



## **Personal Computer Communications**

### **User Manual**

Version 15.0q – April 2004

Copyright © 1987–2004 by Robert Rasmussen

All Rights Reserved



## **Rasmussen Software, Inc.**

---

10240 SW Nimbus Ave. Suite L9 Portland OR 97223 USA  
(503) 624-0360 [www.anzio.com](http://www.anzio.com) [RSI@anzio.com](mailto:RSI@anzio.com)

## Copyright Notice

This software package and this user manual are Copyright © 1987–2003 by Robert Rasmussen. All rights are reserved, worldwide. No part of this manual may be reproduced, transmitted, transcribed, or translated in any manner without the express written consent of Robert Rasmussen and Rasmussen Software, Inc., 10240 SW Nimbus Ave., Suite L9, Portland Oregon 97223.

The information contained within this document is subject to change without notice. No warranty of any kind is held in regards to this material.

## Single PC License Agreement

The Anzio product is sublicensed (not sold) to the original purchaser by Rasmussen Software, Inc., granting purchaser the right to use this software on one and only one personal computer. Backup copies of the software may be made solely for backup purposes in order to protect your investment.

The original nor any of the backup copies shall be sold, leased, gifted or otherwise be transferred or used by any other party, other than the purchaser, or on more than one machine at any given time. Purchaser may not under any circumstance tamper with any copyright protection scheme in use on the Anzio distribution media.

The liability of Robert Rasmussen and Rasmussen Software, Inc. shall not exceed the cost of the software. Neither Robert Rasmussen nor Rasmussen Software, Inc. can be held responsible for any damages due to loss of data that may result in loss of dollars.

## Acknowledgments

Windows is a registered trademark of Microsoft Corporation

Other product names are trademarks of their respective owners.


## Printing History

Edition 8.6r	March 1988
Edition 9.0	November 1988 (Second printing December 1989)
Edition 9.5	February 1990
Edition 9.7	March 1993
Edition 9.8	June 1993
Edition 10.9	July 1996
Edition 11.0	November 1996
Edition 11.4	March 1998
Edition 12.0	November 1999
Edition 12.5f	October 2001
Edition 15.0q	April 2004

## Table Of Contents

<b>Introduction .....</b>	<b>1</b>
Key Features .....	2
Organization of this Manual.....	6
Conventions.....	7
<b>Part I Narrative.....</b>	<b>8</b>
<b>1 Installing and Using Anzio .....</b>	<b>9</b>
1.1 Installing Anzio .....	9
1.1.1 Installing Anzio on the PC .....	10
1.1.2 Installing Files on the Host.....	10
1.1.3 Uninstalling Anzio.....	11
1.2 Making the Connection .....	11
1.2.1 Local Area Network (LAN) .....	11
1.2.2 Permanent Internet Connections .....	11
1.2.3 Internet Dial-up.....	11
1.2.4 Analog Modem .....	12
1.2.5 Direct Serial Connection .....	12
1.2.5.1 Configuring the Serial Port .....	12
1.2.5.2 Serial Cable Wiring.....	13
1.3 Starting Anzio.....	13
1.4 Setting Default Anzio Communication Parameters.....	14
1.4.1 Terminal Type .....	16
1.4.2 Communication Method .....	16
1.4.2.1 Network Parameters.....	16
1.4.2.2 Serial Parameters.....	16
1.4.2.3 Modem Parameters .....	17
1.5 Initial Anzio Terminal Screen .....	17
1.6 Logging On to the Host System .....	18
1.6.1 Anzio's Login Wizard.....	19
1.6.2 Troubleshooting .....	19
1.6.2.1 Network Connections .....	20
1.6.2.2 Direct Serial Connections.....	20
1.6.2.3 Modem Connections.....	21

- 1.6.2.4 General Problems..... 21
- 1.7 The Anzio Window ..... 21
  - 1.7.1 Zoomed Behavior..... 22
  - 1.7.2 Unzoomed Behavior..... 22
  - 1.7.3 80-column and 132-column Modes..... 22
  - 1.7.4 Other User Preference Items ..... 23
- 1.8 Exiting from Anzio ..... 23
- 1.9 Configuring Your Host System..... 24
  - 1.9.1 UNIX Host Configuration..... 24
- 1.10 Typical Terminal Types ..... 25
  - 1.10.1 VT100, VT220, etc. .... 25
  - 1.10.2 WYSE 50 and WYSE 60 ..... 26
  - 1.10.3 ADDS Viewpoint..... 26
  - 1.10.4 SCOANSI ..... 27
  - 1.10.5 Terminal Type "Anzio" ..... 27
- 1.11 Using Function Keys and Special Keys in Terminal Emulation ..... 28
- 1.12 Anzio's Mouse Support ..... 29
- 1.13 Selecting Text Using Keystrokes ..... 29
- 2 Help When You Need It..... 31**
- 3 Talking with Anzio..... 32**
  - 3.1 Anzio Menus..... 32
  - 3.2 Entering Commands..... 34
  - 3.3 Editing the Command Line..... 35
  - 3.4 Predefined Function Keys ..... 36
  - 3.5 Using Anzio's Language Support..... 38
    - 3.5.1 Setting the Host's Character Set..... 40
    - 3.5.2 Setting Print Encodings..... 40
    - 3.5.3 Keystroke Entry ..... 40
      - 3.5.3.1 Windows Keyboard Switching ..... 40
      - 3.5.3.2 Alt-numeric Entry ..... 41
      - 3.5.3.3 Dead-key Entry ..... 41
      - 3.5.3.4 Input Method Editors (IMEs)..... 42
      - 3.5.3.5 Add-on Programs for Far East Input . 42
      - 3.5.3.6 COMPOSE and UNICODE Commands..... 43

- 3.5.3.7 Paste from Clipboard..... 43
- 3.5.3.8 Unicode Characters in Macros ..... 43
- 3.5.4 Automatic Font Selection ..... 43
- 3.5.5 Changing the Language of the Menus..... 44
- 3.6 Using Review Mode to See Previous Data ..... 44
  - 3.6.1 REVIEW Mode Scroll Keys ..... 45
- 4 Defined Keys and Macros ..... 46**
  - 4.1 Which Keys Can I Define? ..... 47
  - 4.2 Defining a Key Macro ..... 48
    - 4.2.1 undefining a Key ..... 48
    - 4.2.2 Blocking a Key ..... 49
  - 4.3 Editing a Key Definition ..... 49
    - 4.3.1 Copying a Defined Key ..... 49
  - 4.4 Executing Key Macros ..... 50
  - 4.5 Overdefining Anzio's Default Function Keys ..... 50
  - 4.6 Using Key Definition Files ..... 50
    - 4.6.1 Saving Key Definitions ..... 51
    - 4.6.2 Loading Key Definitions ..... 51
  - 4.7 Special Characters Inside Defined Keys..... 52
    - 4.7.1 Tabs ..... 52
    - 4.7.2 Using '|' For  ..... 52
    - 4.7.3 Waiting for User Input ..... 53
    - 4.7.4 Displaying Text to the User ..... 54
    - 4.7.5 Quoting Special Keys With '~' ..... 54
    - 4.7.6 Non-ASCII Characters ..... 55
  - 4.8 Using Anzio Functions in a Defined Key ..... 55
    - 4.8.1 Keystrokes For Another Function ..... 56
    - 4.8.2 File and Screen Selection Functions ..... 56
      - 4.8.2.1 DIR/S Example ..... 56
      - 4.8.2.2 MENUBAR Example..... 57
  - 4.9 Nesting Defined Keys..... 57
  - 4.10 Special Uses of Macros ..... 58
    - 4.10.1 Startup Macro..... 58
    - 4.10.2 Onerror Macro ..... 58
    - 4.10.3 Code Activated Switch Macros ..... 59
- 5 Using A Modem to Call the Host ..... 60**

- 5.1 Types of Modem Connections ..... 60
  - 5.1.1 Direct Dial Modems..... 60
  - 5.1.2 Network Modem Connections ..... 60
- 5.2 Making the Modem Connection ..... 61
- 5.3 Signing Off..... 61
- 5.4 Modem Setup ..... 61
  - 5.4.1 Modem Connection Speed..... 62
  - 5.4.2 Data Bits and Parity ..... 62
  - 5.4.3 Error Control..... 62
  - 5.4.4 Flow Control ..... 62
- 5.5 Troubleshooting Modem Connections ..... 63
  - 5.5.1 Failure to Establish Connection ..... 63
  - 5.5.2 Baud Rate Shift..... 63
  - 5.5.3 Flow Control ..... 64
- 6 Printing..... 66**
  - 6.1 Print Levels..... 66
    - 6.1.1 High ..... 67
    - 6.1.2 Low..... 67
    - 6.1.3 Spooler ..... 68
    - 6.1.4 Raw ..... 68
  - 6.2 Choosing a Printer..... 69
    - 6.2.1 The Primary Printer ..... 69
    - 6.2.2 The Spooler Printer ..... 70
    - 6.2.3 The Raw Printer ..... 70
  - 6.3 Setting Printer Options ..... 70
  - 6.4 Choosing A Font..... 70
  - 6.5 Printing The Screen..... 71
  - 6.6 Capture To Printer..... 72
  - 6.7 Passthrough Printing ..... 72
  - 6.8 Secondary Passthrough Print ..... 72
  - 6.9 File Transfer To Printer ..... 73
  - 6.10 Printing Files..... 73
  - 6.11 End-of-Print Flush Timer ..... 73
  - 6.12 AnzioWin's Print Wizard..... 74
    - 6.12.1 Print Wizard Markup Language (PWML) ..... 75

- 6.12.2 Printing HTML ..... 75
- 6.13 Print Initialization Files ..... 75
- 6.14 Troubleshooting Printing Problems ..... 76
  - 6.14.1 When it Doesn't Print At All ..... 76
  - 6.14.2 When it Prints, but Wrong ..... 77

**Part II Data Capture and File Transfer ..... 79**

- 7 Data Capture ..... 80**
  - 7.1 Static Capture ..... 80
  - 7.2 Dynamic Capture ..... 81
  - 7.3 Control Characters During Capture ..... 81
  - 7.4 Character Set of Output File ..... 81
- 8 File Transfer ..... 83**
  - 8.1 UNIX File Transfer ..... 84
    - 8.1.1 UNIX Simple Upload ..... 84
    - 8.1.2 UNIX Simple Download Script (download) ..... 85
    - 8.1.3 UNIX Download Utility (Send-PC.C) ..... 85
    - 8.1.4 UNIX Upload Utility (Recv-PC.C) ..... 86
  - 8.2 Using Kermit ..... 86
  - 8.3 Using Zmodem ..... 88
  - 8.4 C332 File Transfer ..... 88
  - 8.5 FTP ..... 88
    - 8.5.1 Interactive FTP ..... 89
    - 8.5.2 Programmed FTP ..... 89
  - 8.6 SFTP ..... 89
    - 8.6.1 Interactive SFTP ..... 89
    - 8.6.2 Programmed SFTP ..... 90

**Part III Reference Guide ..... 91**

- 9 Command Reference ..... 92**
  - 9.1 Commands By Type ..... 92
    - 9.1.1 Operator Preference Items ..... 93
    - 9.1.2 Communication Parameters ..... 94

9.1.3	File Transfer .....	95
9.1.4	Local Processing.....	97
9.1.5	Printing .....	98
9.1.6	Macro Processing & Key Mapping.....	99
9.1.7	Interaction with Host and User.....	100
9.1.8	Miscellaneous .....	101
9.2	Alphabetical List of Commands.....	102
9.2.1	Syntax Conventions .....	102
9.2.2	Commands.....	102
	7E1 .....	102
	7E2 .....	103
	7M1 .....	103
	7M2 .....	103
	7N1.....	103
	7N2.....	103
	7O1.....	103
	7O2.....	103
	7S1 .....	103
	7S2.....	103
	8E1 .....	103
	8E2 .....	103
	8M1 .....	104
	8M2 .....	104
	8N1.....	104
	8N2.....	104
	8O1.....	104
	8O2.....	104
	ADD128.....	104
	ALLOW-QUIT [ON]   OFF .....	104
	ANSWERBACK <i>string</i> .....	105
	ASEND <i>filename</i> .....	105
	AUTO-CONNECT [ON]   OFF   TOGGLE.....	105
	AUTO-LF [ON]   OFF   TOGGLE .....	105
	AUX [ON]   OFF   TOGGLE .....	105
	AUX-BIDI.....	105
	AUX-DEBUG [ON]   OFF   TOGGLE .....	106
	AUX-READ/S <i>string</i> .....	106
	AUX-SETUP.....	106
	AUX-WRITE <i>string</i> .....	106
	BACKSPACE 8   127 .....	106
	BASENAME/S <i>filepath</i> .....	106
	BAUD <i>nnnn</i> .....	106
	BEEP [ON]   SLOW   OFF .....	107



BEEP IDLE [ON] | OFF | TOGGLE ..... 107  
 BLOCKTYPE COLUMN | STREAM | WRAP . 107  
 BMP *filename* WATERMARK BMP *filename*  
     [CELL] [*window-rect* [*view-rect* [*bright*  
     [*contrastf*]]]] ..... 107  
 BMP-ADJUST *bright* [*contrastf*] ..... 109  
 BMP-CLOSE ..... 109  
 BMP-VIEW *left top right bottom* ..... 109  
 BMP-WINDOW WATERMARK BMP-WINDOW  
     [CELL] *left top right bottom* ..... 110  
 BOX *c1 c2 r1 r2* [*type*] ..... 110  
 BREAK ..... 110  
 C332-NEXT C332-PREV ..... 110  
 CALC ..... 110  
 CALL *macro* ..... 111  
 CAPTURE [ON] | LONG | OFF CAPTURE  
     *filename* | CLOSE CAPTURE WPRN ..... 111  
 CD [*unit*] *directory* ..... 112  
 CHARSET *name* CHARSET/S *name* ..... 112  
 CHOOSEPRINTER [*name* | DEFAULT] ..... 113  
 CLIP [*c1 c2 r1 r2*] ..... 113  
 CLIP "*string*" ..... 114  
 CLONE ..... 114  
 CLOSEI CLOSEI/S ..... 114  
 CLOSEO CLOSEO/S ..... 114  
 COLOR ..... 115  
 COLOREDIT ..... 115  
 COMMTYPE *name* ..... 115  
 COMPOSE/S *x[y]* ..... 116  
 CONNECT  
     [*commtype://*][*username[:password]*@]*host*  
     *name[:port]* ..... 118  
 COPY *filename newfile*  
     COPY/S *filename newfile* ..... 119  
 CURSOR [BLINK] [ON] | OFF ..... 119  
 CURSOR BLOCK | UNDERLINE ..... 119  
 DATE/S [*format*] ..... 119  
 DATA [BITS] 5 | 6 | 7 | 8 ..... 120  
 DDEEXECUTE *command* ..... 120  
 DDEINITIATE *program topic* ..... 121  
 DDEPOKE *item data* ..... 121  
 DDEREQUEST/S *item* ..... 121  
 DDETERMINATE ..... 121

DEFAULTS..... 122  
 DEFINE *x text* ..... 122  
 DELAY *n* DELAY/S ..... 122  
 DELETE *filename* DELETE/N *filename*  
     DELETE/S *filename*..... 122  
 DIAL *phone\_number* [*wait retries*] ..... 123  
 DIR [*pathname*] DIR/S [*pathname*] ..... 123  
 DOWNLOAD-DIR [*pathname*]..... 123  
 DOWNLOAD-LOCK [*pathname*]..... 124  
 EJECT ..... 124  
 END | E ..... 124  
 ENV/S *variable*..... 124  
 ENVIRONMENT *string*..... 124  
 EVENT *eventname macro* ..... 128  
 EXPECT *value1 value2*..... 128  
 FILL *c1 c2 r1 r2 [char]* ..... 129  
 FIND/S *filespec* ..... 129  
 FINDNEXT/S ..... 129  
 FLUSH..... 129  
 FLUSHTIMER *time* ..... 130  
 FONT [*name*] [*size*] | LARGER | SMALLER |  
     BOLD ..... 130  
 FONT-ALT *fontnames*..... 130  
 FTP [PASSIVE] FTP NEW..... 131  
 FTPDELETE *hostfile* FTPDELETE/S *hostfile* 131  
 FTPGET [ASCII] *hostfile* [*pcfile*] FTPGET/S  
     [ASCII] *hostfile* [*pcfile*] FTPPUT [ASCII]  
     *hostfile* [*pcfile*] FTPPUT/S [ASCII] *hostfile*  
     [*pcfile*]..... 132  
 FULL [DUP] HALF [DUP]..... 132  
 GAUGE [ON] | OFF | TOGGLE..... 133  
 HALF [DUP] ..... 133  
 HANGUP ..... 133  
 HELP ..... 133  
 HEXPATCH *address bytes* ..... 133  
 HOLD [ON] | OFF | TOGGLE..... 134  
 HOSTNAME/S ..... 134  
 IGNUll [ON] | OFF | TOGGLE..... 134  
 INTERPRET ..... 134  
 INVOKE *macro*..... 134  
 JUMP OFF | MEDIUM | FAST ..... 134  
 KCOMMAND *command* [*parameters*]..... 135

