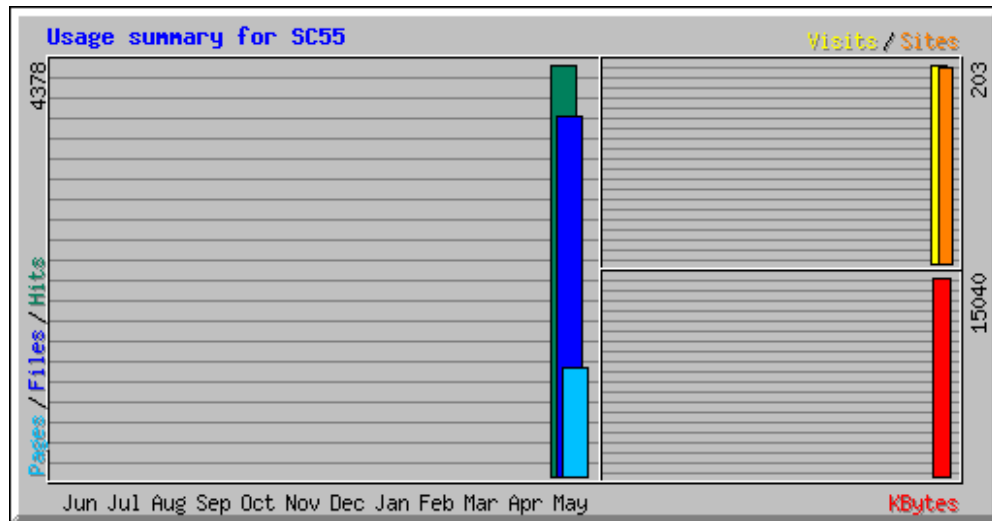


Figure 5. Webalizer monthly report

47.4 License

Webalizer is covered by the GNU GPL, which is explained in detail in Appendix A, "The GNU General Public License" on page 289.

47.5 Getting Webalizer from the Internet

Webalizer can be found on the Internet starting at:

<http://www.mrunix.net/webalizer/>

47.6 Porting Webalizer to OS/390

Webalizer built relatively cleanly from the Internet. We downloaded the package and converted it to EBCDIC as follows:

```
$ export A2E="-o from=ISO8859-1,to=IBM-1047"
$ pax -rf webalizer-2.01-06-src.tar $A2E
$ cd webalizer-2.01-06
```

We edited the configure script, around line 2068, and changed the following line from:

```
DEFS=`sed -f conftest.defs confdefs.h | tr '\012' ' '`
```

to:

```
DEFS=`sed -f conftest.defs confdefs.h | tr '\n' ' '`
```

We then created the makefile and built it as follows:

```
$ export _CC_CMODE=1
$ CC=cc CFLAGS="-O2 -DIS_EBCDIC_HOST" CCC=c++ CCFLAGS="-+" \
> ./configure --with-zlib=/usr/local/lib
> --with-z-inc=/usr/local/include \
> --with-gd=/usr/local/include \
> --with-gdlib=/usr/local/lib
$ gmake
$ su
$ gmake install
```


Chapter 48. Xterm client

This chapter discusses the xterm client open source software package.

48.1 Introduction to xterm client

The xterm program is a terminal emulator for the X Window system. It allows you to start multiple terminal sessions from other sessions and display them to an X server running on your desktop. It differs from telnet or rlogin in that authentication is not performed. Rather than starting from a client program on your desktop and authenticating with the target server, you must first have a terminal session on the target machine where you can start an xterm client session and display back to an X server.

Greg Smith of Lockheed Martin originally ported xterm to OS/390. He sent these changes to Thomas E. Dickey who maintains an xterm client Web site.

A good xterm FAQ page is on the Web at:

<http://dickey.his.com/xterm/xterm.faq.html>

48.2 Installation of xterm client

Here is the core of the install script named xterm.install, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d xterm.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s xterm-128 xterm
cd xterm
export LIBS='/usr/lib/Xaw.x /usr/lib/SM.x /usr/lib/ICE.x \
/usr/lib/X11.x'
export CFLAGS='-O2 -W c,dll -D_ALL_SOURCE'
export LDFLAGS=-Wl,EDIT=NO
./configure --prefix=$installDir
#remove X libraries from Makefile"
cat Makefile | sed 's/-lXaw -lXmu -lXext -lXt -lSM -lICE \ -lX11//g'
> Makefile.new
mv Makefile.new Makefile
make xterm
cp xterm $installDir/bin/xterm
cp xterm.man $installDir/man/$LANG/cat1/xterm.1
```

48.3 Customization and operation of xterm client

Before xterm can be run, an X server has to be running on your PC. We have Hummingbird's Exceed installed, which is a priced product. There is also a "try and buy" X server available from MicroImages at:

<http://www.microimages.com/freestuf/>

Click on **MI/X 2.0 for Windows**.

Once the X server is running, xterm is brought up by first setting the DISPLAY environment variable as follows:

```
$ export DISPLAY=9.12.0.115:0
$ ./xterm &
```

You can choose different foreground and background colors on the command line with the `fg` and `bg` parameters:

```
$ ./xterm -bg black -fg yellow &
```

48.4 License

The terms of use and redistribution of the xterm client are governed by the copyright. It was found as comments in the source files as follows:

```
Copyright (c) 1987, 1988  X Consortium
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

```
Except as contained in this notice, the name of the X Consortium shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from the X Consortium.
```

```
Copyright 1987, 1988 by Digital Equipment Corporation, Maynard.
```

```
All Rights Reserved
```

```
Permission to use, copy, modify, and distribute this software and its
```

documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

48.5 Getting the xterm client from the Internet

The xterm client package is available on the CD that accompanies the hard copy of this redbook. The version of the package on the CD is 165. The package was found at:

<http://dickey.his.com/xterm/xterm.html>

From there, click **The source (gzip'd tar)** at the bottom of the window to get to the package at:

<ftp://dickey.his.com/xterm/xterm.tar.gz>

The package was uncompressed and unwound using `gunzip` and `pax`:

```
$ ls -l
-rw-r--r--  1 MIKEM   SYS1      603298 Jan 18 12:06 xterm.tar.gz
$ gzip -c -d xterm.tar.gz | pax $A2E -r
$ ls -l
drwxr-xr-x  5 AAAAAAA SYS1      8192 Jan  5 20:42 xterm-165
-rw-r--r--  1 MIKEM   SYS1      603298 Jan 18 12:06 xterm.tar.gz
```

48.6 Porting xterm client to OS/390

Xterm was ported to OS/390 by Greg Smith of Lockheed Martin. No notes were found describing the changes made.

As is suggested in the instructions in `README.os390`, it is beneficial to modify the Makefile and remove the libraries `'-lXaw -lXext -lXmu -lXt -lSM -lICE -lX11'` from the LIBS assignment.

Removing the statically linked libraries in favor of dynamically linked libraries (dlls) resulted in an xterm executable that is about one fifth the size

Chapter 49. Zlib library

This chapter discusses the zlib library open source software package.

49.1 Introduction to zlib library

Note: Just before this redbook was published, a security exposure was found with zlib v1.1.3, so it was replaced with v1.1.4, which fixes that exposure. However, the binaries that were built on OS/390 V2 R9 could not be replaced, so using the source code is recommended.

zlib 1.1.4 is a general purpose data compression library. All the code is thread safe.

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 which describe zlib, deflate, and gzip formats. They can be found on the Web at:

<ftp://ds.internic.net/rfc/rfc1950.txt>
<ftp://ds.internic.net/rfc/rfc1951.txt>
<ftp://ds.internic.net/rfc/rfc1952.txt>

These documents are also available in other formats from:

<ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html>

49.2 Installation of zlib library

The tool **gzip** is needed to install the zlib library.

Here is the core of the install script named `zlib.install`, which is included on the CD that accompanies the hardcopy of this redbook (where `$installDir` is typically `/usr/local`):

```
gzip -c -d zlib-1.1.4.tar.gz | pax -o from=ISO8859-1,to=IBM-1047 -r
ln -s zlib-1.1.4 zlib
cd zlib
chmod u+w *
./configure --prefix=$installDir
make
make install
```

49.3 License

(C) 1995-1998 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly

Mark Adler

jloup@gzip.org

madler@alumni.caltech.edu

49.4 Getting zlib library from the Internet

The zlib library packages are available on the CD that accompanies the hardcopy of this redbook. The version of the zlib package on the CD is 1.1.4.

You can download zlib from the Internet starting at:

<http://www.gzip.org/zlib/>

49.5 Porting the zlib library

The zlib library should compile cleanly on OS/390 UNIX.

Chapter 50. Observations

This chapter presents overall observations we made during this project. Some are general hints and tips, while others are common problems that were encountered. We also briefly mention gcc, a package that was not fully explored or documented in this redbook, but which was considered.

50.1 Hints and tips

Following are hints and tips that may be helpful when working with open source software packages on OS/390.

50.1.1 Reducing executable size

Sometimes an executable builds cleanly, but it is much larger (more bytes) than on other platforms. There are two important areas where this can be remedied:

- Use dynamic link libraries (DLLs).

A DLL is a collection of one or more functions or variables in an executable module that is executable or accessible from a separate application module. In an application without DLLs, all external function and variable references are resolved statically at bind time. In a DLL application, external function and variable references are resolved dynamically at run time.

- Use the link flag `-Wl,EDIT=NO`.

This option prevents an executable from being reprocessed by the binder or linkage editor. This removes the symbol table from the executable, which prevents you from doing CSECT-level replacement using the current executable as input to the binder.

Without this flag, the debug format is entirely within the memory image of the program, both as data (partly within additional CSECTs) and additional instructions intertwined with the user program. This is quite different from other platforms, where the information is entirely off to the side, and easily removed (note that the `strip` command on OS/390 does not do this after compile time, as it does on other UNIXes).

For example, xterm will build if the instructions are followed, but the makefile created by configure does not use either of these techniques.

The following file sizes were observed on an OS/390 V2R7 system. (Note that xterm can be less than 7% of its original size.)

xterm with no DLLs, no link flag	6823936
xterm with DLLs, no link flag	1257472
xterm with DLLs, -w1,EDIT=NO flag	450560

Reducing the size of executables can save both time (faster load times) and disk space.

50.1.2 Configure and the debugging option

All configure scripts we saw set the debugging option for the compiler (-g). Because of this, a much bigger executable is built from the source code, since the compiler includes additional debugging information. On OS/390 UNIX, unlike other UNIX systems, this debugging information cannot be removed afterward. Also, the command `strip` exists and runs successfully; however, it does not change the file size.

If you do not want to debug the application, you don't need this debug information. There is a simple way to prevent configure from setting this option.

If the environment variable `CFLAGS` exists, configure doesn't add the option -g to it. For some tools, you need to define the environment variable anyway, so this problem is solved. For other tools, define `CFLAGS` to a string containing a space character, like this:

```
export CFLAGS=" "
```

50.1.3 Setting aliases for missing commands

Some commands are present on other UNIX operating systems which are not present on OS/390 UNIX. The commands `which` and `clear` can be resolved with the `alias` command:

```
alias which=whence
if [ X"$TERM" = "X" -o X"$TERM" = "Xdumb" ]
then
    alias clear="printf '\f'"
else
    alias clear='tput clear'
fi
```

For OS/390 V2R7 and earlier, common TCP/IP commands were preceded with the letter `o`. These, too, can be added via the `alias` command, as shown:

```
alias netstat=onetstat
```



```
alias ping=oping
alias rexec=orexec
alias rpcinfo=orpcinfo
alias snmp=osnmp
alias traceroute=otracert
alias rpcgen=orpcgen
```

For OS/390 V2R8 and later, these commands are named properly.

50.1.4 Setting the DISPLAY variable for X Window applications

To help your X Window application find the X Window server on your workstation, you have to set an environment variable. You can do that with the following command:

```
$ export DISPLAY=9.12.2.104:0
```

where 9.12.2.104 is your IP address and :0 is a fixed suffix for the screen number.

You'll find it quite cumbersome always to check your IP address and set a variable before you can use your application. For improved efficiency, we suggest putting the following code into either your personal `.profile` or into the general `/etc/profile`.

```
#
# Set DISPLAY Variable for X Window
#
# find out if I have used telnet or rlogin to get into USS.
# if so, parse out the arguments sent to it which contain my IP address
set . $(ps -o args -p $PPID)
case "$3" in
*telnet*) DISPLAY="$5";;
*rlogin*) DISPLAY="$8";;
*)      DISPLAY="None";;
esac
#
# set the DISPLAY environment variable
export DISPLAY="$DISPLAY:0"
#
```

This will set the DISPLAY variable automatically every time you log on.

50.1.5 Login from Linux

We logged on to OS/390 UNIX from Linux workstations, using `xterm` or `rxvt` as our terminal. OS/390 UNIX does not know `rxvt`, and even with `xterm` there may be differences.

You can ask your system administrator to add the rxvt entry to the system's terminfo database. We do not recommend replacing the system-provided xterm entries; instead, pass the files created as described in step 3 to your system administrator.

It is also possible to do this configuration for your use only, as follows:

1. Create a directory in your home directory on OS/390 UNIX to hold terminfo data. We used \$HOME/local/terminfo.
2. Export the TERMINFO environment variable, so it points to this directory. Also add this export statement to your .profile.

The system will now first look into your local terminfo directory before checking the system terminfo directory.

3. On Linux, extract the terminfo entries for xterm and rxvt. Do the following:

```
infocmp xterm > xterm.ti
infocmp rxvt > rxvt.ti
```

For your convenience we also include these files on the CD, but it is always better to create them on your system.