KEEP [F7] | c1 c2 r1 r2] KEEP/N [F7] | c1 c2 r1  
r2] ..... 136

KEYS ..... 136

KILL ..... 136

KRECEIVE AUTO-OFF KRECEIVE AUTO-ON  
KRECEIVE [options] [filespec]  
KRECEIVE/S [options] [filespec] ..... 136

KSEND [options] filespec KSEND/S [options]  
filespec ..... 138

LAUNCH [options] program [parameters]  
LAUNCH/S [options] program [parameters] 139

LINE [DELAY] n ..... 140

LOCK [ON] | OFF | TOGGLE [F4] ..... 140

LOG [drive:]directory ..... 140

LOGIN-WIZARD [ON] | OFF | TOGGLE ..... 141

MAPISENDFAX phone filename [profile] ..... 141

MCISENDSTRING command  
MCISENDSTRING/S command ..... 141

MENUBAR c1 c2 r1 r2 c3 c4 off len [cols] ..... 142

MENULANGUAGE xxx ..... 142

MERGE filename ..... 142

MESSAGE string ..... 143

MKDIR dirname MKDIR/S dirname ..... 143

MONITOR [ON] | OFF | TOGGLE ..... 143

OLE INIT program.title OLE GET  
program.property OLE  
program.property=value OLE  
program.method [args] OLE/S  
program.method [args] OLE TERM ..... 144

ONERROR macro ONERROR ..... 145

ONSTARTUP macro ONSTARTUP ..... 145

OPENE filename OPENE/N filename  
OPENE/S filename ..... 146

OPENI filename OPENI/S filename ..... 146

OPENO filename OPENO/N filename  
OPENO/S filename ..... 147

PALETTE BG | BLBRIGHT | BLDIM |  
ULBRIGHT | ULDIM | FULL ..... 148

PARITY EVEN | ODD | MARK | SPACE | OFF 148

PASSTHRU ..... 149

PASSTHRU-CP name PASSTHRU-CP/S  
name ..... 149

PASSWORD text PASSWORD/S ..... 149

PASTE..... 150  
 PICK *c1 c2 r1 r2 type* [...]..... 150  
 PITCH *n* | -1 ..... 150  
 PLAYMEDIA PLAYMEDIA *filename*  
     PLAYMEDIA *filename* [CELL] *x1 y1 x2 y2*  
     [options]..... 151  
 PLAYSOUND *filename* ..... 151  
 PORT *n* ..... 152  
 PREVIEW [ON] | OFF | TOGGLE ..... 152  
 PRINT [*c1 c2 r1 r2*] PRINT/N [*c1 c2 r1 r2*]... 152  
 PRINTER *name* PRINTER WPRN ..... 152  
 PRINTER-SETUP PRINTER-SETUP *text*.... 153  
 PRINTFILE [*/charsef filename*] ..... 154  
 PRINTFONT [*name*] [*size*] [BOLD] ..... 154  
 PRINTFONT-ALT *fontnames*..... 155  
 PRINTINIT *filename*..... 155  
 PRINTLEVEL WIZARD | HIGH | LOW |  
     SPOOLER | RAW ..... 155  
 PRINTLINE [*textf*] PRINTLN [*textf*] ..... 155  
 PRINTLOW [ON] | OFF | RAW | SPOOLER.. 156  
 PRINTMACRO [*macro*] ..... 156  
 PRINTTOFILE [ON] | OFF | *filename*..... 156  
 PROMPT-ID ..... 157  
 PROMPT/S [*caption textf*]..... 157  
 PURGE..... 157  
 RAW-SETUP [DEFAULT] ..... 157  
 READ *filename*..... 158  
 RECEIVE CODED ..... 158  
 RECEIVE QUIET [ON] | OFF | TOGGLE..... 158  
 RECONNECT [ON] | OFF | STAY ..... 159  
 RECORD [ON] RECORD CANCEL RECORD  
     OFF RECORD *keyname*..... 159  
 REGTELNET [RESET]..... 159  
 RENAME *oldname newname* ..... 160  
 REPLACE/S *search-pattern replace-pattern*  
     *content* ..... 160  
 RESET ..... 160  
 RETRANSMIT ..... 160  
 REVIEW ..... 160  
 RUN [options] [*program parameters*] RUN/N  
     [options] [*program parameters*] RUN/S  
     [options] [*program parameters*]..... 160  
 RZ..... 162

SAVE [*filename*] ..... 162

SCREENMODE *option*..... 162

SCREENMODE/S ..... 162

SCREENSIZE/S *param*..... 163

SCRIPT SCRIPT *filename*  
     [E:*lang*][M:*module*][T:*timeout*][*args*] ..... 163

SCROLL [ON] | OFF | TOGGLE ..... 164

SCROLL-LOCK [ON] | OFF | TOGGLE ..... 164

SEND ..... 164

SEND *c1 c2 r1 r2* ..... 164

SEND "*string*" SEND/N "*string*" ..... 165

SET *variable=value*..... 165

SETCOLOR *n* | NORMAL ..... 165

SFTP ..... 165

SFTPDELETE *hostfile* SFTPDELETE/S  
     *hostfile*..... 165

SFTPGET *hostfile* [*pcfile*] SFTPGET/S *hostfile*  
     [*pcfile*] SFTPPUT *hostfile* [*pcfile*]  
     SFTPPUT/S *hostfile* [*pcfile*] ..... 166

SLEEP *hh mm ss*..... 166

SPEAK SPEAK "*text*" SPEAK *c1 c2 r1 r2*..... 166

SPOOLER [*name*]..... 167

STATUS [LINE] [ON] | OFF ..... 167

STAYALIVE *nn*..... 167

STOP..... 167

STOP [BITS] 1 | 1.5 | 2 ..... 167

SWITCHTO /NEXT SWITCHTO /PREV  
     SWITCHTO *title* ..... 168

SZ..... 168

TAB *i j k* ..... 168

TAB RESET ..... 168

TAB CHARACTER *x*..... 168

TAB [ON] | OFF | RESET..... 169

TERM *termtype*..... 169

TERMNAME *name*..... 170

TIME/S [*format*]..... 170

TIMEOUT *n*..... 171

TITLE *string*..... 171

TRANSFER-OPT ..... 171

TRACK-WINDOW [ON] | OFF | TOGGLE ..... 171

TRANSMIT *filename* [*param*] [TRAILER *string*]  
     TRANSMIT [ON] | OFF ..... 172

TYPE *filename* ..... 172

UNICODE/S [num] .....	173
UPPERCASE [ON]   OFF   TOGGLE .....	173
USERNAME <i>name</i> USERNAME/S .....	173
VERSION VERSION/S .....	173
WAIT <i>nnn</i> .....	173
WAITFOR <i>string</i> [timeout] WF <i>string</i> [timeout]	174
WAITQUIET <i>seconds</i> .....	174
WINDOW <i>c1 c2 r1 r2</i> [FILL] [BOX   DOUBLE]	174
WINDOW ZOOM   MINIMIZE   RESTORE .....	175
WINDOW ROTATE [0   90   180   270] .....	175
WINDOWCLOSE .....	175
WINPRINT <i>filename</i> [ <i>printer-name</i> ] .....	175
WINSTART <i>filename</i> [ <i>params</i> ] .....	175
WRITE <i>text</i> .....	176
WRITELINE [ <i>text</i> ] WRITELN [ <i>text</i> ] .....	176
XN .....	176
XRECEIVE [ <i>options</i> ] [ <i>filespec</i> ] XRECEIVE/S	
[ <i>options</i> ] [ <i>filespec</i> ] .....	177
XSEND [ <i>options</i> ] <i>filespec</i> XSEND/S [ <i>options</i> ]	
<i>filespec</i> .....	177
YRECEIVE [ <i>options</i> ] [ <i>filespec</i> ] YRECEIVE/S	
[ <i>options</i> ] [ <i>filespec</i> ] .....	178
YSEND [ <i>options</i> ] <i>filespec</i> YSEND/S [ <i>options</i> ]	
<i>filespec</i> .....	178
ZRECEIVE AUTO-OFF ZRECEIVE AUTO-ON	
ZRECEIVE [ <i>options</i> ] [ <i>filespec</i> ]	
ZRECEIVE/S [ <i>options</i> ] [ <i>filespec</i> ] .....	179
ZSEND [ <i>options</i> ] <i>filespec</i> ZSEND/S [ <i>options</i> ]	
<i>filespec</i> .....	180
<b>10 More On Starting Anzio .....</b>	<b>181</b>
10.1 Command Line Parameters & Default Files .....	181
10.1.1 'NEW' for New Connection .....	182
10.1.2 'NONE' For Clean Startup .....	182
10.1.3 Single Character – Start Defined Key .....	182
10.1.4 Names – Settings File Name or Telnet Host	
Name .....	183
10.1.5 /C Choose Settings File .....	184
10.1.6 /D Define Key .....	184
10.1.7 /H Hostname .....	185
10.1.8 /K Kiosk Mode .....	185
10.1.9 /T Communication Type .....	186
10.1.10 /V Verbose .....	186

**Part IV Technical Reference Guide..... 187**

**11 File Transfer Protocols..... 188**

11.1 Simple Upload – TRANSMIT ..... 188

11.2 TRANSMIT LONG..... 189

11.3 TRANSMIT CRC ..... 189

11.4 RECEIVE..... 189

11.5 RECEIVE WITH CRC..... 190

11.6 Simple Receive – CAPTURE ..... 190

11.7 CAPTURE LONG ..... 190

11.8 Passthrough Printing..... 190

**12 Sending Anzio Commands From The Host..... 192**

**13 Security Options ..... 194**

13.1 SRP ..... 194

13.2 SSH ..... 195

**14 Advanced Options ..... 196**

14.1 General..... 196

14.2 Screen ..... 199

14.3 Keys ..... 201

14.4 Security ..... 202

14.5 Mouse..... 204

14.6 Print Wizard..... 207

**15 Customizing Anzio..... 209**

15.1 Editing the DEF File ..... 209

**Appendix A Error Messages ..... 212**

A.1 Communication Chip Errors ..... 212

A.2 Software and File Error Messages..... 213

**Appendix B Notes On Particular Host Systems..... 216**

B.1 UNIX Hosts..... 216

**Appendix C Migration From Earlier Releases ..... 218**

**Appendix D Serial Communication Problems..... 219**  
**Appendix E Distribution Information ..... 221**  
**Index ..... 225**



## Introduction

---

Anzio is a family of personal computer (PC) communications programs. Anzio provides *terminal emulation* (also known as *telnet*) for a large set of terminal types and operating systems. That is, with your PC connected to a host system, Anzio manages the communication to appear exactly like a standard “dumb” terminal.

Anzio now supports a wide variety of languages and fonts, from Chinese to Cyrillic. Anzio also provides “smart” terminal functions including *passthrough print* and several types of file transfer.

- To install and run Anzio, see *Installing and Using Anzio*, page 9.
- To stop and exit Anzio, press the ALT key and the X key at the same time:



This will return you to the PC operating system.

---

➤ If you are migrating from an earlier version of Anzio, check Appendix C, *Migration From Earlier Releases*, page 218.

---

While Anzio is running, the PC behaves very much like the terminal it is emulating. The host system thinks it has a terminal attached, and displays information accordingly.

Pressing a keyboard key sends one or more characters to the host. Function keys () and special keys () are usually configured to send certain control-character sequences to the host system.

- Some key combinations are predefined by Anzio to control Anzio’s operation. Anzio allows you to change the definition of any key. Keys can also be defined to perform a series of Anzio commands and operations – see page 38.

Anzio sometimes uses the line at the bottom of the terminal emulation window to accept commands and display messages and status information. In general, anything you type on the bottom line goes to Anzio, and is not sent to the host.

The Anzio family currently consists of two members, AnzioWin and Anzio Lite. Both are 32-bit applications, that run on Windows 95 through Windows 2003. Older members of the family include versions of AnzioWin and Anzio Lite for 16-bit platforms (Windows 3.x), as well as versions for DOS.

### Key Features

---

Anzio's primary feature is reliable, fast terminal emulation. Other key features are listed here. Note that some features are in AnzioWin only, not in Anzio Lite.

**Multiple emulations** Anzio emulates many popular terminal types: VT100/220/320/420, SCO ANSI, AT386, Wyse 50, Wyse 60, Versyss C332, and ADDS Viewpoint, among others.

**Terminal features** Graphics character sets, answerback, downloadable keys, downloadable characters, status lines, video attributes, color, passthrough print.

**Connection methods** Anzio can connect to the host via SSH, telnet, rlogin, modem, direct serial, or PicLan.

**80- and 132-column screens** Anzio provides 80- and 132-column screens, and custom screen sizes.

**Screen control** Run in full-screen mode, with no extraneous screen clutter, or in a smaller window.

**Security** Anzio supports SSH1, SSH2, and SRP, for secure encrypted communication.

**Login wizard** If Anzio knows your username and password, it can handle the login for you.

**Language support** Anzio can display and send characters for many languages. Anzio uses 16-bit Unicode characters internally. Unicode defines over 39,000 characters used in European, Hebrew, Arabic, Chinese,


	Japanese, Korean, and other languages. If your host system provides language support, Anzio can display and print your language.
Diacritic support	Anzio has special treatment for combining diacritics, especially those used in the MARC standards for libraries.
Font coverage	Anzio automatically finds installed fonts to display special and international characters.
Linedraw characters	The characters used to draw boxes and other lines on terminals are always displayed properly, connecting to adjoining lines.
Euro support	Anzio has special treatment of the Euro character (€), so that it can be assigned to an arbitrary character value.
Faxing	Anzio has special techniques by which passthrough print jobs can be faxed automatically.
Browser integration	Anzio can register itself as the standard telnet client in Windows, so that clicking on a telnet cross-reference in a web page will invoke Anzio and create a telnet session to the indicated host.
Mouse support	Various mouse button actions can be configured to do different things, including sending screen data to the host, launching a URL, copy and paste, and more. The mouse wheel can control screen scrollbar (review). Mouse-to-host protocols as in xterm are supported, so host-end programs can work with Anzio to receive mouse events.
Sound	Anzio can issue a beep, play a standard Windows sound, or play WAV and other media files from the local PC or the Internet.
Pictures	Anzio can display a bitmap image, for screen watermarks, text-over-forms, pictures of products or employees, etc.

AUX serial devices	Anzio can accept input from auxiliary serial devices, such as barcode readers, postage scales, etc. It can also do bidirectional communication with AUX devices, such as for point-of-sale support.
Kiosk mode	Anzio can be run with a limited menu system, disabling unneeded user commands, for situations such as public-access PCs in libraries.
Lockdown	Anzio can be locked down in other ways, so users can't save any changes, can't exit improperly, etc.
Screen review	Anzio buffers data that scrolls off the top of the screen, including cleared screens. You can later scroll this data back into view. Buffer size is configurable.
Key mapping	You can control what code sequence is sent out by various special keys, including 48 function key combinations, control-keys, alt-keys, and special keys. Numeric pad keys can be mapped separately in NumLock on and NumLock off states.
Macro language	Assign user macros to unused keys. Macros can send common keystroke sequences to the host, or use Anzio's more than 200 commands. A macro can be run on program startup. Macros can be associated with mouse events.
Control from host	Programs running on the host system can control and configure Anzio and its PC, including PC file operations, DOS commands, Windows operations, and local printer control. The host can also send special commands to place and read character-based "windows" and "menus" on the Anzio screen.
Local operations	Anzio provides many local PC file operations, such as DIR, RENAME, TYPE, LOG, etc. Anzio can call any other Windows or DOS program, optionally waiting for it to finish.
Screen capture	Capture incoming data to disk file, with various options for handling control characters. International

characters can be written in any of three character sets, including Unicode UTF-8.

File transfer	Anzio provides a variety of file transfer protocols for different host systems and file types, including Kermit, Xmodem, Ymodem, and Zmodem
Interactive FTP	Anzio provides a point-and-click FTP client for transferring files to and from a host system.
Programmed FTP	Anzio provides a command-driven FTP client, allowing fully automated file transfers.
SFTP	Anzio provides both point-and-shoot and command-driven SFTP (Secure File Transfer Protocol) operations.
Copy-and-paste	Anzio can copy part or all of the screen onto the Windows clipboard (in both text and bitmap mode), and can paste clipboard text to the host program, with full international language support.
Configuration	You can configure many “comfort” features, such as colors, mouse click actions, non-blinking cursor, status/gauge line, beep pitch, etc.
Scripting	Anzio can invoke scripts in VBScript or JavaScript, with data exchange.
DDE client	Anzio can act as a DDE client, and send commands to other programs.
OLE client	Anzio can act as an OLE client, and send commands to other programs.
Languages for menus	Choose to display Anzio’s menus, prompts, etc. in Spanish, French, etc.

*And then there’s printing. Anzio has so many features for printing they require a separate list:*

Print screen	Anzio can print the current screen (even with the  key). It can also print just a portion of the screen.
Passthrough print	Anzio can print print jobs sent from the host in the “transparent print” or “local print” protocol.
Capture to printer	The operator can cause data coming to the screen to be sent to the printer.
Print levels	Anzio can send data to the printer at 4 different levels, for different print requirements. Printing can use the Windows printer driver, or can bypass it.
Auto-fit	The Print Wizard feature will analyze each print job and automatically fit it to the page. Wide format jobs can be printed in landscape mode automatically.
Mini-page	For one-at-a-time address labels, and similar jobs on continuous feed paper, Print Wizard can create a custom page length, just as long as is needed.
Multi-column	Plain text input can be converted to multi-column mode, for labels.
Multilingual	The full range of Unicode text can be printed, with special handling of linedraw characters and diacritics, and automatic font selection.
Forms overlays	Bitmap or PCL forms can be added to print jobs.
Barcodes	Barcodes can be printed on almost any printer.
Markup language	The Print Wizard Markup Language gives you the means to direct all kinds of fancy printing, including font changes, rectangles, pictures, paper size, and more.

---

### **Organization of this Manual**



---

This manual is organized in five parts:

- Part I, Narrative, describes Anzio concepts, operations, and usage. Please humor us and read Part I before you call with questions.
- Part II, Data Capture and File Transfer, describes how to use Anzio to capture host data off the screen, and how to transfer files between your PC and different host operating systems.
- Part III, Command Reference, describes all Anzio commands in alphabetical order, followed by the startup options for the Anzio program itself.
- Part IV, Technical Reference, provides some byte-level specifics on communication protocols, sending Anzio commands from the host, terminal emulation, and data capture.
- Part V, Appendices, provides additional information on Anzio and various hosts:
  - Appendix A, *Installing and Using Anzio*
  - Appendix B, *Notes On Particular Host Systems*
  - Appendix C, *Migration From Earlier Releases*
  - Appendix D, *Serial Communication Problems*
  - Appendix E, *Distribution Information*

### Conventions

---

- “Anzio” refers to any version of Anzio; that is, Anzio Lite or AnzioWin.
- “PC” is the personal computer running Anzio.
- “Host” is the computer (server) to which you are connected.
-  represents the “Carriage Return” character, generated by the “Enter” key on the PC keyboard.
-  represents function key F6 on the PC keyboard.
- “Communicate/Hangup” indicates the Communicate menu, and the Hangup menu item.

**Part I Narrative**



## 1 Installing and Using Anzio

---

This section describes how to get the Anzio program up and running:

- *Installing Anzio*
- *Making the Connection*
- *Starting Anzio*
- *Setting Default Anzio Communication Parameters*
- *Initial Anzio Terminal Screen*
- *Logging On to the Host System*
- *The Anzio Window*
- *Exiting from Anzio*
- *Configuring Your Host System*
- *Typical Terminal Types*
- *Using Function Keys and Special Keys in Terminal Emulation*
- *Anzio's Mouse Support*
- *Selecting Text Using Keystrokes*

---

➤ The README.TXT and HOWTO.TXT files on the distribution disk provide additional information about Anzio installation and usage.

---

### 1.1 Installing Anzio

---

Anzio can be installed on and run from any disk accessible to your PC – floppy, local hard disk, or networked disk. Anzio cannot be run directly from the distribution CD, because those program files are compressed.

Installing Anzio consists of:

- Transferring the program files from the distribution CD to your PC.
- Building Start menu items to initiate Anzio.
- Building one or more desktop icons to initiate Anzio.
- Placing certain information in the Windows registry

For more information on the contents of the distribution disk, and the system requirements for Anzio, see Appendix E, *Distribution Information*, page 221.

- 
- After installing the Anzio executable, you may customize it, by using a Windows resource editor to add or remove menu items and accelerator keys – see *Anzio Menus*, page 32.
- 

### **1.1.1 Installing Anzio on the PC**

Anzio is available on CD-ROM, and also on the Internet at <http://www.anzio.com>. Both methods use standard Windows installation techniques. In both cases, the installed Anzio Lite or AnzioWin will run as a demo, for up to 60 days, until and unless you enter your serial number and license code as delivered by Rasmussen Software or your software supplier.

To install from the web site:

1. Point your web browser to the appropriate page at the web site shown above.
2. Click the “download” item.
3. Depending on your browser’s capabilities, you may be able to run the installer directly from the web site.
4. Otherwise, save the file to disk, then run it.

To install from CD:

1. Insert the distribution CD into any available CD or DVD drive.
2. The Setup program may start automatically; if not, continue...
3. Start an Explorer window and navigate to the disk drive.
4. Double-click on the SETUP.EXE program.

### **1.1.2 Installing Files on the Host**

For normal connection and terminal emulation, you will not need to do anything to your host system. However, the Anzio distribution disk in-

cludes some files for use on the host computer, to a) better support the 'Anzio' terminal type, b) provide some kinds of file transfer, and c) support certain application programs that run on the host. Because these files are generally **not** needed, we'll address them later.

### 1.1.3 Uninstalling Anzio

To uninstall Anzio, just go to the "Add/Remove Programs" item in the control panel, and select Anzio.

## 1.2 Making the Connection

---

Anzio can communicate with the host system in several ways, that reduce down to a) network, b) modem, and c) direct serial. Note that if you are connecting your PC to the Internet through a modem connection to an Internet Service Provider (ISP), or to another network through a modem ("PPP dialup"), Anzio considers this a *network* connection. This is an important distinction; if you are not sure what kind of modem connection you ought to be using, check with your system administrator.

The following sections will help you get your connection in place.

### 1.2.1 Local Area Network (LAN)

If your PC is going to communicate over a local area network, simply plug the network cable into your network interface card. You will also need to configure TCP/IP at the Windows level. That is beyond the scope of this manual.

### 1.2.2 Permanent Internet Connections

If your LAN is connected to the Internet through a router, a cable modem, a DSL modem, etc., you'll be able to reach any server on the Internet, as well as local servers.

### 1.2.3 Internet Dial-up

You may need to connect to the Internet through an ISP, by telling Windows to dial your modem. Again, you'll need to have that configured

already. And you'll need to have that connection established before you run Anzio.

#### **1.2.4 Analog Modem**

To connect your PC as a terminal, through your PC's modem to a modem on a host system, you'll obviously need a modem on your PC, with its Windows driver installed, and its cord plugged into a live telephone jack. In this situation, Anzio will initiate the dialing operation.

---

➤ A DSL modem or a cable modem is different. See above.

---

#### **1.2.5 Direct Serial Connection**

This section applies only to a direct serial connection from your PC to the host system's serial port. This is often done where a dumb terminal has been replaced by a PC running Anzio.

The physical hookup between your PC and your host machine is usually a standard terminal cable. The host machine sees your PC as a plain terminal (CRT). The cable coming from the host is the same for a CRT as for Anzio.

For the initial installation test, we recommend that you unplug a line from a working CRT and connect that line to the Anzio PC. This way you'll know that the line works and is properly configured, eliminating one potential source of problems.

##### **1.2.5.1 *Configuring the Serial Port***

Your PC must have a *serial port (comm port)*. This may be on the mother board, on an add-in board, or on a USB-to-serial adapter. The port must be installed to Windows, with no hardware conflicts, and operational.

The configuration of the serial port (baud rate, etc.) is all handled by Anzio. Settings in the control panel are ignored.

### 1.2.5.2 Serial Cable Wiring

The PC serial port always has a male plug, usually 9-pin, sometimes 25-pin. A 25-pin female plug in the back of a PC is probably a parallel printer port - don't use it for a serial connection.

If the cable from your host system ends in a male plug, you will need a 25-pin *female-to-female gender converter*, available from any PC supplier. In addition, if your cable terminates in a 25-pin plug, and your PC has a 9-pin plug, you will need a standard 25- to 9-pin adapter.

---

➤ Do not use a “null modem” adapter.

---

The standard 9-pin wiring connection is:

1	CD (carrier detect)
2	SD (send data)
3	RD (received data)
4	DTR (data terminal ready)
5	GND (ground)
6	DSR (data set ready)
7	RTS (request to send)
8	CTS (clear to send)
9	RI (ring indicator)

The standard 25-pin wiring connection is:

2	SD (send data)
3	RD (received data)
4	RTS (request to send)
5	CTS (clear to send)
6	DSR (data set ready)
7	GND (ground)
8	CD (carrier detect)
20	DTR (data terminal ready)
22	RI (ring indicator)

## 1.3 Starting Anzio

---

Start Anzio from its icon, or use the Windows Start menu – Start/Programs/Anzio xxx/AnzioWin.

Anzio will prompt for communication parameters, described in the next section.

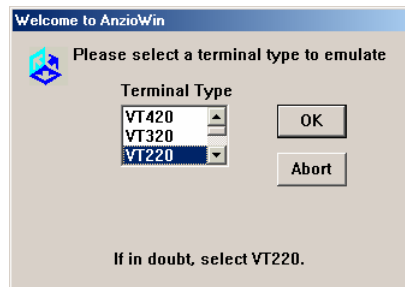
## 1.4 Setting Default Anzio Communication Parameters

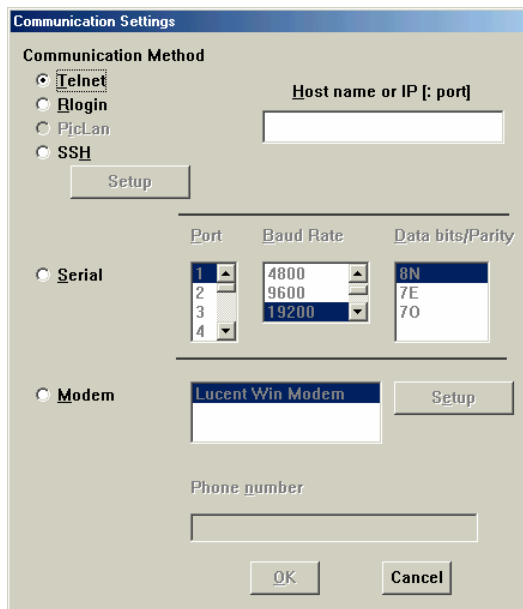
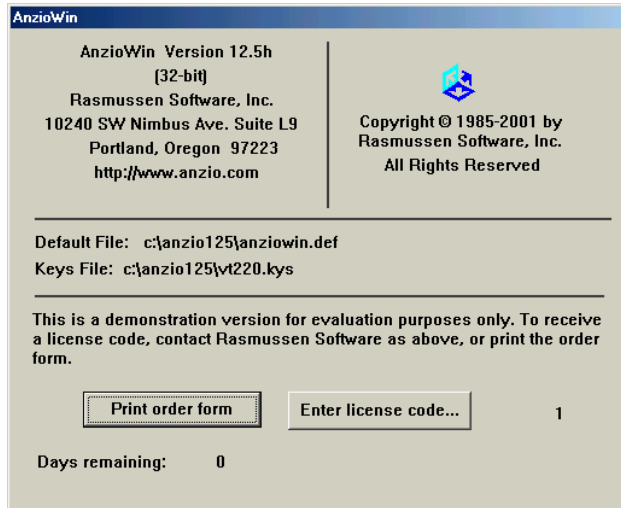
---

The first time you run Anzio, you must set several communication parameters:

- *Terminal Type*
- *Communication Method*
  - *Network Parameters* – Type, Host name or address
  - *Serial Parameters* – Port, Baud rate, and data bits and parity.

Anzio shows three dialogs, the first to select terminal type, the second showing the Anzio version and any files used, and the third for selecting communication parameters:





The parameters you enter here will be saved in a *defaults file* (also known as a *settings file*), normally named "AnzioWin.def". The settings file also contains all the configuration settings for Anzio, except for key mappings.

When you exit Anzio, it will ask if you want to save changes to the settings file. The next time you run Anzio, it reads the settings file to set the default system settings. You may change these settings at any time with Anzio commands or menu items.

#### 1.4.1 **Terminal Type**

Anzio can emulate many different terminal types. The host is probably expecting a particular terminal type. Select that terminal type to start. If in doubt, select "VT220". Later, you can change the current terminal type with Communicate/Terminal Type or the TERM command.

See *Typical Terminal Types*, page 25, for more information on choosing a terminal type.

#### 1.4.2 **Communication Method**

Anzio needs to know how your PC will communicate with the host.

- If you do not know some of these parameters, ask a local expert (*system guru*) for help.

##### 1.4.2.1 **Network Parameters**

A network connection requires a previously installed Windows interface. Anzio checks to see which of the interface types are installed, if any, and enables the corresponding radio buttons.

- Select the network interface type (Telnet, Rlogin, PicLan, etc.).
- Enter either the host name, or an IP (*Internet Protocol*) address, for the host (server) you're connecting to, **not** for this PC.
- If you need to connect to a non-standard port, follow the host name or IP with a colon (":") and a host port number.
- If you choose SSH, you can configure the SSH parameters, by clicking the Setup button next to "SSH".

##### 1.4.2.2 **Serial Parameters**

Port                      Anzio needs to know which serial port is connected to your host computer – see *Configuring the Serial*



*Port*, page 12.

Select (or enter) the PC's serial port number.

**Baud Rate** Your PC and your host machine must communicate at the same speed, known as the *baud rate* (data bits per second). The host's baud rate is generally preset, and Anzio must match it.

Select the baud rate at the prompt.

**Data Bits/Parity** Each "byte" sent or received has some *data bits*, an optional *parity bit*, and one or two *stop bits*. The parity can be odd, even, mark, space, or off. Typical configurations have one stop bit and: 8 data bits/no parity (8N), 7 data bits/parity bit even (7E), or 7 data bits/parity odd (7O). Select these parameters to match the host.

Use the DATA BITS, PARITY, and STOP BITS commands to set nonstandard configurations.

### **1.4.2.3 Modem Parameters**

Select the modem (if your PC has more than one). You may also want to configure the modem, by clicking the nearby Setup button.






Then enter the phone number to be dialed. As soon as you click "OK", Anzio will tell the modem to dial.

## **1.5 Initial Anzio Terminal Screen**




---

By now you should be looking at a blank screen with a *status line* at the bottom. Anzio also has a typical menu bar at the top (File, Edit, View, etc.).

The status line shows status indicators (of course) and some error messages. You can turn the status line on and off with View/Show Status Line or the STATUS LINE command. The status indicator words are:


CAPS	The  key is on – all alphabetical characters will be in UPPER CASE.
NUM	The  key is on – the numeric keypad sends digits, rather than the keycap codes.
LOCK	The keyboard is locked – nothing you type will go through to the host. To send keyboard keys, either the host must unlock the keyboard, or you must use Communicate/Unlock,   , or the LOCK OFF command.
HOLD	The output from the host to Anzio has been suspended, with either the HOLD command or the  key.
AUX	Indicates the aux port is enabled for connection to a serial device, such as a barcode reader.
PAGE <i>n</i>	You are viewing page or screen <i>n</i> in a multi-screen environment.

From the live terminal screen, any “normal” key you press will go out through the communication line to the host computer. Function keys and special keys usually send their own terminal-specific codes. You can define almost any keyboard key to send special character sequences or to call Anzio functions.

The status line can also be used to enter Anzio commands, called with   . We'll get into that later. For now, let's see if we can communicate.

## 1.6 Logging On to the Host System

---

If you are connected to a UNIX (or similar) system, but haven't seen a login prompt, press . You should see the “login:” prompt (or “user” or “username” or ...). Enter your login name and password just as you would with a terminal. Watch out for uppercase vs. lowercase.

After you log on, you will see the operating system's "banner" at the top of the screen. Try entering some host commands, or run a host program, to check the connection.

If something goes wrong at this point, read the next section, *Anzio's Login Wizard*.

To use Anzio menus and enter commands, see *Talking with Anzio*, page 32. The *Command Reference* section, starting on page 92, describes all commands.

### **1.6.1 Anzio's Login Wizard**

For future connections, Anzio can login for you automatically (for most kinds of connections). To do so, it must know your login and password.

Go to Communicate/Login/Username, and enter your username. Go to Communicate/Login/Password, and enter your password. Finally, in Communicate/Login, make sure Login Wizard is checkmarked.

Quit from Anzio, saving your parameters. The next time you run Anzio, it will attempt to login for you.