4. Use ftp to transfer the files to OS/390 UNIX. You must transfer them in text mode so that they are translated to EBCDIC.
5. Compile the xterm and rxvt terminfo entries, using the program tic like this:

```
tic xterm.ti
tic rxvt.ti
```

The tic program stores the compiled entries in subdirectories of the directory pointed to by the environment variable TERMINFOI.

50.1.6 Downloading an X Window server from the Internet

There are a number of different X Window servers available. One of the most popular is Exceed, from Hummingbird Software. You can download a trial version of MI/X written by Microimages from the Internet at:

<http://www.microimages.com/freestuf/>

This downloaded version will only work for a limited time, and it does not provide the full functionality.

50.1.7 Testing for EBCDIC and OS/390 in configure

You can put the following into the configure.in or aclocal.m4 files. The first test was initially written by Paul Gilmartin for the port of Lynx to OS/390 UNIX.

```

AC_MSG_CHECKING(if character set is EBCDIC)
AC_TRY_COMPILE([ ], [ /* TryCompile function for CharSet.
Treat any failure as ASCII for compatibility with existing art.
Use compile-time rather than run-time tests for cross-compiler
tolerance. */
#if '0'!=240
make an error "Character set is not EBCDIC"
#endif ],
[ # TryCompile action if true
cf_cv_ebcdic=yes ],
[ # TryCompile action if false
cf_cv_ebcdic=no])
# end of TryCompile ])
# end of CacheVal CvEbcDic
AC_MSG_RESULT($cf_cv_ebcdic)
if test "$cf_cv_ebcdic" = "yes"; then
    AC_DEFINE(IS_EBCDIC_HOST)
esac

```

When you run configure, and an EBCDIC target system is detected, configure adds the following line to the created file config.h:

```
#define EBCDIC
```

For a non-EBCDIC target system, it adds the following line to config.h:

```
/* #undef EBCDIC */
```

The following is a way your configure script can detect the OS/390 platform and set some environment variables for the make process:

```

dnl
dnl
dnl Check for OS/390 Unix. We test for EBCDIC also -- the Linux port
dnl (with gcc) uses ASCII internally.
dnl
AC_DEFUN(OS390,
[cf_cv_os390="no"
if test "$cf_cv_ebcdic" = "yes"; then
    AC_MSG_CHECKING([for OS/390 Unix])
    case `uname` in
        OS/390)
            CFLAGS="$CFLAGS -D_ALL_SOURCE"
            cf_cv_os390="yes"
            AC_MSG_RESULT(yes) ;;
        *)
            AC_MSG_RESULT(no) ;;
    esac
fi])dnl

```

50.2 Problems encountered

Following are problems that may be commonly encountered when working with open source software packages on OS/390.

50.2.1 Configure problems

If a problem occurs, first check the INSTALL file. See the discussion in 4.3, “Configuring and compiling” on page 28, for an overview of the configure script structure.

50.2.2 Using the wrong compiler

Though there is actually only one OS/390 C/C++ compiler, there are three common ways to invoke it:

Via the command `cc` to invoke the pre-ANSI C compiler

Via the command `c89` to invoke the ANSI C compiler

Via the command `c++` to invoke the C++ compiler

Sometimes the compiler `cc` must be invoked. On the other hand, sometimes the compiler `c89` must be invoked. For example, when `cc` was invoked to compile GNU patch, the following error was encountered

```
...
cc -c -DHAVE_CONFIG_H -Ded_PROGRAM=\"/bin/ed\" -I. -I. -g util.c
ERROR CBC3343 ./util.c:489 Redclaration of fatal differs from
previous declaration on lin15 of \"./util.h\".
ERROR CBC3045 ./util.c:493 Undeclared identifier va_alist.
...
```

50.2.3 Order of compiler command line parameters

Sometimes the compiler `cc` must be invoked. For example, a call to the xterm makefile with `c89` resulted in:

```
c89 -O2 -W c,dll -D_ALL_SOURCE -o xterm button.o charproc.o charsets.o
cursor.o data.o doublechr.o fontutils.o input.o main.o menu.o misc.o
print.o ptydata.o screen.o scrollbar.o tabs.o util.o VTPrsTbl.o
TekPrsTbl.o Tekproc.o -L/usr/lib -lXaw -lXmu -lXext -lXt -lSM -lICE
-lX11
FSUM3008 Specify a file with the correct suffix (.c, .i, .s, .a, .o, or
.x), or a corresponding data set name, instead of -L/usr/lib.
FSUM8226 make: Error code 1
```

This is because the OS/390 C/C++ compiler is enforcing ANSI rules that other compilers don't enforce. The workaround is to define the environment variable `_CXX_CCMode`.

Putting the following line in your `.profile` will help:

```
export _CXX_CCMode=1
```

50.2.4 Static constructor test fails

When the configure script is run, it may fail with a notice that static constructors and destructors don't work:

```
$ CC=cc CFLAGS="-O2 -D_ALL_SOURCE" CCC=c++ CCFLAGS="-+" ./configure
checking for gcc... cc
checking whether the C compiler (cc -O2 ) works... yes
checking whether the C compiler (cc -O2 ) is a cross-compiler... no
checking whether we are using GNU C... no
checking whether cc accepts -g... yes
checking for c++... c++
checking whether the C++ compiler (c++ ) works... yes
checking whether the C++ compiler (c++ ) is a cross-compiler... no
checking whether we are using GNU C++... no
checking whether c++ accepts -g... no
checking that C++ compiler can compile simple program... yes
checking that C++ static constructors and destructors are called... no
configure: error: a working C++ compiler is required
```

The message is not correct—static constructors and destructors *do* actually work properly. This is a bug in the OS/390 Language Environment; APAR PQ35739 was accepted to remedy it. At the time of writing, applying PTF UQ42006 was supposed to fix this, but the test program is still returning the wrong value (1). To work around this problem, you can comment out the following in the configure script (near line 957). This will effectively bypass the test of the static constructors and destructors.

```
#if { (eval echo configure:957: '$ac_link') 1>&5; (eval $ac_link) 2>&5;
} && test -s conftest${ac_exeext} && (./conftest; exit) 2>/dev/null
#then
  echo "$ac_t""yes" 1>&6
#else
# echo "configure: failed program was:" >&5
# cat conftest.$ac_ext >&5
# rm -fr conftest*
# echo "$ac_t""no" 1>&6; { echo "configure: error: a working C++
compiler is required" 1>&2; exit 1; }
#fi
```

50.2.5 Newline denoted by a constant

Another common problem is when a newline character is represented by a constant instead of the ‘\n’ character. This was encountered in a configure script:

```
...creating cache ./config.cache
creating ./config.status
creating Makefile
sed: conftest.s1 (line 9): FSUM7296 newline or end of file found in
pattern
```

Looking into that part of the script will show an ASCII/EBCDIC issue. Rather than newline being denoted by an octal constant, it should be denoted by the escaped ‘\n’:

```
DEFS=`sed -f conftest.defs confdefs.h | tr '\012' ' '`
```

Correct this line to the following, and the script will generate the makefile:

```
DEFS=`sed -f conftest.defs confdefs.h | tr '\n' ' '`
```

50.2.6 Compilation warning

You may receive the following compilation warning:

```
<sys/param.h> not found and MAXHOSTNAMELEN or MAXPATHLEN unresolved.
```

Usually if you get a compilation warning complaining that `sys/param.h` is not found, you can ignore it. But if `MAXPATHLEN` or `MAXHOSTNAMELEN` are unresolved, it is likely that you need to define those symbols in your code. Replace the “`#include <sys/param.h>`” in your code by this code:

```
#ifdef __MVS__
# define MAXHOSTNAMELEN 64
# define MAXPATHLEN 1024
#else
# include <sys/param.h>
#endif
```

In some cases, to define the macro `_ALL_SOURCE` helps to get rid of the undefined symbol.