---

➤ Login Wizard will not work on direct serial connections, for technical reasons. It is disabled on SRP connections, for security reasons. In SSH connections, its use is governed by settings in Edit/Advanced Options/Security.

---

---

➤ The username is stored in the settings file in plaintext. The password is stored in a rather simple encoded format. This may constitute a security risk.

---

### **1.6.2 Troubleshooting**

This section describes some common problems and typical solutions. Note that you can obtain more connection diagnostics by running Anzio in "verbose" mode, with a command-line parameter "/v". See page 186.

### 1.6.2.1 Network Connections

If TCP/IP networking is properly configured, and the host system is accepting connections of the type you're trying, Anzio should connect.

If you get an error message "WINSOCK ERROR 10065", it means your TCP/IP networking is not set up on this PC.

If you get a "Connection refused" or a "Connection timed out" message, your host system is not allowing you in. Check your host name or IP address. Make sure the host allows this kind of connection. Make sure any user limits have not been reached. It is also possible that your connection is hitting a firewall or router that does not let this type of connection occur.

For other network problems, including other Winsock errors, test your TCP/IP network with the "ping" or "tracert" utility, or try running telnet.exe that came with Windows, and connecting to the same host.

### 1.6.2.2 Direct Serial Connections

If you have gotten no response at all, you have a problem with:

- your PORT or BAUD setting (change with Communicate/Setup or the corresponding command),
- your serial port hardware, or
- your cabling.

If you get "garbage" on the screen, chances are that your BAUD rate is incorrect. Try another baud rate, using Communicate/Setup.

If you can't see the characters you're typing, turn off Communicate/Full Duplex, or enter:

   HALF DUP 

If you see two of every character you type, turn on Communicate/Full Duplex, or enter:

   FULL DUP 

If you see "foreign" characters, you may need to change parity and data bits with Communicate/Setup or the corresponding command.

### 1.6.2.3 Modem Connections

If the modem fails to connect, or if you get a “TAPI error”, it can be hard to diagnose. The answering modem may have been busy, in which case you just need to try again. Try dialing the same phone number with a regular telephone – you should hear the modem answer and squeal. If you hear a recording, or a person answers, that will tell you the problem. Or, plug an analog phone into the same line as the modem is on (with a 2-phone adapter, if necessary), and listen during the call.

If you get connected, but then see garbage characters, your host-end modem may not be configured properly. That modem must be configured to talk to the host system at a *locked-in* baud rate, that matches the baud rate of the host’s serial port, regardless of the rate established in the modem-to-modem connection.

### 1.6.2.4 General Problems

If you see intelligible characters, but the character positioning is wrong, the host is expecting a different terminal type. Try a different terminal type with Communicate/Terminal Type.

If the screen looks OK, but function keys do not generate the correct response from the host software, you may need to try a different keys file, with Communicate/Read Keys.

If you’re having trouble seeing the proper non-Latin (Chinese, Russian, Hebrew, etc.) characters, go to Communicate/Character Sets.

## 1.7 The Anzio Window

---

Now let’s take another look at Anzio’s main window. It conforms to standard Windows user interface guidelines.

At the top is a Title Bar, with a System menu icon, a Minimize button, a Restore or Zoom button, and a Close button. You can move the window around on your screen by dragging the title bar, or minimize it or quit it with the appropriate buttons. You can also *zoom* it, but we’ll get to that in a moment.

Below the Title Bar is the Menu Bar, which shows Anzio's menu options. At the bottom is the Status Line.

The remainder of the window is the Active Area, which is used by the host program to display characters. This area is divided (although you can't see it) into rows and columns of character cells. To start with, it shows 80 columns and 24 or 25 rows; this is known as the *screen size*. Every character cell is the same size, and can contain a character in a certain *font size*. As you might image, the screen size and the font size are related to the *window size*, the size of the entire space that Anzio occupies. Understanding how Anzio treats that relationship will help you get the best use out of Anzio.

### 1.7.1 **Zoomed Behavior**

You can *zoom* or *maximize* Anzio's window by a) clicking the Zoom button on the title bar; b) double-clicking the title bar itself, c) hitting the View/Zoom menu item, or d) typing `[ALT] [Z]`. When the window is zoomed, it takes over the entire screen. Anzio will automatically adjust the font size to best display the necessary number of rows and columns in the window space provided.

### 1.7.2 **Unzoomed Behavior**

If you repeat the above procedure, to make the window *unzoomed*, Anzio will make the window the necessary size for to display the needed rows and columns (screen size) at the needed font size. That is, the window size follows the font size. If you change the font size or the screen size (using the menu), the window size will also follow.

If you *resize* the window, by dragging one of its borders, however, Anzio will automatically adjust the font size to follow the window size. This makes it easy to clear some space on your screen while still seeing all the characters displayed by the host.

### 1.7.3 **80-column and 132-column Modes**

Sometimes you will need to display 132 columns of data, such as when previewing a print job at the host system. The host program can send an escape sequence (which varies by terminal type) to switch the "terminal"

into 132-column mode. Or, you can change the screen size (rows and/or columns) by using the View/Screen Size menu item. This will cause Anzio to adjust its screen display as follows.

Anzio remembers two sets of preferences; one for 80-column mode (or less), and one for more-than-80-column mode (such as 132). For each mode, Anzio remembers whether the window was zoomed and, if not, what its font size and window position were.

The first time you see it switch to 132-column mode, it is disconcerting. But go ahead and configure it for how you want 132-column mode to look, by either zooming the window or moving and sizing it. After you have set your preferences for both 80-column mode and 132-column mode, you'll be set.

It is also possible for Anzio to remember all these settings between sessions. Just make sure that, in the View menu, you have Track Window Position checked. When you quit from Anzio, it will ask if you want to save your settings. Answer "Yes".

#### **1.7.4 Other User Preference Items**

It is also possible to set various other things that affect how Anzio looks (and sounds). These are generally found in the View menu. You can change screen and font colors, cursor type, beep behavior, and whether or not the status line or gauge line is displayed. You can also set a picture as the *background bitmap*, which is displayed behind the screen text.

More advanced options are available in Edit/Advanced Options, under the General and Screen tabs.

### **1.8 Exiting from Anzio**

---

To exit Anzio, press the **[ALT]** and **[X]** keys, or use File/Exit or the EXIT command.

If you have changed either the Anzio settings (defaults file) or the defined keys (keys file), you will be prompted to save them. For now, just accept each default file name.

Each time Anzio starts up, it looks for a settings file named AnzioWin.def. If the file is found, Anzio loads those settings. A settings file may include the name of a key file – if so, those defined keys are then read in (see *Loading Key Definitions*, page 51).

You can use multiple settings files, and multiple key files, for different Anzio configurations (for different users or hosts) on the same PC. See *More On Starting Anzio*, page 179.

## **1.9 Configuring Your Host System**

---

To communicate, both Anzio and your host computer must speak the same language. We have already covered most of the hardware configuration, such as the terminal type and communication method. Software configuration tells the host about Anzio's terminal.

This section describes how to set terminal configuration parameters for UNIX systems. For more information, see Appendix B, *Notes On Particular Host Systems*, page 216.

### **1.9.1 UNIX Host Configuration**

On a UNIX system, host terminal parameters are set in up to three files:

- `/etc/inittab` or `/etc/ttytype`  
Sets the default terminal type for each physical connection to the host system. May reference an entry in the file `/etc/gettydefs`, next.
  
- `/etc/gettydefs`  
Terminal ("tty") definitions file for parity, echo, etc.
  
- `~/.profile`  
Optional shell script file in your UNIX home directory. This script can query the terminal for its type, and then call `set tty` (`stty`) commands.



## 1.10 Typical Terminal Types

---

Anzio can emulate many common terminal types. This section describes some of them:

- *VT100, VT220*
- *WYSE 50 and WYSE 60*
- *ADDS Viewpoint*
- *SCOANSI*
- *Terminal Type "Anzio"*

Your host system or application software may support only one particular terminal type. However, if it is flexible, you may be able to choose which terminal type to have Anzio emulate. This section will help you make that choice.

Most terminals support various *character attributes*, which affect the appearance of characters on the screen. Character attributes include reverse video, blinking, underline, intensity, etc. Anzio can emulate these attributes. Some terminal types can also explicitly use colors for foreground (text) and background.

Terminals also vary in the number, names, and placement of special keys, such as function keys, arrows, and "PageUp".

### 1.10.1 VT100, VT220, etc.

The VT100/220 is the basis of the *ANSI standard terminal*. Each screen position can have its own colors and attributes. Unlike some terminals, the VT100/220 colors and attributes do not themselves occupy a screen position. The VT320 and VT420 are based on the VT220, with additional features.

The VT100 has only four function keys and a few special keys. The VT220 (and higher) has more function keys, but they do not correspond well to the PC's keyboard.




Anzio in VT220 mode responds to control sequences for the VT100, VT102, and VT220 (except for some obscure VT220 commands). Anzio also provides some of its own control sequences.

Anzio includes several keymap files to handle different requirements for mapping PC keys to VT keys.

### 1.10.2 **WYSE 50 and WYSE 60**

The WYSE 50/60 is a widely-supported terminal. The WYSE has many function keys, which have a “standard” definition, as well as several special keys.

The WYSE has an advantage in that its arrow keys send single-byte control codes. Some software seems to prefer this, notably “vi” under UNIX.

Wyse terminals can have local key definitions downloaded from the host. These local definitions, if any, take precedence over an Anzio-defined key with the same name. Downloaded key definitions are not normally saved. The KEYS display (   KEYS) will show both definitions, with the local definition on top. See *Defining a Key Macro*, page 48.

The WYSE 50 supports both *field* and *screen* attributes. Each field attribute occupies one screen character position. The screen attribute indicates “protected” fields, but does not occupy a screen character. Anzio supports both field and screen attributes, including protected fields.

The WYSE 60 provides character attributes for each screen position, but the attributes do not occupy a position. This approach saves screen real estate, and is preferred for use with Anzio.

---

➤ Anzio does not support WYSE *page mode*.

---

### 1.10.3 **ADDS Viewpoint**

The ADDS Viewpoint is a commonly-emulated terminal. The original model did not have function keys, so there is little consistency in emulating them. Like the WYSE, arrow keys are single bytes. The ADDS is limited in its attributes, using “tagged” attributes. The host specifies one attribute per screen. Each location on the screen is either tagged or not tagged (one bit). Tagged locations use the given alternate attribute.

#### 1.10.4 **SCOANSI**

When Anzio is set to SCOANSI, it emulates the console of SCO UNIX systems. This is a popular choice because it has full explicit color control, as well as many function keys and special keys.

Anzio emulates the console behavior as of Open Server 5.0.6. However, it can be set to a compatibility mode which causes it to behave like 5.0.5 and earlier.


#### 1.10.5 **Terminal Type “Anzio”**

You can tell your UNIX host that your terminal is really Anzio. The advantages to this are:


- many function keys and special keys are defined
- color is supported
- the host has access to enhanced Anzio capabilities

We have provided some files for use with UNIX and Anzio. Use the following procedure to configure these files:


1. Set Anzio to act as "Anzio", with either Communicate/Terminal Type or the TERM ANZIO command.
2. Load the keys file "AnzioTIC.kys" with File/Read Keys.
3. Upload the *terminfo source file* "Anzio.tic" to your UNIX host, using the *UNIX Simple Upload* procedure on page 84. On the UNIX host, change the name to lower case ("anzio.tic").
4. Tell UNIX to run the *terminfo compiler* "tic" with the "anzio.tic" file. You will need superuser privileges – you may have to ask a local expert for help.

```
tic anzio.tic 
```

5. You will probably get warnings, since this file has entries for both UNIX and AIX. Barring fatal errors, your UNIX host should now consider "ANZIO" or "anzio" a legal TERM type, for any program that uses the UNIX terminfo utility. Tell the host to set its TERM variable to "Anzio":

```
TERM=ANZIO; export TERM 
```

6. Finally, tell UNIX to initialize the terminal:

```
tput init 
```

---

➤ If the host application software does not use terminfo (such as Word Perfect), it will not understand a TERM of “ANZIO”. Rasmussen Software has developed Anzio definitions for popular software – see the README.TXT file, our web site, or contact us for more information.

---

### 1.11 Using Function Keys and Special Keys in Terminal Emulation

---

To use function keys and special keys (such as the arrows) with your host software, Anzio must be set up to send the correct character sequence for each key. Each terminal type that Anzio can emulate has a basic set of key definitions built in. These can be superseded by definitions contained in a “keys” file. You can either load one of the supplied keys files, or you can define the keys yourself (see *Defined Keys and Macros*, page 38).

To load a keys file, use File/Read Keys or the READ command. Also, if you change your terminal type with the Communicate/Terminal Type menu item, Anzio will ask whether you want to load the usual keys file for the new terminal type.

Supplied key definition files, such as VT220.kys, are listed in Appendix E, *Distribution Information*, page 221. Each key definition file has a corresponding “.DOC” file explaining it. You can TYPE the “.DOC” file, read it with an editor or viewer, or print it out.

Note that when emulating one of the VT terminals, you may have to experiment with using the various VT keys files. An actual VT terminal has a keyboard that does not match a PC keyboard exactly, especially in the function keys. Anzio’s online help has additional information.

Note also that the Backspace key is also a frequent point of confusion. It has its own setting in the menu as Communicate/Backspace.

### 1.12 Anzio's Mouse Support

---

Anzio is designed so that mouse actions will do what you would expect them to do, but you can override that. By default (assuming right-hand mouse configuration in Windows):

1. Left-drag will highlight a section of the screen. If you hold the **ALT** key down when you release the button, the highlighted region is copied to the Windows clipboard. The region is copied in three formats: as text, as a bitmap, and as Unicode characters. In the destination Windows program, use Edit/Paste to paste from the clipboard, or Edit/Paste Special to access the different formats.
2. Left-clicking while holding down the SHIFT key will extend the highlight to the point clicked.
3. Left double-click will send to the host the word that is pointed to by the mouse cursor, followed by a carriage return.
4. Right-click will popup a context menu, offering you several options.
5. Scrolling the wheel up will take the screen into REVIEW (scrollback) mode, where you can see what has scrolled off the top of the screen.
6. If the host system tells Anzio, via escape sequences, that it wants to receive mouse actions, Anzio will comply. Anzio supports several mouse-to-host protocols, including some defined by "xterm" and also the DEC VT "Locator" mode.

You can change the behavior caused by various mouse actions, by going to Edit/Advanced Options/Mouse. There you can also change the shape of the highlight that is created when you drag the mouse.

### 1.13 Selecting Text Using Keystrokes

---

In addition to being able to select (highlight) text with the mouse as described above, it is possible to select text with arrow key combinations, *unless* those combinations have been redefined.

All highlighting keystrokes use the **SHIFT** key in conjunction with a special key. The arrow keys move the highlight in the indicated direction. **HOME** and **END** move to the left and right end of the line, respectively.


If you haven't already created a highlight, Anzio will begin by extending the highlight from the current cursor location. Thus the cursor location is one end of the block, and you are moving the other end of the block. You can *switch ends* by hitting the numeric-pad 5 key while holding the **SHIFT**.

If you do **SHIFT** **PG UP**, or you **SHIFT** **↑** past the top of the screen, Anzio will take you seamlessly into REVIEW mode, with the highlight still in effect. You can continue to scroll upwards, and thereby select a large part of your REVIEW buffer. To exit REVIEW mode, hit **ESC**.

## **2 Help When You Need It**

---

Anzio provides two kinds of online help:

First, a Windows help system is included, and is accessed via the Help menu and the Contents item. You can also get help on specific menu items by hitting  when the menu item is highlighted.

Second, a great deal of information is posted on Rasmussen Software's web site, at <http://www.anzio.com>.

Oh, and we do actually answer questions by email or telephone!

### 3 Talking with Anzio

---


Anzio can act as a simple terminal, just sending and displaying characters. To use all of Anzio's features, you will need to enter Anzio commands. Anzio provides menus to access most Anzio commands. The following sections describe:

- *Anzio Menus*
- *Entering Commands*
- *Editing the Command Line*
- *Predefined Function Keys*
- *Using Anzio's Language Support*
- *Using Review Mode to See Previous Data*

#### 3.1 Anzio Menus

---

Anzio provides a set of menus, with menu items for most Anzio commands.

➤ See the online help (Help/Contents) for more information about each menu and item. You can do this by activating the menu, positioning the highlight to the menu item in question, and hitting 

##### File Menu

- Open input and output files, change directories
- Save settings files
- Read, save, and merge defined key (macro) files
- Set up printers and the Anzio Print Wizard
- Print the screen
- Set and clear the background bitmap
- Exit the Anzio program

##### Edit Menu

- Copy and paste screen contents, optionally to a file



#### View Menu

- Set user preference items, such as beep, gauge, status line, scrolling, cursor type
- Set screen colors and font
- Set window title, size, and preferred position

#### Diagnose Menu

- Monitor communications
- Interpret all received characters, including control codes
- Review previous screens of data

#### Communicate Menu

- Send a BREAK
- Unlock the keyboard
- Set host protocol: linefeed or carriage return, duplex mode, lock mode
- Set terminal emulation parameters: name, backspace character, environment variables, Caps Lock
- Set modem parameters: dial-in number, answerback string
- Hang up the modem

#### Transfer Menu

- Set the download directory for incoming files
- Capture the screen to a file, or a printer
- Transmit files, optionally with a trailing end-of-file string
- Set receive mode
- Stop a file transfer, and retransmit
- Send and receive Kermit files
- Send and receive Zmodem files
- Set character and line delay times

#### Help Menu

- Display on-line help contents and topics



- Show Rasmussen Software support contact information, including the Anzio version and copyright, and which Windows communication module is in use

➤ You may customize Anzio using a Windows resource editor to add or remove menu items and accelerator keys. For example, if you remove File/Quit, Alt-X and Alt-x, a user cannot exit a running Anzio.


### 3.2 Entering Commands




---


You can enter any Anzio command in the status line at the bottom of the screen. Anzio displays “FUNCTION:” on the status line when it is expecting a command. There are three ways to get into command entry mode:

 Moves the cursor to the status line and waits at the FUNCTION prompt. Enter any Anzio command, then press  to run the command and return to your “live” screen.

#### Macros/Enter Command

This menu item does the same as  above.

 As above, but also displays an old DOS-style HELP screen, showing Anzio commands and certain current configuration options. When you enter a command, you can see the configuration change. The HELP screen lets you enter a series of commands – to exit the FUNCTION prompt, press an extra  or .

➤ Press  to exit from the FUNCTION prompt without performing any command.

Anzio commands can also be called from:

- A defined key macro – see *Defined Keys and Macros*, page 38.

- The host computer – see *Sending Anzio Commands From The Host*, page 192.

Anzio allows some leeway when entering command names and options:

- Commands may be entered in UPPER or lower case.
- The command name can be abbreviated, to any unique abbreviation. For example, INTERPRET can be called as INT. The only exception is the DEFINE command, which cannot be abbreviated.
- Command options can be shortened, and sometimes omitted. For example, FULL instead of FULL DUP, or DATA 8 instead of DATA BITS 8.
- Spaces are ignored, except as part of defined keys.
- The option ON, such as GAUGE ON, is optional. That is, GAUGE by itself means GAUGE ON.
- Commands having options ON and OFF also accept SWITCH or TOGGLE to change between states. For example, GAUGE SWITCH alternately turns the gauge off and on.
- When a command allows or requires an argument, and you need to enter an argument that contains an embedded space, put double quotes around the argument.
- If you need to embed a double-quote inside a quoted string, put a backslash ("\") before it.

### 3.3 Editing the Command Line

---

You can edit a command while entering it, using the following keys:

#### BACKSPACE

Deletes the character to the left of the cursor, and backs up the cursor.



Moves the cursor right or left.



Switches between *insert* and *overtyp*e mode. In insert mode, displays INS at the right end of the command line.



Deletes the character at the cursor.

<b>HOME</b>	Moves the cursor to the beginning of the command.
<b>END</b>	Moves the cursor to the end of the line.
<b>F1</b>	Delete-to-end-of-line – erases all characters from the cursor to the right end of the line.
<b>F7</b>	Used with the KEEP, PRINT, and PICK functions to define a rectangular area (column) on the live screen – see <i>Data Capture</i> , page 80.
<b>CTRL P</b>	<i>Prefix key</i> used to enter character codes. The next character is included as-is in the command line. For example, to enter an ASCII “escape” character, press <b>CTRL P ESC</b> . Without the prefix, <b>ESC</b> is interpreted by Anzio as an editing command (next).
<b>ESC</b>	Throws away the edited command and returns to the live screen.

---

➤ To include an **ESC** in a command, press **CTRL P** then **ESC**.

---

Function keys entered as part of a command are shown in reverse video with the function key name, such as F5. If a shift, control, or alt key is used with the function key, that key is shown by:

s	<b>SHIFT</b>
^ (caret)	<b>CTRL</b>
a	<b>ALT</b>

Other special keys are displayed by name, or by hex codes.

The line shifts over if you type past the right margin.

### 3.4 Predefined Function Keys

---

Anzio, at its core, predefines several function keys for its own use. For instance, **F6** initiates an Anzio command. However, any of the function keys may be redefined, either with a user macro, or to send a control sequence to the host. This is referred to as *overdefining* the key.







➤ Use **CTRL** **SHIFT** **F** KEYS to see the current function key assignments.

In fact, most function keys are *usually* overdefined. However, their core definitions are listed here because they are sometimes used inside macros, or for other purposes.

If a function key has been overdefined, you can still access its Anzio function using a *modifier key*. For example, if your configuration uses **F3** (that is, **F3** sends a code sequence to the host), you can access Anzio's core **F3** action with **SHIFT** **F3**, **ALT** **F3**, **CTRL** **F3**, or **CTRL** **SHIFT** **F3**. Of course, you may also overdefine these *modified* function key combinations, but we don't recommend it. See *Overdefining Anzio's Default Function Keys*, page 50.

Some of Anzio's default function keys also have an alternative key combination, as listed below.

- F1** "Send defined key". Press **F1**, then a defined key – that key is INVOKEd and the results are sent to the host. See *Executing Key Macros*, page 50.
  
- F3** *Function prefix key*, used for emulating the FUNCTION key on certain terminals. Displays "Func:" on the status line and moves the cursor there. The next key you press is sent in ADD128 format: decimal 128 is added to the character code. An exception is if Anzio is emulating an NCR 7900, in which case Anzio sends a hex 02, then the key code, and then an **ENTER**. Press **ENTER** alone to exit the Func: prompt without sending anything.
  
- F4** or **ALT** **U** Unlocks the keyboard.
  
- F5** or **ALT** **M** HELP key. The DOS-style help screen is displayed, showing Anzio commands, any open files, and the current status of many options (Baud rate, duplex mode, etc.). Anzio commands can be typed in at the bottom of the HELP screen.
  
- F6** or **CTRL** **SHIFT** **F** "Anzio function" – displays the FUNCTION prompt for you to enter an Anzio command.

-  or   PANIC BUTTON (Grand abort)– aborts a defined key in progress, turns TRANSMIT OFF, exits any endless loops, and empties the keyboard buffer.
-  or   BREAK key – sends the BREAK signal to the host computer. Can also be pressed at any time to clear the type-ahead buffer and unlock the keyboard.

### 3.5 Using Anzio's Language Support

---

Anzio is able to handle many kinds of characters, including Latin, Cyrillic, Hebrew, Arabic, Turkish, Chinese, Japanese, Korean, and more, as well as line-drawing characters and many technical symbols. Internally, Anzio stores all these characters in Unicode, a 16-bit coding scheme which defines all these characters. Anzio uses sophisticated techniques, working with routines provided by Windows, to display, print, and accept from the keyboard these characters.

But the host system that Anzio is connected to may not work in Unicode; even if it does, passing 16-bit characters is a problem. So when communicating with the host, Anzio translates characters between Unicode and some *encoding scheme*, also known as a *character set*. Anzio supports many different character sets. All are based on ASCII, where, for instance, hex 41 is always "A". However, the host system may need to store, display, print, and accept as input a character such as "â". There are various standards for coding these non-ASCII characters, and Anzio supports many of them.

For older systems, where it was important to keep characters limited to 7 bits, coding schemes were developed that replaced less used special characters, such as "\", with the accented characters used in, for instance, Norway. These coding schemes are called National Replacement Character sets, or NRCs.

More recent efforts left all the 7-bit codes the same as ASCII, but established standards for 8-bit codes, that is, codes from hex 80 to hex FF (decimal 128 – 255).

Also, schemes have been developed to store characters in languages that have more than 256 characters, such as Chinese, Japanese, and Korean, using more than one byte per character.

In addition to NRCs, Anzio deals with the following character codings:

- ISO - the “Windows” character set, which varies based on the language of your Windows installation.
- DEC – very similar to ISO 8859-1.
- OEM – sometimes called “linedraw” or “DOS”. Corresponds to the DOS codepage on your Windows installation.
- ISO-8859-x, where “x” is 2 to 9 or 15. These are used in Europe and elsewhere.
- ASMO449 – Arabic
- KOI8-R – Russian
- VISCII – Vietnamese
- DOS and Windows code pages, including Far East code pages. You must have installed Windows support for the particular codepage.
- Library of Congress USMARC
- Byte-encoded Unicode (UFT8)
- CCCII for Chinese, Japanese, and Korean characters.
- T160E encodings of diacritics.
- Custom encodings, as defined by the user

---

➤ To use these encodings, you must set the terminal emulation type to VT100, VT220, VT320, VT420, SCOANSI, LINUX, or ANZIO.

---

There are four areas affected by character encodings:

- Characters received from the host must be displayed properly.
- Characters typed on the keyboard, or pasted from the clipboard, may need to be translated before being sent to the host.
- Characters sent from the host as part of a passthrough print job need to be printed properly.
- Characters received from the host and CAPTUREd to a file must be encoded properly.

### **3.5.1 Setting the Host's Character Set**

Click the Communicate/Character Set menu item. This will bring up a dialog box that allows you to choose a "Character set for screen". The "7-bit" selection refers to NRCs, as explained above. If you do NOT need an NRC, as most people won't, select "International".

Under "8-bit", specify one of the sets as described above. If in doubt, choose "ISO".

You can also use the CHARSET command to specify the host's character encoding.

### **3.5.2 Setting Print Encodings**

In order to passthrough-print non-ASCII characters, you must use AnzioWin (not Anzio Lite), and turn on Print Wizard. With any other configuration, coding is determined partly by the terminal type you are emulating, and partly by the printer itself. However, screen prints should still work.

To tell AnzioWin what coding scheme to apply to passthrough print, go to Communicate/Character Set, and make settings in the area labeled "Character set for passthrough print", similar to what was described above.

The printer's encoding can also be set with the PASSTHRU-CP command.

### **3.5.3 Keystroke Entry**

You may need to enter any of a wide range of characters into Anzio, or through Anzio into the host system. Some of these characters, or character-diacritic combinations, may not exist on your keyboard. This section describes the many ways of entering characters into Anzio.

#### **3.5.3.1 *Windows Keyboard Switching***

Your Windows system can be configured to support various keyboards. For instance, you can have both an English (US) and a Russian keyboard driver installed on your Windows, and then you can switch



between them with a special key combination. See your Windows documentation for more information. Anzio will correctly understand keystrokes entered from any Windows keyboard driver.

Note that some versions of Windows have an "on-screen keyboard" available under the "accessibility" section. This feature may help you know what characters are assigned to what keys.

### 3.5.3.2 *Alt-numeric Entry*

Since the days of DOS, it has been possible to enter non-ASCII characters using the Alt key and the numeric pad (NOT the numbers at the top of the alpha keyboard). For instance, alt-144 enters an E-acute (on a US Windows system). This number is taken from the "DOS" or "OEM" character set.

For a character from the Windows ("ANSI" or "ISO") set, precede the number with a zero, such as Alt-0255 for a y-umlaut.

In Anzio, you can also enter any Unicode character by its *decimal* value. For instance, a Y-tilde has a Unicode value of hex 1EF8, which is decimal 7928. So you can do Alt-7928 to enter this character.

Finally, you can enter a Unicode character by its *hex* value, by starting with an Alt-plus. In this one case, you will need to use characters **not** on the numeric pad, for A through F. So for the example above, hold the Alt key and type +1EF8 (where "+", "1", and "8" are on the numeric pad).

### 3.5.3.3 *Dead-key Entry*

Anzio has built-in support for treating certain control-key combinations as dead keys, that will combine with the next character. This is modeled after Microsoft Word. The following table shows each key combination and its meaning. Note that in some cases you must include Shift in order to get the correct character:

Key combination	Meaning	Combines with (examples)
Ctrl-`	grave accent	a e i o u A E I O U
Ctrl-'	acute accent	a e i o u A E I O U d D
Ctrl-6	circumflex	a e i o u A E I O U
Ctrl-:	umlaut	a e i o u A E I O U y space

Ctrl-,	cedilla	c C space
Ctrl-~	tilde	a n o A N O
Ctrl-&	ligature	a s A
Ctrl-/	slash	o O
Ctrl-.	dot below	a b d e A B D E
Ctrl-?	hook above	a e A E
Ctrl-(	breve	a e A E

So for example, if you type ctrl-colon (which is ctrl-shift-semicolon), you will see nothing; then when you type "a", Anzio will get an a-umlaut (ä).

---

➤ There are many more combinations available than are shown here. Any combination defined in Unicode, which involves one of the diacritics above, should be supported. However, you may be limited in what your host system will accept, or you may be limited by the character set you are using.

---

### **3.5.3.4 Input Method Editors (IMEs)**

If you have a Far East version of Windows, such as Korean, your keyboard driver for Korean will be an Input Method Editor (IME). When the IME is active, you go through a several steps to choose each character. Windows 2000 and later have support for all languages that Windows supports. So even if you have an English version of Windows 2000, you can enable support for Chinese, Japanese, and Korean, and obtain their IMEs. Anzio will properly recognize characters coming from all these IMEs.

### **3.5.3.5 Add-on Programs for Far East Input**

Anzio will also work with most add-on programs for Far East input, such as Union Way, WinMass, Twin Bridge, etc. Because Anzio can handle all the text display and printing tasks, these programs are useful only for providing input methods.

It is important to coordinate the encoding scheme that an add-on uses to output a keystroke, with what Anzio expects as input. See "*Editing the DEF File*" page 209.

### **3.5.3.6 COMPOSE and UNICODE Commands**

Anzio's command language includes two commands to aid in character input.

The COMPOSE command is useful to combine a character and a diacritic.

The UNICODE command translates a hexadecimal value to its Unicode character equivalent.

As with other Anzio commands, these can be programmed into macros and used in a variety of ways.

### **3.5.3.7 Paste from Clipboard**

Whenever you paste into Anzio, whether to the host program or to the Function prompt, Anzio will pull Unicode characters from the clipboard if possible. So you can Copy data to the clipboard from various multilingual applications, such as Internet Explorer, Word, or Outlook, and paste it into Anzio. The converse is also true; you can copy from Anzio and paste into other applications.

### **3.5.3.8 Unicode Characters in Macros**

Anzio's macro ("defined key") feature can handle any Unicode character. So it is possible to define macros that send one or more characters to the host system, with proper encoding.

## **3.5.4 Automatic Font Selection**

Anzio's ability to display and print Chinese, Russian, Turkish, and more implies a need to deal with multiple fonts. In fact, Anzio has a unique ability to find appropriate fonts from those installed in Windows.

You configure a primary font; for the screen (in View/Screen font), and for the printer (in File/Printer Font). As Anzio is displaying (or printing), if it encounters any non-ASCII character, it checks to make sure that character is included in the primary font. If it is not, Anzio will search through the fonts installed on Windows to find a font that contains the character, and use that font for this character. This operation is

automatic. So you can go from displaying English to Chinese to Russian to Turkish, with no user intervention. Preference is given to mono-spaced fonts, and to fonts that have the correct width for Far East characters.

If Anzio is unable to find a character in ANY font installed in Windows, it will display that character as a rectangle with an X in it.

You can download and install non-Roman fonts, such as for Far East languages, from Microsoft's web site. To do so, simply point your Internet Explorer to the Internet Explorer update web site at Microsoft, and install the particular language support (such as Korean). This will get you the fonts you need, and they will be available to Anzio the next time you run it. You may also be able to install the fonts through your Windows Control Panel, depending on your version of Windows.

Because diacritic handling is important to many of our customers who are libraries, Anzio also has special handling of character/diacritic combinations. It can handle combinations that are not part of the Unicode standard, by combining them dynamically. When displaying combinations, Anzio goes to considerable lengths to make sure the diacritics are available and properly placed, to always render good results.

### **3.5.5 Changing the Language of the Menus**

You can change the language used in Anzio's menus, messages, and dialog boxes, from the native English to Spanish, French, etc. Go to View/Menu Language, or use the MENULANGUAGE command.

Menu language support is provided in the form of files named ANZIOxxx.DLL, where xxx is an abbreviation for the language name. At the time of this writing, we provide Spanish, Italian, and French in addition to English. If you are interested in creating another translation, please contact us.

### **3.6 Using Review Mode to See Previous Data**

---

Anzio saves incoming data in a special *review memory*, large enough to contain many screens full of data. The review memory is initially blank (filled with spaces). The amount of space in the review memory is configurable; see page 199.