You may also receive the following warning:

```
sys_nerr and sys_errlist unresolved: use strerror() function
```

Usually this happens when someone is trying to get at error messages, such as those printed by `perror()`, without using the defined interface `strerror()`. Common UNIX implementations of `strerror()` have an array of strings in

sys_errlist which is dimensioned to be sys_nerr, so code like the following can be used:

```
if(errno < sys_nerr)msg = sys_errlist[errno] ;
    else msg = "Unknown system error";
```

In OS/390, such code must be replaced by:

```
msg = strerror(errno);
```

Portable code correctly configured via autoconf/configure should not have this problem.

50.2.7 Linker does not find libraries

An error was encountered where the linker could not find a library:

```
SANDORB:/u/sandorb/mc-3.2.1: >make
cd ../slang; make libmcslang.a
`libmcslang.a' is up to date
cc -o mc dir.o util.o screen.o dialog.o key.o menu.o file.o win.o
color.o help.o find.o profile.o user.o view.o ext.ob
FSUM3067 The archive library mcslang cannot be found.
```

It is obvious that the library exists, and the path is correct, too. Nothing helps until you set the `_CC_CCMODE` variable:

```
SANDORB:/u/sandorb/mc-3.2.1: >export _CC_CCMODE=1
SANDORB:/u/sandorb/mc-3.2.1: >make
cd ../slang; make libmcslang.a
`libmcslang.a' is up to date
cc -o mc dir.o util.o screen.o dialog.o key.o menu.o file.o win.o
color.o help.o find.o profile.o user.o view.o ext.ob
SANDORB:/u/sandorb/mc-3.2.1: >
```

50.2.8 Using make

A makefile generated by newer configure scripts (produced by autoconf 2.13) does nothing when you try to use make. The command `make -v` complains that it finds no default job. Trying `make all` gives the following output:

```
groff-1.15: >make all
FSUM9380 make: Unknown option +-
Usage: make [-eBinpqrstuvVx] [-k|-S] [-c dir] [-D macro=value] [-f file]
[macro=value ...] [tar]
FSUM8226 make: Error code 2
FSUM8226 make: Error code 255
```

`make clean` is not any better.

50.2.9 Differences between make and gmake

The following makefile recipe works with /bin/make, but not with gmake:

```
c89 -Wl,EDIT=NO -W0,"langlvl(extended)" $(CFLAGS) -o $$ @$@.c
```

This is due to the construct `$$`

50.2.10 Binder problems in OS/390 V2.9

This problem is due to the incorrect behavior of the C/C++ binder under OS/390 V2.9. When this step fails, the partially generated file is not deleted. When the make process is rerun, the partial file is considered as a good one, making the new process fail again (incomplete library found, incorrect executable, and so on). In those cases, the file must be deleted manually. This bug is not present in V2.10.

50.2.11 Codeset for internal strings

The C compiler can be forced to store all constant strings in the source in a different code set. Here is how to store them in ISO8859-1:

```
cc -D__STRING_CODE_SET__="ISO8859-1" -o iso e.c
```

Following is a brief example:

```
$ cat e.c
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Hello world!\n");
    return 0;
}
$ cc -O o ISO e.c
$ ./ISO
Hello world!
$ cc -D__STRING_CODE_SET__="ISO8859-1" -o ISO e.c
$ ./ISO
çÄ%?€İ?ÊÀŽ$
```

If we compile the file `e.c` the normal way, it works perfectly and prints the expected greetings. After we compile it with `-D__STRING_CODE_SET__="ISO8859-1"` the output is unreadable, as the character codes used are ASCII.

50.2.12 Function getenv()

Using the function `getenv()` to get the content of an environment variable returns a pointer to a buffer containing the buffer of this variable. On

successive calls to this function, this buffer is reused and previous results are overwritten.

Example:

```
main()
{
    char *p1, *p2;
    p1 = getenv("TERM");
    printf("p1:%s\n", p1);

    p2 = getenv("HOME");
    printf("p1:%s\n", p1);

    printf("p2:%s\n", p2);
}
```

Output: "rxvt"

Output: "/u/ralfs"

Output: "/u/ralfs"

To fix this problem, use the function `__getenv()` instead, which returns a pointer to the value in a copy of the environment.

50.2.13 Configure fails because of ls alias

The following error was generated from a number of configure scripts:

```
checking whether build environment is sane...
configure: error: ls -t appears to fail.
Make sure there is not a broken alias in your environment
configure: error: newly created file is older than distributed files!
Check your system clock
```

After investigation, the problem was determined to be the following line in the `.setup` file:

```
alias ls='ls -tCF'
```

Because it was in the user's `.setup` file, this alias was being set for every process created, including all configure scripts. Although it could be argued that the configure script should not fail when such an alias is set, the solution was to move the alias to the user's `.profile` script.

50.2.14 Problems with `setuid()`, `setgid()` or `__passwd()` functions

OS/390 UNIX and OS/390 security must first be considered. OS/390 UNIX can be run in one of two different security models:

- Standard UNIX security
- UNIX security together with traditional OS/390 security

If BPX.DAEMON is not defined to the RACF FACILITY class, OS/390 UNIX is using standard UNIX security. This scenario leaves open doors into the MVS side of OS/390. Any superuser can change the UID of a process using `setuid()` or `seteuid()` functions, or use the `__passwd()` function.

If BPX.DAEMON is defined to the FACILITY class, the superuser must be permitted to the BPX.DAEMON FACILITY class profile in order to change UIDs or check system passwords.

With BPX.DAEMON defined, a superuser process can successfully run this service only if both of the following conditions are met:

- The caller's user identity is permitted to the BPX.DAEMON facility class profile.
- All programs running in the address space have been fetched from a security product-controlled library. A library identified to the RACF program control facility is an example of a security product-controlled library.

When BPX.DAEMON is defined, all programs using privileged functions must be put under program control, because this daemon has to fork itself and `setuid()` to the user who is requesting access to a share.

For OS/390 Version 1 Release 3 and earlier, the only way to put OS/390 UNIX executables under program control was to relink the executable into a PDS or PDSE, put it in the link pack area, and set the sticky bit on from OS/390 UNIX.

For OS/390 Version 2 Release 4 and later, this can be done from OS/390 UNIX by invoking the `extattr` command. This command allows the extended attributes to be set. To view the extended attributes, the `-E` flag can be used with the `ls` command.

Primarily, the `extattr` command resulted in the following error:

```
FOMF0303I smbd: chattr() error: rv=-1, errno=8B, rsn=0924041B
```

This error message is less than intuitive, but the following more concise message in the OS/390 system log was issued:

```
ICH408I USER(MIKEM ) GROUP(SYS1 ) NAME(MIKE MACISAAC)
BPX.FILEATTR.PROGCTL CL(FACILITY)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

The problem was that the user ID did not have enough authority to modify the RACF Facility class profile BPX.FILEATTR.PROGCTL. To give authorization to the user ID, the following command was entered from the ISPF command line (ISPF option 6):

```
RLIST FACILITY BPX.FILEATTR.PROGCTL ALL 1
PERMIT BPX.FILEATTR.PROGCTL CLASS (FACILITY) ID (MIKEM) ACCESS (READ) 2
SETR RACLIST (FACILITY) REFRESH 3
```

1. This command verifies the existence of the BPX.FILEATTR.PROGCTL facility. A list of the details of this facility should be shown.
2. This command gives the user MIKEM permission to read the facility.
3. This command refreshes RACF.

After these commands were issued, the `extattr +p` command ran successfully from OS/390 UNIX.

50.2.14.1 gcc

An entire redbook would be required in order to do justice to the gcc port to OS/390. However, we did try a few operations with gcc and our experience was as follows:

We obtained the gcc compiler from:

<http://www.cozx.com/~dpitts/gcc.html>

```
$ ls -l *.Z
-rw-r----- 1 MIKEM   SYS1      6718117 Feb 25 15:28 gcc-2_8_1-bin_tar.Z
```

```
$ c++ types.C
$ gcc types.C
gcc: installation problem, cannot exec `cc1plus': EDC5129I No such file
or directory.
```

```
In file included from server1.c:27:
/usr/include/arpa/inet.h:164: unbalanced `#endif'
```

```
$ head -10 inet.h

??=ifndef __arpa_inet
??=ifdef __COMPILER_VER__
??=pragma filetag("IBM-1047")
??=endif
#define __arpa_inet 1
#pragma nomargins nosequence
#pragma checkout (suspend)