- To start REVIEW mode, press **ALT R**, or use Diagnose/Review or the REVIEW command. Exit REVIEW mode with **ENTER** or **ESC**. The mouse wheel can also take you in and out of Review mode, depending on its configuration.

When REVIEW mode is active, the status line displays

USE UP-ARROW, DOWN-ARROW TO REVIEW SCREEN

Anzio stores in review memory all lines that are scrolled off the top of the screen. In addition, if you turn SCROLL ON (View/Scroll On is checked), any screens “erased” by the host will be “scrolled” into memory instead.

### 3.6.1 **REVIEW Mode Scroll Keys**


REVIEW mode uses only the following keys:






- |                             |  |
|-----------------------------|--|
| <b>ENTER</b> or <b>ESC</b>  | Restores the working screen and exits REVIEW mode.               |
| <b>↑</b> , <b>↓</b>         | Moves the view window up or down one line.                       |
| <b>PG UP</b> , <b>PG DN</b> | Moves the view window up or down one screenful.                  |
| <b>←</b> , <b>→</b>         | Moves the view window left or right one column, if possible.     |
| <b>HOME</b> , <b>END</b>    | Moves the view window to the far left or far right, if possible. |
| <b>P</b>                    | Prints the current screen.                                       |


## 4 Defined Keys and Macros

---

Anzio lets you assign characters, strings, and commands (macros) to almost any keyboard key. When you press a *defined key*, Anzio looks up the key definition and processes that definition as if you just typed it in. This feature is used to establish what sequences special keys send out, as well as to store commands into macros for various purposes.

For example, if you frequently have to type the host command “DEALLOCATE”, you could assign it to the  key with

   DEFINE  DEALLOCATE 

The next time you need to enter that command, just press , and Anzio will send the characters as defined.

---

➤ Earlier versions of Anzio had a limit of 246 keystrokes for each defined key. That limit has been removed.

---

You may also include most Anzio functions in a defined key, as described below. While a defined key is running, it can display strings to the user (*prompts*) that are not sent to the host, and can wait for the user to enter strings or select filenames.

After defining one or more keys, you can save the current set of key definitions to a file. You can read in or merge other key definition files, either replacing or adding to the current set.

Use the KEYS command to see the current defined key sequences:

   KEYS 

The KEYS display shows all system-assigned keys and user-defined keys. Each defined key is displayed in reverse half video (or the color equivalent), followed by its contents, ending with a reverse half video space. Non-display characters are shown in reverse video by name or as an ASCII hex number (01 for a *control-A*, etc.).

The following sections describe:

- *Which Keys Can I Define?*

- *Defining a Key Macro*
- *Editing a Key Definition*
- *Executing Key Macros*
- *Overdefining Anzio's Default Function Keys*
- *Using Key Definition Files*
- *Special Characters Inside Defined Keys*
- *Using Anzio Functions in a Defined Key*
- *Nesting Defined Keys*
- *Special Uses of Macros*

#### 4.1 Which Keys Can I Define?

---

There are three general categories of keys that can be DEFINEd.

- All printable ASCII characters, from space (hex 20) to tilde (hex 7E). Definitions for printable keys are NOT triggered by simply pressing that key; other steps are required as described below.
- Special keyboard keys, including function keys, named keys such as **HOME** and the numeric pad period **.**. The numeric pad keys are only accessible with **NUM LOCK** off, except that separate definitions are possible for numeric pad **/** **\*** **-** **+** **ENTER** with **NUM LOCK** on.
- ASCII control codes, hex 00 through hex 1F.


Function keys **F1** through **F10** can be defined four ways each: normal, with **SHIFT**, with **CTRL**, and with **ALT**. Also, if your system supports keys F11 and F12, Anzio will support them also, in all four states.



---

➤ Anzio treats the **CTRL** **SHIFT** combination as synonymous with **ALT**. So for instance, if you define **CTRL** **SHIFT** **F1**, it will be recorded as **ALT** **F1**, and you can press either key combination to invoke it.

---

The **NUM LOCK** key can be defined if necessary. Most often, this is done in order to make the top four keys on the numeric pad emulate keys PF1 through PF4 of a VT terminal, for users who are accustomed to that. To allow the **NUM LOCK** key to be programmed, go to Edit/Advanced options (see page 201). To load standard PF1 to PF4 key definitions, go to File/Merge keys, and select the file "VT-NUM.KYS". Note that setting up

Anzio in this configuration often leads to user confusion over the operation of the  key.





- It is possible to define a keystroke combination that is normally used as a *menu accelerator* key, such as   which normally does a Zoom. To define the key, temporarily disable accelerator keys by turning off “Allow accelerator keys” in Edit/Advanced Options/Keys. After defining the key, you can turn that setting back on, because defined keys take precedence over accelerator keys.

---

## 4.2 Defining a Key Macro

---

Key macros are defined with the DEFINE command:

   DEFINE *x* *string* 

where *x* is the key to be defined, and *string* will become its contents. There must be one space between DEFINE and the key *x*, and another space between the key and the definition *string*.

When the macro *x* is executed, characters in *string* are, by default, sent to the host system, unless they fall into one of the special cases below.

- If you redefine an existing defined key, the previous definition is deleted without a warning.

### 4.2.1 Undefining a Key






You can define a key with a null definition, to override a previous key definition. Enter nothing for *string*:

   DEFINE *x* 







If you later execute key *x*, the default code for that key will be sent to the host. You can also *block* a key, so that it sends nothing.







### 4.2.2 **Blocking a Key**

You may need to block some keys, so that an Anzio user can't accidentally send those character codes to the host. For example, , , ,  , and so forth.

To block a key, define it with “do-nothing” empty braces {}:

   DEFINE   {} 

If you later press  , nothing happens – the   keycode will not be sent to the host.

The empty braces are a special case of a “user prompt” – see *Displaying Text to the User*, page 54.


### 4.3 **Editing a Key Definition**

---

While you are entering a defined key, you can use command-line editing, as described in *Editing the Command Line*, page 35.

To return to an existing key definition and make changes, use DEFINE with a question mark:


   DEFINE x?

When Anzio sees the question mark after the key (without a space), it displays the existing contents of the key. You can then edit the definition. Press  to finish editing.

#### 4.3.1 **Copying a Defined Key**

To copy a definition from one key to another, begin editing the existing key, as just described, then back up and change the key name. For instance, to copy key X's definition to key Y, enter:

   DEFINE X?

When Anzio displays the current definition of X, back up and replace X with Y, then press .

---

#### 4.4 Executing Key Macros

---

To execute a key macro defined for a regular printable key, press some form of `F1` then the key, for example, `ALT F1` then `X`. This assumes `ALT F1` has not been redefined.

To execute a key macro defined for a function key or control character, just press the key(s), for example `F10` or `CTRL X`.

You can also execute a macro with the INVOKE command:

```
CTRL SHIFT F INVOKE X
```

---

#### 4.5 Overdefining Anzio's Default Function Keys

---

Anzio's predefined function keys, such as `F5` for Help and `F6` for Function, can be *overdefined* with your own definition:

```
CTRL SHIFT F DEFINE F5 . . .
```

In fact, for most terminal types, these function keys WILL be overdefined. For this reason, Anzio predefines all variations of its function keys as well: `ALT F5`, `CTRL F5`, and `SHIFT F5` will all display the Help screen unless overdefined.

You can also use `CTRL SHIFT F` for `F6`, and `ALT M` for `F5` – see *Predefined Function Keys*, page 36.

---

➤ We strongly advise you not to redefine `ALT A`, `CTRL SHIFT F`, `ALT M`, `ALT U`, or `ALT X`.

---



---

#### 4.6 Using Key Definition Files

---



When Anzio starts up, it can read a *key definition file*. The key filename is contained in the settings file.

If you want Anzio to start up without reading any key file or settings file, use the command line parameter NONE – see *'NONE' For Clean Startup*, page 182.

When a key file is read in, the definitions in the file are added to any key definitions in memory. A key definition from the file will replace an existing definition having the same key.

Anzio remembers the name of the last key file read in, either during startup or later. This key file name can be saved in Anzio's settings file for use the next time.

---

➤ If you change a key definition, then exit from Anzio using File/Exit, Windows Close (X), END, STOP, or  , Anzio will ask if you want to save the changes to the default key file.

---

#### 4.6.1 Saving Key Definitions

To save the current set of defined keys in a key file, using File/Save Keys or the SAVE command:

   SAVE *filename* 

All current key definitions are saved in *filename*, including your new definitions, Anzio's default defined keys, and any previous definitions.

The SAVE command without a filename saves the current key definitions to the default key file.

#### 4.6.2 Loading Key Definitions

There are two ways to load key definitions from a key file: *read* and *merge*:

- When you **read** in a key file, all existing definitions are erased before loading the new definitions. Predefined keys will be kept unless overridden by new definitions.
- When you **merge** in a key file, the new definitions are added to the existing ones. A merge will overwrite an existing definition with the new one only if they both have the same key.

To read in a key file, use File/Read Keys or the READ command:

   READ *filename* 



To merge in a key file, use File/Merge Keys or the MERGE command:

  **F** MERGE *filename* 


---

## 4.7 Special Characters Inside Defined Keys



Some characters and keys have special meaning in a defined key:

- Tabs – see *Tabs*
- | vertical bar – see *Using '|' For* 
- # number sign – see *Waiting for User Input*
- {} braces – see *Displaying Text to the User*
- ~ tilde – see *Quoting Special Keys With '~'*
-  call function – see *Using Anzio Functions in a Defined Key*


### 4.7.1 Tabs

Tab characters may be included in a defined key by simply pressing the Tab  key. A tab is displayed as a reverse video “TAB”, but is just one keystroke in the defined key. When the defined key is sent, the tab is processed according to the current tab settings, just as though you had typed it.





### 4.7.2 Using '|' For

When you are entering a command from the keyboard, a carriage return  is required to end the command and begin processing. In a defined key, use the vertical bar character '|' to represent an end-of-command .



For example, to define key E to get you into an I-system text editor, enter the following (as one line):

```
DEFINE E AS EWF EWF(3) |  
        AS A TEXTFILE(2),OW |  
        AS LO (LP) |  
        EX $EDIT,RE | 
```


After this definition, press   **F1** E and Anzio will send to the host:

```
AS EWF EWF(3)   
AS A TEXTFILE(2),OW   
AS LO (LP)   
EX $EDIT,RE 
```

### 4.7.3 Waiting for User Input


In a defined key, the number sign '#' causes Anzio to stop for operator input. When this character is found while processing a macro, the Anzio program will stop transmission, accept your entry up to a  and send the entry (the  is not sent).


Transmission of the key string continues with the character following the '#'. For example, to create a key sequence to copy files on an I-system (see UNIX example later):


```
DEFINE C AS A #(3) |  
        AS B #(1),NE,300,AP |  
        MOV A B | 
```

Then, when you enter    C, the PC will send to the host:

```
AS A
```

Then it will wait for you to enter text (a file name). Any key you enter will be sent, until (but not including) the . The "defined key" then takes over, and sends:

```
(3)   
AS B
```

and waits for another filename. Your filename is sent to the host. When you enter , the defined key continues with


```
(1),NE,300,AP   
MOV A B 
```

The PC is then finished transmitting its defined key sequence, and returns to normal text entry.

#### 4.7.4 Displaying Text to the User


To display, but not send, text in a defined string, enclose it in {curly braces}. When a left brace { is encountered during transmission, all characters up to the right brace } are displayed on the PC, but not sent to the host. The special defined key characters '|' and '#' are not processed inside curly braces, just displayed on the screen.

A typical use of displayed text is to place prompts on the screen for use with user input (#), described above. For example, to define C as an I-series "copy" key, enter this on one line:


```
DEFINE C AS A {Source filename}#(3) |
      AS B {Destination name}#(1),NE,300,AP |
      MOVE A B | 
```

➤ You may include escape sequences, such as for cursor movement, in locally displayed text.

Here is a UNIX example, defining the v key to call the vi editor with a user-entered filename:



```
DEFINE v vi {Enter file to edit}# | 
```

When you enter   v, Anzio sends:

```
vi 
```


and then displays on the PC screen:

```
Enter file to edit
```

After you enter a file name and press , Anzio sends the filename and a  (for the | at the end of the key definition).

#### 4.7.5 Quoting Special Keys With '~'

Use a tilde ~ to "quote" a special character. The character after a tilde is transmitted, not interpreted. So, to program a key to send {A}, enter:

```
DEFINE a ~{A} 
```


The tilde causes the left brace to be sent, rather than displaying “A” as a user prompt. You don’t need to quote the right brace, because Anzio did not “see” the starting left brace and so is not expecting an ending brace.

#### 4.7.6 **Non-ASCII Characters**

Defined keys can contain any Unicode character, including combining diacritics, Cyrillic, Chinese, etc. These will be encoded properly, according to the host’s Character Set, when they are transmitted to the host. Characters can be pasted in from the clipboard, or can be entered using the numeric pad as described in *Alt-numeric Entry*, page 41




### 4.8 **Using Anzio Functions in a Defined Key**




---

Inside a defined key, you can call most Anzio functions with  *functionName*.

Two useful functions are DIR/S and MENUBAR – see *File and Screen Selection Functions*, page 56.

For instance, if every day you had to open a file named UPFILE and send it to the host, you could enter:

```
DEFINE S  OPENI UPFILE |  TRANSMIT | 
```

- Each embedded  function must end with a vertical bar |, as shown, to represent the end-of-function .
-  is recognized inside a defined key, even if that function key is overdefined – see *Overdefining Anzio’s Default Function Keys*, page 50.

You can also use the “wait for user entry” special character # in an embedded function inside a defined key. For example, if you wanted to enter a filename (instead of UPFILE, above):

```
DEFINE S  OPENI # |  TRANSMIT | 
```




#### 4.8.1 Keystrokes For Another Function

The SEND command reads characters from a screen line and returns those characters to a function or to the host. For example, to open for input a file whose name is in the first 12 characters of screen line 17:

```
DEFINE O  OPENI  SEND 1 12 17 17 | 
```

Several of Anzio's functions can also take keystrokes from a defined key. For instance:

```
DEFINE   REVIEW | 
```

defines the  key to start REVIEW mode (" REVIEW|"). The first key that REVIEW sees is its command key , telling REVIEW to show the prior screenful.

#### 4.8.2 File and Screen Selection Functions

Anzio provides two selection functions for use inside a defined key, DIR/S and MENUBAR:

- |         |  |
|---------|--|
| DIR/S   | Displays a directory browsing dialog for the user to select a file. Normally, DIR/S sends its result to the host (indicated by "/S"), but inside a defined key, the result is passed back as user input. |
| MENUBAR | Sets up a selection grid on the screen, with one or more columns. The user selects a grid location, and the characters on the screen at those positions are passed back as user input.                   |

For example, if the host has displayed a list of file names, or a table of data, MENUBAR lets the user select an entry right off the screen.


##### 4.8.2.1 *DIR/S Example*

If you need to transmit a particular type of file frequently, you may define a macro to first select a file, then transmit it:






```
DEFINE O OPENI  DIR/S *.DAT|#||  
 TRANSMIT | 
```

In this case, the filename OPENI is expecting comes from the DIR/S command. DIR/S presents a directory of all files matching “\*.DAT”. When you select a file, DIR/S returns the name to OPENI. Next, TRANSMIT sends the opened file.


Note the characters after the DIR/S command: The first vertical bar is the end of the  DIR/S command, and the # tells Anzio to wait for operator input. The second vertical bar is the end of the # command, and the last vertical bar is the end of the OPENI command.

#### 4.8.2.2 *MENUBAR Example*

Suppose that the host has displayed a list of filenames on the screen, and you want to receive one of these files. If you know that the host will display the filenames in columns 1-13, you can call MENUBAR in your key definition:

```
DEFINE K  KRECEIVE  
 MENUBAR 1 13 1 23 0 0 1 13|#||  

```

The KRECEIVE command’s input filename comes from the MENUBAR function. The numbers are 1 13 for the start and end column, 1 23 for the start and end lines, 0 0 for no second column, and 1 13 for the start and end returned character offsets.

Note the characters after the MENUBAR command: the first vertical bar is the end of the  MENUBAR command, and the # tells Anzio to wait for operator input. The second vertical bar is the end of the # command, and the last vertical bar is the end of the KRECEIVE command.

### 4.9 Nesting Defined Keys

---

You can use the CALL and INVOKE commands inside a defined key to start another defined key. CALL X starts defined key X, and returns to the original defined key when X is done.

INVOKE X starts defined key X, abandoning the original defined key. When X is done, control returns to the keyboard.

## 4.10 Special Uses of Macros

---

There are some special ways macros (defined keys) can be used.

DEFINE the macro as described above. For simplicity in this section, we will assume that these macros are attached to alphanumeric keys.

Remember that macros get saved in the keys (.KYS) file, so be sure to save keys.

### 4.10.1 Startup Macro

Anzio can run a designated macro at startup, in order to automate the process of logging in and doing additional commands.

➤ When a startup macro is used, the Login Wizard is not invoked.

You can specify a particular macro to always run on startup, by configuring that in the settings file. To do this, go to Edit/Advanced Options/Keys, clicking on Startup macro, and entering the letter of the macro in the box. Then quit, saving settings. The next time Anzio is run, the macro will begin executing as soon as the connection with the host is established.

Alternatively, you can specify a start macro in the command line that invokes Anzio, whether that is typed manually, initiated by a start menu item, or initiated by a desktop icon. Simply include the letter of the macro as a command line parameter. See *Single Character – Start Defined Key*, page 182.

A typical startup macro would do a WAITFOR for a login prompt, send the username, WAITFOR the password prompt, send the password, WAITFOR a particular prompt, send a command, etc.

### 4.10.2 Onerror Macro

You may need to have some error handling in your macro processing. For instance, if Anzio is running a macro that opens an input file, but that file does not exist, you may want Anzio to handle that cleanly. For this purpose there is the ONERROR macro.

You define this macro with the DEFINE command, as with any other macro. Let's assume it's the "E" macro. Then use the ONERROR command to designate it as the ONERROR macro:

```
ONERROR E
```

Of course, this command could be part of another macro. Now start your main macro. At any point that a command fails, ALL nested executing macros will be terminated, and the "E" macro will be invoked. The ONERROR macro remains in effect until it is cancelled or replaced.

Some examples of commands that can trigger ONERROR:

WAITFOR (with timeout)	Times out
MKDIR	Illegal directory name
CD	Directory not found
LAUNCH	Can't start program
Various OPENS	File not found, etc.
COPY	File not found
RENAME	File not found, target exists

#### **4.10.3 Code Activated Switch Macros**

Anzio's passthrough print can be configured so that receipt of certain control codes, in the print data stream, triggers certain actions. More specifically, a particular control character would trigger a particular macro, which might flush a print job, switch to a different printer, and begin printing again on the new printer.

A range of 6 macros is associated with 6 control characters, control-A through control-F. The PRINTMACRO command specifies what the *first* of these 6 macros is; that is, which one is associated with control-A. For instance, if you did:

```
PRINTMACRO a
```

then receipt of a control-A would trigger the "a" macro, control-B would trigger the "b" macro, etc., up to "f".

When Anzio's passthrough print routine detects a triggering control character, it quits from passthrough print and then CALLs the appropriate macro.

## 5 Using A Modem to Call the Host

---

You may need to connect to your host system using a modem. Anzio uses Windows drivers to talk to modems, and should be able to work with any modem that your Windows system supports.

➤ Note: The old technique of talking to the modem as a serial device is no longer recommended.

The following sections describe:

- *Types of Modem Connections*
- *Making the Modem Connection*
- *Signing Off*
- *Modem Setup*
- *Troubleshooting Modem Connections*

### 5.1 Types of Modem Connections

---

It is important to understand the type of modem connection you need. After reading the next two subsections, if you are still unsure, check with whoever supports your host system.

#### 5.1.1 Direct Dial Modems

This is the “old school” of modem connections. A modem on your PC dials a modem on the host system. Once that modem answers, you are connected to the host computer system via a serial connection, and typically you get a login prompt. If this is the kind of connection you need, this is the right part of this manual.

#### 5.1.2 Network Modem Connections

In this approach, the networking software on your PC dials a modem either a) connected to your host system’s network, or b) at an Internet Service Provider (ISP). This establishes a network-over-modem connection, and your PC becomes part of the network at the other end, possibly part of the Internet. Once connected, you can run a variety of

network protocols over the modem connection (including a telnet connection). The term “PPP” applies to this kind of connection.

➤ If this describes the connection you need, then this chapter is NOT relevant. Instead, you would want to establish the connection BEFORE you run Anzio, and then tell Anzio to connect via a network protocol, such as telnet, rlogin, or SSH.

---

## 5.2 Making the Modem Connection

---

In Anzio’s Communicate/Setup dialog, choose “Modem” as your type of connection. If more than one modem is listed, select the one you wish to use. Enter the phone number to be dialed. Click “OK”. Anzio should dial the modem, establish a connection with the host-end modem, and present a login prompt as generated by the host system.

## 5.3 Signing Off

---

Signing off is the process of disconnecting the modems, and “hanging up” the phone line. Either modem can initiate the process, depending on the host operating system. When one modem hangs up, the other will detect that and hang itself up, notifying the computer to which it is connected.

To end the session from the host, enter a host command such as “logout”, “exit”, or “BYE”. On most operating systems, the host then tells its modem to hang up. When *your* modem detects the hangup (“carrier drop”), it hangs up, and then Anzio behaves as indicated in the Communicate/On Line Drop setting. Anzio can quit, redial, or stay running in a “not connected” state.

To hang up the modem from your end, use the Communicate/Hangup menu item or the HANGUP command.

## 5.4 Modem Setup

---

If it is necessary, you can exercise more control over the modem on your PC. In the Communication Settings dialog, click the Setup button next to the list of modems. This brings up a Windows standard “Modem

Properties” dialog. Following are some notes on items you might want or need to change.

#### **5.4.1 Modem Connection Speed**

Ordinarily, the two modems used in the connection will negotiate a maximum possible speed, for the greatest throughput. However, you may need to force a lower speed. You can **try** to do this by setting “Maximum speed” to a lower value. However, note that many modems, especially so-called Winmodems, do not obey this setting. You can determine what connection speed is actually established by running Anzio with the “/V” parameter. See page 186.

#### **5.4.2 Data Bits and Parity**

In the Connection tab, you can set the data bits, parity, and stop bits for a modem connection. For instance, if you are dialing in to a host system that expects a connection with 7 data bits, even parity, set it here (setting it on the serial port will have no effect).

#### **5.4.3 Error Control**

Under the Advanced button, you can enable or disable error control. We recommend this be enabled. If enabled, you can also enable or disable compression. Compression will give you higher throughput, but may be reliable only if flow control is set correctly.

#### **5.4.4 Flow Control**

Also under the Advanced button, you can enable flow control, and specify hardware or software flow control. We recommend this be enabled, and we have a slight preference for hardware flow control. Note that this setting governs the flow control between your PC and its modem. There can also be flow control problems at the host end. See below.

## 5.5 Troubleshooting Modem Connections

---

There are, of course, many things that can go wrong with a modem connection. We'll try to offer some tips here on the most common ones. Several of these make use of Anzio's "/v" runtime parameter. See page 186.

### 5.5.1 Failure to Establish Connection

If Anzio fails to connect your modem to the host-end modem, it will give you a message box indicating, for instance, that it received a busy signal. If that doesn't give you enough information to troubleshoot the problem, there are a couple other things you can do.

You can run Anzio with the "/v" mode to get more information on the status codes returned by the modem interface.

You may also want to listen to the phone connection. Plug a two-phone adapter somewhere between the modem and the wall jack, and plug a standard (analog) telephone into it. Then listen as the modem tries to connect. This will enable you to hear a recorded message that says you have to dial an area code, for instance.

### 5.5.2 Baud Rate Shift

If the modems connect, but then you see only garbage characters on Anzio's screen, you may have a problem of baud rate shift. This is caused by an incorrect configuration of the *answering* (that is, the host-end) modem. This problem can appear when a newer, faster modem is first used to connect to the host's modem.

The host-end modem has to know what baud rate to use when talking to the host machine. It SHOULD be configured to always talk to the host at a particular baud rate. The PROBLEM is when it's configured to talk to the host at whatever baud rate the modems link up at. The latter is how modems are often configured at the factory.

For instance, suppose the host system's serial line is configured for 19,200 baud. A modem is connected to that line, and people have succeeded for quite some time in connecting at 19,200. Now, you have a brand new PC, with a built-in modem that is capable of 56k. You dial

in, and the two modems establish a link at 38,400. The host-end modem, being improperly configured, starts trying to talk to the host at 38,400. But the host port is still working at 19,200. So there is a baud rate mismatch, and you see garbage.

The proper solution is to configure the host-end modem to always communicate with the host at a fixed baud rate, the same rate as the host's serial line is configured for.

### **5.5.3 Flow Control**

Flow control (also called handshaking) is a protocol by which one party to a conversation can say "hold on a minute", until it's ready to receive more data. In general, there are two kinds of flow control. Hardware flow control makes use of separate wires in the serial interface, such as the Clear-To-Send (CTS) wire. Software flow control uses certain characters (XON, XOFF) to start and stop data.

There can be issues of flow control in both directions between the PC and the host. Most often, though, it will be a problem in data coming from the host to the screen, especially when a large amount of text is sent without pausing. There is an even greater chance of flow control problems coming to light when doing passthrough print, because the host can push out data much faster than the printer can print it. To restate, you may have a problem in flow control but it will show up only when you do passthrough print or display large amounts of data.

On a modem connection, there are really three segments: host-to-modem, modem-to-modem, and modem-to-PC. The modem-to-modem connection is handled by the error-correcting protocol of the modems, so it is generally not a problem. The other two segments are actually independent of each other.

In the host-to-modem segment, the flow control (handshaking) configured in that modem must be coordinated with the flow control used by the serial port driver of the host's operating system. For instance, if the host is running Unix, then the "stty" settings must specify the same kind of flow control as is configured into the modem.

In the modem-to-PC segment, Windows does a good job of coordinating the kind of flow control between the modem and Windows itself. The



kind of flow control is chosen in the Advanced part of the Modem Properties, as mentioned above.

## 6 Printing

---

Anzio can do several kinds of printing:

- printing the current screen or window
- printing incoming text as it is displayed (*capture to print*)
- printing incoming text without displaying (*passthrough print*)
- sending an incoming file transfer directly to the printer
- printing a file from the PC's disk

The following sections describe:

- *Print Levels*
- *Choosing a Printer*
- *Setting Printer Options*
- *Choosing A Font*
- *Printing The Screen*
- *Capture To Printer*
- *Passthrough Printing*
- *Secondary Passthrough Print*
- *File Transfer To Printer*
- *Printing Files*
- *End-of-Print Flush Timer*
- *AnzioWin's Print Wizard*
- *Print Initialization Files*
- *Troubleshooting Printing Problems.*

### 6.1 Print Levels

---

Anzio can use several methods of writing to the printer. These methods are referred to as *print levels*. Different print levels are required for different printing situations, especially during passthrough print.

In some situations, the host will send out just plain text, with returns and linefeeds, but no escape sequences. In that case, Anzio can use the Windows printer interface to “draw” the text on the page, in the correct orientation, font, height, width, spacing, margins, etc.

In other situations, the host application knows what kind of printer is connected, and it wants to be in control. It includes escape sequences in the print data, to control the printer. In this case, Anzio must stay out of the way, passing on the data transparently.

This is an example of the need for different print levels. Following is an explanation of each Print Level that Anzio offers. These can be selected with the File/Print Level menu item, or with various commands.

---

➤ AnzioWin (but not Anzio Lite) has an extra feature called Print Wizard, that will in most cases decide automatically on the print level. Print Wizard is explained below.

---

### **6.1.1 High**

In high level, Anzio is in charge of layout. Various properties you have selected in File/Printer Setup will be in effect, including paper size and orientation, paper source, output bin, etc. Also, the selections in File/Printer Font will be used. Anzio will determine line spacing, lines per page, etc.

If the print level is high, and the data coming from the host computer contains printer control codes, the codes will be drawn on the page instead of being obeyed. Also, if a host program (such as Word Perfect for UNIX) is generating PostScript code during passthrough print, the PostScript code will be printed on the page, rather than obeyed. These are indications that you need to switch to a different Print Level.

### **6.1.2 Low**

We no longer recommend low level.

In low level, Anzio uses the printer driver, but uses a “passthrough” capability within the driver. Escape sequences are obeyed. Some

printer setup settings will be obeyed. The font selection may be obeyed; which may cause problems (if a non-printer-resident font is selected).

### 6.1.3 **Spooler**

Spooler level is the recommended level when data contains escape codes.

In Spooler level, Anzio sends print data directly to the Windows spooler. Anzio and Windows are completely transparent, so escape sequences from the host are obeyed by the printer. But settings under Printer Setup and Printer Font are ignored.

In this level, if the data from the host does NOT contain escape sequences to configure the printer, then formatting (font, size, linespacing, etc.) will be up to the printer. This may mean the formatting is affected by the last job that was printed. If the printer has front panel controls, you should be able to control formatting there.

The Windows printer driver is not used, so it does not matter what driver is installed. The printer setup is used only to get the device name of the printer. However, the spooler settings in the Advanced part of the printer's control panel WILL have an effect on spooling.

It is possible to have a *different* printer be used for Spooler level than for High level. This is done with File/Print Level/Spooler Setup. If you choose a different printer as the Spooler printer, then switching the Print Level will also effectively switch you from one printer to another.

### 6.1.4 **Raw**

Raw level is as low as it goes. In Raw level, output is sent directly to a device (or file), with appropriate handshaking and timeout management. Like Spooler level, Raw level is completely transparent, and so Anzio exerts no control of formatting or other printer settings.

The Windows printer driver is not used, but is referenced to determine the port location of the printer.

It is also possible to have a *different* printer/device used for Raw level than for High level (or for Spooler level). This is done with File/Print

Level/Raw Setup. Here you can enter the name of a file or device, such as "LPT2" or "COM3". If you specify a COM (serial) device, you will be prompted for the baud rate and other communication parameters.

Once you have specified a *port name* in Raw Setup, you do not need to have a driver installed for that port. In fact, if a driver is installed, it may interfere with printing. In Windows NT and later, the driver will intercept the raw-level output and route it to that printer driver's spooler. This can effect the timing of your printing output. You can address this by a) changing the spooler settings in the Advanced area of that printer driver; or b) deleting the printer driver.

---

➤ Raw printing will not work for printers attached by certain methods, such as USB, USB-to-serial adapters, network interfaces, or Novell servers.

---

## **6.2 Choosing a Printer**

---

Anzio lets you select a primary printer, using File/Printer Setup. Normally, this is used for all levels of printing. However, it is possible to specify a different printer to be used for Spooler-level printing, and a different printer to be used for Raw-level printing. When configured for different printers, changing the Print Level also changes what printer you will output to.

### **6.2.1 The Primary Printer**

To select the primary printer output device, use File/Printer Setup, or the CHOOSEPRINTER command. The printer can be connected to another PC on a network, connected directly to a network, or connected directly to the PC via a parallel port, a serial port, or a USB port. Anzio simply sends data to the selected printer name.

The primary printer can generally be any printer that has a Windows driver, including "Windows-only" printers.

### **6.2.2 The Spooler Printer**

To specify a different printer to be used when Print Level is Spooler, use File/Print Level/Spooler Setup, or use the SPOOLER command. The printer is identified by its Windows name.

### **6.2.3 The Raw Printer**

To specify a different printer to be used when Print Level is Raw, use File/Print Level/Raw Setup, or use the RAW-SETUP command. The printer is identified by the name of its port, such as LPT1 or COM2. If a serial port is specified, you will be prompted for the baud rate and other communication parameters. These must match how the printer is configured.

You can also specify a disk file name, in which case Anzio will write the printout to disk. Note that if a file of the specified name already exists, it will be overwritten with no warning.

## **6.3 Setting Printer Options**

---

➤ The settings in Printer Setup apply only when Print Level is High.

The File/Printer Setup menu in Anzio brings up the standard Windows printer selection dialog box. Use the dialog's Properties or Options buttons to change print quality and other printer-specific settings.

The next time you print something, Anzio will use the specified printer settings. Unlike many Windows programs, Anzio does NOT pop up the Printer Setup dialog every time you want to print.

Anzio saves changes to your printer settings in its settings file (normally "AnzioWin.def"). These settings will override settings made at the Windows control panel level.

## **6.4 Choosing A Font**

---

➤ The font settings apply only when Print Level is High.

Use File/Printer Font to choose a font and size for printing.

You can use any font on your PC, but some will work better than others. Fixed-width fonts, such as Courier New, will line up properly in columns. TrueType fonts will give you more options for size. Printer-resident fonts will generally print faster than TrueType fonts.

We recommend "Courier New" for most applications. However, if you are printing to a dot matrix printer, this may make your printing painfully slow. In this case, choose a font that is resident in your printer; these will generally have a printer icon beside them in the font selection dialog. For instance, in an Okidata 320, you might choose "Courier 10CPI". You might need to choose different names in order to get different sizes.

If Anzio encounters international or other special characters that are not in the font selected, it will attempt to find an appropriate font, for just those characters.

The font size you choose will directly affect horizontal spacing (therefore characters per line) and vertical spacing (therefore lines per page). An 80-column page should fit with 12 point text, or 10 characters per inch (*cpi*), on an 8-inch wide page. For a 132-column page, you will probably want to change your font size down to 7 point.

---

➤ The *AnzioWin's Print Wizard*, page 74, can be set to scan incoming printer text, then fit the text to the page automatically.

---

## 6.5 Printing The Screen

---

To print the currently active screen, use File/Print Screen, **ALT P**, **PRTSC**, or the PRINT command. If you have highlighted a portion of the screen using the mouse, only the highlighted text will be printed.

The PRINT command can also print any rectangular region on the screen. Use **F7** to select the printed region, or enter the row and column addresses of the corners.

---

➤ Printing the screen does not automatically eject the page – see *End-of-Print Flush Timer*, page 73.

---

- To print any buffered printer data and eject the page, use File/Eject, or the EJECT command.
- 

## 6.6 Capture To Printer

---

Anzio can send incoming screen text to the printer as it arrives (*capture to print*). You can start and stop print capture at any time, without affecting the host.

To start capture to print, checkmark the menu item Transfer/Capture to Printer, or execute the command CAPTURE WPRN. To stop, uncheck menu item Transfer/Capture to Printer, or use CAPTURE OFF.

## 6.7 Passthrough Printing

---

Anzio supports *passthrough print*, also called *transparent print*, *local print*, or *slave print*, to send incoming host data directly to the printer, instead of to the screen.

Passthrough print is controlled by the host system, using terminal-type specific escape codes. Anzio recognizes passthrough print codes for all the emulated terminal types. If your host provides passthrough print, it will be directed to the current PRINTER setting automatically.

If your host does not provide passthrough print, and you would like to use it, contact Rasmussen Software for more information.

## 6.8 Secondary Passthrough Print

---

When Anzio is emulating an ANSI-style terminal (VTxxx, SCOANSI, LINUX, AT386, or ANZIO), it supports a secondary passthrough print channel. Secondary passthrough print always goes to the Raw Printer as described above. This provides a way to support multiple printers, especially for point-of-sale environments.

To initiate secondary passthrough print from the host, send out an escape character, followed by "[7i". To turn OFF secondary passthrough print, send out escape followed by "[6i".



## 6.9 File Transfer To Printer

---

To send incoming file data to the printer, using Kermit or similar transfer protocols, specify the PC output file as "WPRN". When Anzio sees this special name, it will send the received data to its print handler.

## 6.10 Printing Files

---

With Anzio's command interface, it is possible to print files that exist on a local disk, on a networked drive, or on an FTP, HTTP, or HTTPS server. Just use the PRINTFILE command, or use the COPY command to copy a file to "WPRN".

## 6.11 End-of-Print Flush Timer

---

When you print from Anzio, the output does not necessarily include page feeds (paper eject).

Printer data is saved up in a *print job*, until the job is *flushed* (sent) to the printer or print spooler.

To finish printing a page or job, use File/Eject. See also the FLUSH and EJECT commands.

You can set a delay time to use between the last printed character and an automatic eject. Use File/Flush Timer, or the FLUSH TIMER command. The default FLUSH TIMER is 5 seconds. If something has been sent to the printer, and 5 seconds go by without anything more being sent to the printer, Anzio assumes you are done printing, and it flushes the job. Five seconds after a screen print, or 5 seconds after your last passthrough print data comes in, the data is printed on the printer.

Note that the passthrough print protocol does not have a mechanism by which the host can tell Anzio that it is finished with a print job. That is why the FLUSH TIMER setting is critical. If passthrough print data arrives sporadically from the host, with pauses, you may need to increase the FLUSH TIMER, to prevent Anzio from breaking the one print job into many. Conversely, if you need the print job to be processed immediately, set FLUSH TIMER lower, as low as 1 second.

You can disable this feature by setting FLUSH TIMER to 0. In this case, the print job will be closed by File/Eject, an EJECT or FLUSH command, or when you exit Anzio.

## 6.12 AnzioWin's Print Wizard

---

AnzioWin's Print Wizard feature is intended to solve a problem. Incoming print data arriving at its passthrough print channel can be structured many different ways. The host may assume certain printer settings, such as paper size, or it may try to set them with "standard" escape codes.

The Print Wizard will analyze the print data, looking for typical host assumptions, and try to determine the best way to print out the data. The goal is to have an acceptable printout in a wide range of circumstances.

Use File/Print Wizard to turn the Print Wizard on or off. The default is on.

When the Print Wizard is on, it will store and analyze data to be printed. It determines the maximum number of columns per line, while dealing with tab expansion, backspace-bolding, backspace-underlining, wide Far East characters, etc. It also determines the number of lines per page, if the print data contains formfeeds, and makes some assumptions if it does not. Finally, it calculates and then uses logical values for margins and character height, width, pitch, and linespacing. The result is that data in almost any format is printed in a logical way, with no user intervention.

During its analysis, Print Wizard will notice if the file contains escape codes or PostScript source. If so, it will stop further processing and switch to spooler-mode printing, so those codes in the data can control the formatting of the printout.

---

➤ It is also possible to have Print Wizard interpret and translate escape codes that are designed for a PCL printer.

---

Print Wizard also gives you a unique ability to control the printer from a host program. Print Wizard uses its own extensions to the Web's HTML markup language to specify everything from paper orientation to bitmaps.

### 6.12.1 **Print Wizard Markup Language (PWML)**

Print Wizard's markup language provides the programmer a way to control Anzio's printing in a device-independent manner. With markup tags, the programmer can specify, among other things, paper size, paper orientation, fonts (name, size, bold, italic, underline, etc.), form overlays, margins, etc., as well as printing bitmaps, barcodes, rectangles, and more. The PWML code can be generated in nearly every programming language, as it consists of only ASCII characters.

PWML code can also specify a full range of international (Unicode) characters. Print Wizard will use its sophisticated font substitution logic to find a font and print these characters appropriately.

### 6.12.2 **Printing HTML**

Print Wizard can process and print many kinds of HTML files. In fact, PWML is based on HTML, with some additional tags added.

However, there are some elements of HTML that Print Wizard will NOT handle:

- Frames
- Tables
- CSS
- Free-form text with word-wrap

---

➤ For more information on Print Wizard, see the file PrintWiz.doc that came with AnzioWin.

---

### 6.13 **Print Initialization Files**

---

In some situations, the user needs to specify certain printing parameters or customize the printing of passthrough print jobs. For these situations, Anzio allows a *print initialization* file.

The standard name for the print initialization file is “\_prtinit.txt”. When Anzio receives a passthrough print job, it looks in the current directory to see if this file exists. If it does, the contents of this file are **prepended to**,

that is, inserted onto the beginning of, the print data, before it is routed to Anzio's print handler (or Print Wizard).

This file is a plain text file, and can be constructed in any text editor, such as "notepad". Any text, control codes, or markup tags will be sent to the print handler ahead of the passthrough print job. Text, for instance, would be printed at the beginning of the print job. Control sequences (escape codes) will be sent to the printer (assuming the print level is not High).

The most common use of the print initialization files is to include PWML tags to change the formatting of the printout. These tags might set orientation, paper size, multi-column labels, font, lines per page, etc. PWML tags can also specify a bitmap logo to be printed at the beginning of the job.

See the file PRINTWIZ.DOC for more information.

---

## **6.14 Troubleshooting Printing Problems.**

There are many ways that printing can go wrong. This section will help you fix some of the more common ones.

First, go to Edit/Advanced Options/General, and turn on "Debug printing". This will cause Anzio to pop up an information window whenever it processes a print job. That windows gives the most information when Print Wizard is on, but can give some useful information in any case.

### **6.14.1 When it Doesn't Print At All**

1. Make sure you can do a test print from the Windows printer driver.
2. Try printing the screen from Anzio, with Print Level set to High.
3. During passthrough print, if Debug Printing is on, and the debug box doesn't pop up, then Anzio is not getting the print job. There is a problem in the configuration of the host system.
4. If data goes to the screen when it should be going to passthrough print, then the host system may not have the correct

code to turn printing on and off in its terminfo, termcap, or equivalent file.

5. If you're trying to print in Raw mode, try Spooler instead.

#### **6.14.2 When it Prints, but Wrong**

1. If you're getting extra characters, especially at the beginning of the job, those may be escape sequences sent by the host. Try setting Print Level to Spooler.
2. Use the correct printer driver for your kind of printer. If it's a new printer, check to see if there's a later printer driver available from the manufacturer's web site. If there's still a problem, try using a driver for an older related printer, such as a LaserJet 5 driver for any HP laser printer.
3. Do not use a Generic driver if at all possible.
4. If Print Level is not High, and print layout is wrong, the problem is in the host application. In these levels, Anzio is completely transparent.
5. If passthrough print data is intermixing with screen data, then the passthrough print mechanism on the host is poorly designed. See our web site for help.
6. If the print data starts OK, but subsequent pages creep up or down the page, there is a problem with lines per page. Try experimenting with font size. If you're using Print Wizard, either adjust the host system to send 66 lines per page, or use a print initialization file (see above) to tell Print Wizard how many lines per page the host data contains.
7. (AnzioWin only) If your host system lets you tell it what kind of printer you have, try telling it you have a "generic" or "no codes" printer, so the application will send out plain text (without control sequences). Then turn on Print Wizard. Print Wizard will automatically fit each report to the page.
8. If international characters or diacritics are wrong, Anzio may not be set to the correct character set. Go to Communicate/Character Sets/For Printer/8-bit, and set it to match the encoding scheme used in the data.

9. If printing is too slow on a dot matrix printer, and Print Level is High, try choosing a font that is resident in the printer (one without a "TT" icon).
10. If there is a long delay before printing starts, try going to File/Flush Timer, and setting it to 1.

## **Part II Data Capture and File Transfer**

## 7 Data Capture

---

Anzio lets you capture screen data to the printer or to a file. This captured data can be *static*, that is, what's on the screen at any point in time, or *dynamic*, where it's captured as it is received from the host.

The next section, *File Transfer*, describes how to transfer files to and from the host.



The following sections describe:

- *Static Capture*
- *Dynamic Capture*
- *Control Characters During Capture*
- *Character Set of Output File*

### 7.1 Static Capture

---

This section describes capturing data that is already on the screen. You can capture screen data as characters (KEEP), as data columns (PICK), or as printed text (PRINT). The following descriptions refer to Anzio's commands:

KEEP	Reads the screen, or an area of the screen, and saves the characters to the current output file. Use OPENO first to open an output file. Enter the KEEP command to save the entire screen, or use KEEP  to select a part of the screen.
PICK	Reads a highlighted area of the screen as data columns, and saves them to the current output file. PICK writes each row of data in the standard comma-separated-values (CSV) format, for use with other programs.
PRINT	Prints all or part of the live screen, except the status line. You can also use PRINT  to select a part of the screen.



Through the menu system, you can also do the equivalent of PRINT: use File/Print Screen or **[ALT] [P]** to print the screen, or press **[PRTSC]**.

Through the menu system, you can also do the equivalent of KEEP. First, open a file using File/Open New File to Receive. Then, do Edit/Copy Screen to Output File.

## 7.2 Dynamic Capture

---

This section refers to capture of characters as they come from the host system.

With Transfer/Capture to Printer, you can tell Anzio to capture incoming data to the printer. Data will go to the Print Wizard, or to the printer at a particular Print Level, as configured. Note that the print job will not finish printing (and may not even start printing) until you turn OFF Capture to Printer (Flush Timer is ignored here).

With Transfer/Capture, you can capture to a file. Anzio will allow you to choose the location and name of the file to be created, then will start capturing to it.

## 7.3 Control Characters During Capture

---

When you are capturing data coming from the host, often the readable data will also contain control sequences, such as for cursor positioning, that you don't want to capture. However, there may be control characters that you need to capture also. So you have the option of specifying which, if any, control characters to capture, in Edit/Advanced Options/General. There you can set "Capture control characters" to "None", "All", or "All except escape sequences". We suggest experimentation.

➤ Certain settings will alter how the data is formatted on the screen.

## 7.4 Character Set of Output File

---

If data being captured is other than plain ASCII, there is the question of what character set (encoding scheme) should be used in writing it to the

file. Go to Edit/Advanced Options/General, and see “Character set for output file”. Your options are “ISO”, “OEM”, and “UTF-8”.

For the best capture of international characters, choose “UTF-8”. The resulting file can be imported in Microsoft Word, for instance, if you identify it as “encoded text” with encoding “UTF-8”.

## 8 File Transfer

---

Anzio provides several methods to transfer files between your PC and the host. The previous section described how to capture screens of data to a file.

This section shows how to transfer files in either direction:

- PC to host, called “upstream” or “upload”
- Host to PC, called “downstream” or “download”

In many cases we can transmit data directly into existing software on the host system, and it will think we are typing it. In nearly all cases we are limited to passing ASCII data only.

A *simple upload* involves sending sequential ASCII records upstream. Most text files can be sent to the host with this method. See:

- *UNIX Simple Upload*, page 84

To receive simple files, or to transfer binary files or files with records longer than 80 bytes, you will need special host software. The Anzio distribution diskette includes several simple file transfer utility programs for different hosts. See:

- *UNIX File Transfer*, page 84

Anzio can also send and receive files using the standard protocols *Kermit* and *Zmodem*. Each of these protocols requires that the host have the corresponding software installed. See:

- Using Kermit , page 86

The following sections describe:

- *UNIX File Transfer*
- *Using Kermit*
- *Using Zmodem*
- *C332 File Transfer*
- *FTP*

- *SFTP*

## 8.1 UNIX File Transfer

This section describes several file transfer methods for UNIX operating systems:


- *UNIX Simple Upload*
- UNIX Simple Download Script (download)
- UNIX Download Utility (Send-PC.C)
- Using Kermit



➤ The utility programs Send-PC and Recv-PC are provided in C source form. To use a utility, you must first upload the source file to the host. Then, you will need to compile the source code into an executable program. See the example under UNIX Download Utility (Send-PC.C), page 85.











### 8.1.1 UNIX Simple Upload

You can upload PC text files directly to a UNIX host.

First, tell UNIX to copy incoming characters to a file:

```
cp /dev/tty filename 
```

Then, tell Anzio to open and transmit the PC file, following it with UNIX end-of-file ( ):


```
   OPENI pcfilename   
   TRANSMIT TRAILER   
```

➤ If you have problems with missing characters, or with the process locking up, you need to increase the character delay time and/or the line delay time. Use Transfer/Delays, or the DELAY and LINE DELAY commands.


### 8.1.2 UNIX Simple Download Script (download)

One of the utility files provided is a UNIX shell script called `download`. This script can download one or more text files at a time to the PC. Unlike the Send-PC utility described below, the `download` script does not need to be compiled.


1. Use the *UNIX Simple Upload* procedure to upload the `download` file to the UNIX host.
2. Tell UNIX to make the file executable:

```
chmod +x download 
```

3. To download a file, tell UNIX:

```
download filename 
```

`download` sends the indicated file(s) to Anzio, using each host file name for the PC file name. *filename* can contain UNIX wildcard characters, for example, to download all C source files in the current directory, tell UNIX:

```
download *.c 
```

---



➤ `download` will overwrite an existing PC file of the same name without warning.

---











### 8.1.3 UNIX Download Utility (Send-PC.C)

The SEND-PC program will transfer virtually any UNIX text file down to the PC. You will need to upload the C source file `Send-PC.C` and compile the program on the host.

First, tell UNIX:

```
cd   
cp /dev/tty send-pc.c 
```

Then, tell Anzio:

```
   OPENI SEND-PC.C   
   TRANSMIT TRAILER   
```

Wait until the transfer is finished, then tell UNIX to compile send-pc.c into the object file "send-pc":

```
cc send-pc.c -o send-pc
```

If you do not have a C compiler ("cc"), we may have provided you with objects (executable programs) for your particular UNIX system. Check the README.TXT file on the distribution diskette.

To use Send-PC, tell the host:

```
send-pc unixfilename pcfilename
```

The program will delete any existing *pcfilename*, create a new one, and send the *unixfilename* down.

#### 8.1.4 UNIX Upload Utility (Recv-PC.C)

The program RECV-PC.C is included for use with PC text files. You will need to upload and compile Recv-PC.C as described above.

To transfer a file, tell UNIX:

```
recv-pc unixname pcname
```

## 8.2 Using Kermit

---

The Kermit file transfer protocol is available for most UNIX, Linux, and similar systems, on the Internet. It may be installed already on your system. Kermit allows transfer of many kinds of files between many kinds of systems, with error checking, data compression, and more. Contact us for information on obtaining Kermit for your UNIX machine.

- Anzio can respond to an incoming Kermit file download request automatically, if Transfer/Kermit/Auto Reception is checked on (default is off).

Anzio provides several Kermit-specific commands: KSEND, KRECEIVE, and KCOMMAND. These commands are described beginning on page 102.

Kermit on UNIX should come with