/*****
* <arpa/inet.h>    header file
*
* *****/
```

Chapter 51. Summary

“The Free Software movement aims to provide certain freedoms for all computer users, including the right to change a program, the right to redistribute copies, and the right to publish modified versions. The most effective way to protect these rights is to deny anyone else the power to take them away from you.”

Richard Stallman on freedom and the GNU GPL

As stated in Chapter 1, “Introduction” on page 1, the cultural change that open source software is creating is a direct contradiction to the OS/390 (MVS) culture. During our short redbook project, besides writing this book and the accompanying CD, we tried to give credit where it was due and to give back to the open source community.

Some of us have used free software for more than ten years. For this project, we carefully selected some of the tools that we’ve used for a long time, and ported some of them to OS/390, in order to make your work and ours more pleasant and effective.




If you manage to port other software, perhaps one of your favorite toys or tools, consider sharing it with the rest of us in the software community!

51.1 Summary of packages

Table 2 is a list of all the packages addressed in this redbook and on the CD. Because of legal issues with the licences of some of the packages, most but not all of the code is included on the CD.

Table 2. Open source software packages

Package	Version	ASCII or EBCDIC	License	z/OS changes in main-stream	z/OS changes on CD?	Package owner/person or team porting to OS/390
Apache	1.3.22	ASCII	Apache	Yes	No	apache.org
Autoconf	2.13	ASCII	GPL	No	Yes	GNU, changes by IBM Redbook team
Automake	1.4	ASCII	GPL	No	Yes	GNU, changes by IBM Redbook team
bash	2.03	ASCII	GPL	No	Yes	GNU, changes by Greg Smith of Lockheed Martin

Package	Version	ASCII or EBCDIC	License	z/OS changes in main- stream	z/OS changes on CD?	Package owner/person or team porting to OS/390
Bison	1.25	ASCII	GPL	No	Yes	GNU, changes by MKS
CVS	1.11.05	ASCII	GPL	No	Yes	cyclic.com/GNU, changes by IBM Redbook team
diffutils	2.7	ASCII	GPL	No	Yes	GNU, changes by IBM Redbook team
Emacs	19.34	ASCII	GPL	No	Yes	GNU changes by MKS
fgres	1.0	ASCII	fgres 	None needed	No	W.Stumvoll
findutils	4.1	ASCII	GPL	None needed	No	GNU, changes by IBM Redbook team
flex	2.5.4a	ASCII	GPL	No	Yes	GNU, changes by MKS
gd	1.8.4	ASCII	gd 	No	Yes	GNU, changes by IBM Redbook team
Gmake	3.79.1	ASCII	GPL	No	Yes	GNU, changes by MKS
gnats	3.113	ASCII	GPL	No	Yes	Cygnus, changes by IBM Redbook team
GNU grep	2.4.2	ASCII	GPL	No	Yes	GNU, changes by IBM Redbook team
Groff	1.17	ASCII	GPL	Yes	Yes	GNU, changes by IBM Redbook team
Gzip	1.2.4	ASCII	GPL	none needed	No	GNU
ID-utils	3.2d	ASCII	GPL	No 	Yes	GNU, changes by IBM Redbook team
Infozip	zip 2.2 unzip 5.32	ASCII	unique	Yes	No	Infozip, changes by Paul von Behren, Storage Technologies
ltools	6.3	ASCII	X con- sortium	No	Yes	Jim Tison, IBM
jpeg library	v6b	ASCII	Thomas G. Lane	none needed	No	Independent JPEG Group, IBM Redbook team

Package	Version	ASCII or EBCDIC	License	z/OS changes in main- stream	z/OS changes on CD?	Package owner/person or team porting to OS/390
Korn shell	2001-07-04	ASCII	AST open source licence	Yes	Yes	David Korn, David Korn and IBM Redbook team
Less	358	ASCII	GPL/ Less License	None needed	No	GNU/Mark Nudelman
libpng	1.0.10	ASCII	Glenn Randers-Pehrson	None needed	No	Guy Eric Schlnat, Group 42, Inc, changes by IBM Redbook team
libtool	1.28	ASCII	None	N/A	Yes	Equivalent of GNU libtool, coded from scratch by Jeff Trawick
lynx	2.8.3	ASCII	GPL	None needed	No	GNU, changes by IBM Redbook team
m4	1.4	ASCII	GPL	No	Yes	GNU, changes by MKS
MySQL	3.23.39	EBCDIC	GPL	No	Yes	MySQL AB, changes by IBM Redbook team
nedit	5.1.1	ASCII	GPL	No	Yes	Nedit Organization, changes by Phil Sidler, Airborne Express
openssh	2.9p1	ASCII	BSD licence	No	not on CD	openssh.com
openssl	0.9.6a	ASCII	OpenSSL License and the original SSLeay license	No	Yes	www.openssl.org
patch	2.5.4	ASCII	GPL	No	Yes	GNU, changes by Peter Prymmer, Forte Software
Perl	5.6.1 and 5.00.13	ASCII	Artistic/ GPL	None needed	No	Perl organization, changes by IBM Redbook team
php	4.0.5	ASCII	The PHP Group	No	Yes	The PHP Group, changes by IBM Redbook team

Package	Version	ASCII or EBCDIC	License	z/OS changes in main- stream	z/OS changes on CD?	Package owner/person or team porting to OS/390
rsc	5.7	EBCDIC	GPL	No	Yes	GNU, changes by MKS
rlogin client	old	EBCDIC	Univ. of California	N/A	Yes	No apparent owner, changes by Jean Yves Girard, IBM
rsh client	old	EBCDIC	Univ. of California	N/A	Yes	No apparent owner, changes by Jeff Trawick, IBM
Samba	2.2.1a	ASCII	GPL	No	Yes	Samba team, changes by IBM Redbook team
sntp	1.6	ASCII	N. Maclaren	No	Yes	N. Maclaren, changes by IBM Redbook team
squid	2.4stable1	ASCII	GPL	No	Yes	GNU, IBM Redbook team
The Hessling Editor - THE	3.0	ASCII	Mark Hessling	No	Yes	Mark Hessling, changes by IBM Redbook team
vim	6.0a1	ASCII	vim license	Yes	No	Bram Moolenaar, changes by - Ralf Schandl, IBM
Webalizer	2.01-06	ASCII	GPL	No	Yes	GNU, changes by IBM Redbook team
xterm	XFree86 3.9.18	ASCII	X con- sortium	Yes	No	Thomas E. Dickey, OS/390 changes by Greg Smith of Lockheed Martin
zlib	1.1.4	ASCII	Jean-loup Gailly and Mark Adler	None required	No	Jean-loup Gailly and Mark Adler, and changes by IBM Redbook team
Notes: 1 See the License section in the respective chapter. 2 OS/390 changes have been sent to package owner.						

Appendix A. The GNU General Public License

The article also repeats a widespread misunderstanding of the GNU GPL:

But the GPL's expectation is that even the tiniest bit of GPLd source code, combined with your own, can "infect" your program with the GPL's redistribution terms.

Strictly speaking, nothing we free software developers do can legally require others to release their code in any particular way. If someone has used some of my GPL-covered code in a program, and releases the program, I cannot make that person release the program under the GPL. I can, however, deny permission to release my code on any other basis. That is what the GPL does.

If he has used a large amount of my code, he might decide to release the whole program under the GNU GPL, rather than taking out my code (which he is always entitled to do). This is how the GPL works, pragmatically, to encourage free software development.

Richard Stallman rebutting an article by Stig Hackvan criticizing the GPL

Following is the GNU General Public License (GPL), with explicit reprint permission, from the Web site:

<http://www.gnu.org/copyleft/gpl.html>

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the

freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included

without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then

this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING

THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy *name of author*

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain
conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program 'Gnomovision'
(which makes passes at compilers) written
by James Hacker.
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix B. The Open Source Definition

The Open Source Definition is important in that it is widely agreed upon in the open source community. The most current version is maintained on the Web at:

<http://www.opensource.org/osd.html>

Following is the Open Source Definition (version 1.7), reprinted with permission. Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

B.1 Free redistribution

The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

B.2 Source code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

B.3 Derived works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

B.4 Integrity of the author's source code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The

license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

B.5 No discrimination against persons or groups

The license must not discriminate against any person or group of persons.

B.6 No discrimination against fields of endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

B.7 Distribution of license

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

B.8 License must not be specific to a product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

B.9 License must not contaminate other software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

B.10 Sample licenses

The GNU GPL, BSD, X Consortium and Artistic licenses are examples of licenses that we consider conformant to the Open Source Definition. So is the MPL.

Appendix C. The IBM Public License

The IBM Public License is IBM's version of a license for both open source software and documentation. Following is the text of the IBM Public License:

IBM PUBLIC LICENSE

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS IBM PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of International Business Machines Corporation ("IBM"), the Original Program, and
- b) in the case of each Contributor,
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means IBM and any other entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Original Program" means the original version of the software accompanying this Agreement as released by IBM, including source code, object code and documentation, if any.

"Program" means the Original Program and Contributions.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright

license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability

- for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
- iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
- iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Each Contributor must include the following in a conspicuous location in the Program:

Copyright 1994, 1998, International Business Machines Corporation and others.
All Rights Reserved.

In addition, each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, If Recipient institutes

patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

IBM may publish new versions (including revisions) of this Agreement from time to time. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. No one other than IBM has the right to modify this Agreement. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Appendix D. Using the additional material

This redbook contains additional material in CD-ROM format. See the appropriate section below for instructions on using or downloading each type of material. See Chapter 3, "What's on the CD-ROM" on page 17 for a complete description. This appendix is included for a quick reference.

D.1 Using the CD-ROM

The CD-ROM that accompanies this redbook contains the following:

<i>File name</i>	<i>Description</i>
5944src.tar	Source code
5944bin.tar.Z	Binaries
README.txt	README file

D.1.1 System requirements for using the CD-ROM

The following system configuration is recommended for optimal use of the CD-ROM:

Hard disk space:	440 MB for source, or 137 MB for binaries
Operating System:	OS/390 (V2R7 or later for binaries)

See 3.4, "Installation size" on page 22 for a complete description of disk space requirements.

D.1.2 How to use the CD-ROM

You can access the contents of the CD-ROM by FTPing it to OS/390 in binary. Unwind the source distribution with the command:

```
$ pax -rf 5944src.tar
```

Unwind the binary distribution with the command:

```
$ pax -rzf 5944bin.tar.Z
```

There will be a README file with a brief description of the two distributions. See Chapter 3, "What's on the CD-ROM" on page 17 for a much better description of how to use the CD.

D.2 Locating the additional material on the Internet

The CD-ROM associated with this redbook is also available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG245944>

Alternatively, you can go to the IBM Redbooks Web site at:

<http://www.redbooks.ibm.com/>

Select the **Additional materials** and open the directory that corresponds with the redbook form number.

Appendix E. Special notices

This publication is intended to help OS/390 programmers and system administrators to create and maintain a more usable, current system. The information in this publication is not intended as the specification for any of the open source software packages that are described.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®	AS/400
DB2	Language Environment
Manage. Anything. Anywhere.	Netfinity
Open Edition	OS/2
OS/390	RAA
RACF	RS/6000
S/390	SP
System/390	Lotus
Domino	Notes
Tivoli	TME
NetView	Tivoli Ready
Tivoli Certified	Redbooks

Redbooks Logo 

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Københavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix F. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

F.1 IBM Redbooks publications

For information on ordering these publications, see “How to get IBM Redbooks” on page 315.

- *Networked Applications on OS/390 UNIX*, SG24-5447
- *S/390 File and Print Serving*, SG24-5330

F.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at <http://www.redbooks.ibm.com/> for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
S/390 Redbooks Collection March 2000	SK2T-2177
Networking Redbooks Collection June 2000	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection Kit (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

F.3 Other resources

These Web sites and publications are also relevant as further information sources:

- <http://www.gnu.org>

The GNU Web site is a perfect place to find open software or documentation. We hope this organization will continue to provide this good service for the software development community. The site is quoted

verbatim a number of times in this book. The following explicit permission is used:

Copyright (C) 1999 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

- *Open Sources: Voices from the Open Revolution*, authored by many, edited by Chris DiBona, Sam Ockman & Mark Stone, 1st edition, O'Reilly & Associates, January 1999; ISBN: 1565925823

The *Open Sources* book is cited many times in this redbook. The following line gives permission:

Verbatim copying and duplication is permitted in any medium provided this notice is preserved.

You can find this book on the Internet at:

<http://www.oreilly.com/catalog/opensources/book/copyright.html>

- *Software Portability with Imake* by Paul DuBois, 2nd edition, O'Reilly and Associates, October 1996; ISBN: 1565922263
- Richard Stallman is quoted in the conclusion chapter. Explicit permission is granted in the line:

Copyright 1999 Richard Stallman. Verbatim copying and redistribution of this article are permitted in any medium, provided this notice is preserved.

- *Using GNU configuration tools* by Dirk Vermeir.

You can find this book on the Internet at:

<http://tinf2.vub.ac.be/~dvermeir/manuals/autocookbook/cookbook.html>

F.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- <ftp://ftp.mks.com/pub/s390/gnu/>

Several open source packages ported by MKS are available at this location.

- <http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxalpor.html>

The Porting Guide (March 2001) is a "porting webbook" of practical advice and conceptual information, designed to assist you with your port.

- <http://home.flash.net/~marknu/less/>

The official home page for the utility less.

- <http://sources.redhat.com/gnats/>

The home page for the GNU Bug Tracking System gnats.

- http://sources.redhat.com/gnats/gnats_toc.html

The online gnats manual *Keeping Track: Managing Messages With GNATS - The GNU Problem Report Management System*, Version 3.110 (January 1996). Authors: Jeffrey M. Osier, Brendan Kehoe, Cygnus Support.

- <http://www.jclark.com/bio.htm>

James Clark (jjc@jclark.com), author of groff, has an interesting home page

- <http://groups.google.com/>

The Google Groups USENET archive has articles dealing with many problems you met during your port.

- <http://personal.riverusers.com/~thegrendel/build.html>

Here you'll find a document explaining open source packages, *Building and Installing Software Packages for Linux* by Mendel Cooper. It is a comprehensive guide to building and installing "generic" UNIX software distributions, especially under Linux.

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** <http://www.redbooks.ibm.com/>

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

<input type="checkbox"/> Invoice to customer number	
---	--

<input type="checkbox"/> Credit card number	
---	--

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

A

- Apache 33
 - customization and operation 37
 - getting from the Internet 42
 - history 6
 - installation 33
 - license 41
 - RACF set up 37
 - running CGI programs 39
 - starting the server 38
- Artistic License 14
- autoconf 43
 - customization and operation 44
 - getting from the Internet 46
 - installation 43
 - license 45
 - porting to OS/390 46
- automake 49
 - customization and operation 50
 - getting from the Internet 51
 - installation 49
 - license 51
 - porting to OS/390 51

B

- bash 53
 - customization and operation 54
 - getting from the Internet 54
 - installation 53
 - license 54
- Behlendorf, Brian 8
- bind 4
- bison 75
 - customization and operation 76
 - getting from the Internet 77
 - installation 75
 - license 77
 - porting to OS/390 78
- BSD
 - original distribution 2

C

- Cathedral and the Bazaar, The 14
- CD
 - what is on it 17

- Cerf, Vint 3
- compiler
 - link flag EDIT=NO 271
 - order of parameters 276
 - which one to use 276
- configure
 - problems encountered 276
- configure script 28
- contributors xxi
- CVS 57
 - customization and operation 58
 - getting from the Internet 59
 - installation 57
 - license 58
 - porting to OS/390 59
 - use in OSS development 8

D

- diffutils 61
 - customization and operation 62
 - getting from the Internet 63
 - installation 61
 - license 62
 - porting to OS/390 63
- Dignus Web site 11
- DISPLAY
 - setting environment variable 273
- DNS
 - origin 4

E

- emacs 65
 - customization and operation 67
 - getting from the Internet 67
 - installation 65
 - license 67
 - porting to OS/390 67
- e-mail
 - origin 4
- Entropy Gathering Daemon (EGD) 173
- environment variable
 - DISPLAY 273
- environment variables
 - MANPATH 23
 - PATH 23
- executable size

- reducing 271
- F**
 - fgres 69
 - customization and operation 69
 - getting from the Internet 70
 - installation 69
 - licence 69
 - porting to OS/390 70
 - findutils 71
 - customization and operation 72
 - getting from the Internet 73
 - installation 71
 - license 73
 - flex 75
 - customization and operation 76
 - getting from the Internet 77
 - installation 75
 - license 77
 - free software 1
 - Free Software Foundation 5

- G**
 - gcc 283
 - gd 81
 - customization and operation 81
 - getting from the Internet 83
 - installation 81
 - license 83
 - porting to OS/390 84
 - getenv() 280
 - gmake 85
 - customization and operation 86
 - getting from the Internet 86
 - installation 85
 - license 86
 - GNATS
 - use in OSS development 9
 - gnats 89
 - block diagram 90
 - customization and operation 91
 - getting from the Internet 91
 - installation 90
 - licence 91
 - porting to OS/390 92
 - GNU
 - brief history 5
 - Web site 11

- GNU/Linux 7
- GPL 13
 - complete text 289
- grep, GNU 93
 - getting from the Web 95
 - installation 93
 - licence 95
 - porting to OS/390 95
- groff 97
 - getting from the Internet 99
 - installation 98
 - licence 99
 - porting to OS/390 99
- gzip 105
 - customization and operation 106
 - getting from the Internet 107
 - installation 105
 - license 107
- H**
 - hints and tips 271
 - httpd.conf 37

- I**
 - IBM Public License 15
 - complete text 299
 - id-utils 109
 - customization and operation 110
 - getting from the Internet 110
 - installation 109
 - license 110
 - porting to OS/390 110
 - IETF 3
 - imake 117
 - porting to OS/390 122
 - imboot 119
 - infozip 113
 - customization and operation 113
 - getting from the Internet 115
 - installation 113
 - license 114
 - porting to OS/390 115
 - installation
 - package ordering 17
 - packages on CD 18
 - to which directory 23
 - Internet
 - origin 3

itools 117
 customization and operation 121
 getting from the Internet 122
 installation 119

J

Joy, Bill 2
JPEG library
 getting from the Internet 127
 installation 125
 license 126

K

Kahn, Bob 3
Korn shell 129
 customization and operation 130
 getting from the Internet 130
 installation 129
 license 130

L

less 131
 customizing and operation 133
 getting from the Internet 135
 installation 133
 licence 134
 porting to OS/390 135

LGPL 14

libpng
 getting from the Internet 140
 installation 138
 license 138
 porting to OS/390 140

libtool
 installation 143
 license 144

libz 173

licensing software 13

Linux 7
 kernel 7

list server
 mvs-oe 10

locate command 72

lynx
 customization and operation 146
 installation 145
 license 146

M

m4 147
 customization and operation 147
 getting from the Internet 148
 installation 147
 license 147
 porting to OS/390 148

make
 GNU make 85
makedepend 119
MANPATH
 setting environment variable 23

McCool, Rob 33

MI/X 274

mkdirhier 119

mkid 110

MKS

 OS/390 GNU utilities 11

Mosaic 6

Mozilla Public License 14

MSNTP 233
 customization and operation 233
 installation 233
 license 234

msub 119

mvs-oe list server 10

MySQL 151
 customization and operation 152
 license 153

N

nedit 163
 customization and operation 165
 getting from the Internet 166
 installation 163
 license 166
Netscape Public License 14
news groups
 origin 4

O

observations in general 271
Open Source Definition 14
 complete text 297
Open Source Software xvii
open source software
 brief history 2
 cultural change 1, 285

- development model 7
- giving back 32
- introduction 1
- overall build process 27
- packages in general 25
- summary table 285
- terminology 1
- using the source code 32
- Web sites 10
- OpenSSH 167
 - getting from the Internet 172
 - license 170
- openSSH
 - porting to OS/390 172
- OpenSSL
 - getting from the Internet 172
 - license 169
- openSSL 167, 173
 - porting to OS/390 172
- OS/390
 - cultural differences 1, 285
 - logging in from Linux 273
 - missing commands, using aliases 272
 - porting hints and tips 271
- OS/390 UNIX
 - Tools and Toys Web site 10

P

- patch 185
 - customization and operation 186
 - getting from the Internet 187
 - installation 185
 - license 187
 - porting to OS/390 187
- patch file 187
- PATH
 - setting environment variable 23
- Perens, Bruce 1
- Perl 189
 - accessing MVS datasets 193
 - customization and operation 192
 - getting from the Internet 195
 - installation 189
 - license 195
 - limitations on OS/390 194
 - porting to OS/390 196
- Perl 5.6 197
 - getting from the Internet 198

- installation 197
- license 198
- PHP 199
 - customization and operation 201
 - installation 200
 - license 201
- Preface xvii
- problems encountered 276
 - aliases set in profile 281
 - make vs. gmake 280
 - newlines as a constant 278
 - order of compiler parameters 276
 - static constructor test fails 277
 - using wrong compiler 276
- Public Domain 14

R

- RCS 209
 - customization and operation 209
 - getting from the Internet 210
 - installation 209
 - license 210
 - porting to OS/390 211
- redbook team xix
- Ritchie, Dennis 2
- rlogin 213
 - customization and operation 213
 - getting from the Internet 214
 - installation 213
 - license 214
- rsh 217
 - customization and operation 217
 - getting from the Internet 219
 - installation 217
 - license 218

S

- Samba
 - customization and operation 232
 - getting from the Web 232
 - license 232
 - porting to OS/390 223, 228, 232
- Samba V2 221
 - customization and operation 222
 - getting from the Internet 223
 - installation 221
 - license 222
- SNTP 233

Squid 237
 customization and operation 238
 installation 237
 license 239
Stallman, Richard 5
summary table 285

T

TCP/IP
 origin 3
THE
 customization and operation 244
 installation 244
 license 246
Thompson, Ken 2
Tomlinson, Ray 4
Tools and Toys Web site 10
Torvalds, Linus 7
troff 97

U

UNIX
 brief history 2
 System V 3
unzip 113
updatedb command 72
Usenet 4

V

VIm
 getting from the Internet 257
Vim 249
 customization and operation 254
 installation 249
 licence 256
 porting to OS/390 257

W

Webalizer 261
 customization and operation 262
 getting from the Internet 263
 installation 261
 license 263
-WI,EDIT=NO 271
World Wide Web
 origin 6

X

X server
 MI/X 274
xmkmf 119
xterm 265
 customization and operation 266
 getting from the Internet 267
 installation 265
 license 266
 porting to OS/390 267

Z

zip 113
zlib library 269
 getting from the Internet 270
 installation 269
 license 269

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at ibm.com/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number	SG24-5944-01
Redbook Title	Open Source Software for z/OS and OS/390 UNIX
Review	<div></div> <div></div> <div></div> <div></div> <div></div> <div></div>
What other subjects would you like to see IBM Redbooks address?	<div></div> <div></div> <div></div>
Please rate your overall satisfaction:	<input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor
Please identify yourself as belonging to one of the following groups:	<input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	<input type="radio"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. ibm.com/privacy/yourprivacy/



Open Source Software for z/OS and OS/390 UNIX



(0.5" spine)
0.5" <-> 0.875"
250 <-> 459 pages



Open Source Software

for z/OS and OS/390 UNIX

Freeware source and binaries for the mainframe!

Obtain, build and enhance open source software packages

Detailed descriptions of the most popular programs

This IBM Redbook is a survey of open source software available for OS/390. It provides a brief history of this type of software, addresses licensing issues, and describes the general model of obtaining, building and enhancing open source software packages. Many different packages are addressed, each having its own chapter.

This book includes a CD that is not an IBM product and not formally supported, but does contain many open source software packages ported to OS/390. Both the source code and the compiled executables are included.

The following packages are discussed:

Apache • Autoconf • Automake • Bash • CVS • Diffutils
Emacs • Findutils • Flex and Bison • Fgres • Gd
Gmake • Gnats • GNU grep • Groff • Gzip • Id-utils
Info-ZIP • Itools • Jpeg Library • Korn shell • Less
Libpng • Libtool • Lynx • M4 • Misctools • MySQL
Nedit • OpenSSH • OpenSSL • Patch • Perl • PHP
RCS • Rlogin client • Rsh • Samba • MSNTP • Squid
THE • Vim • Webalizer • Xterm client • Zlib



CD-ROM
INCLUDED

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks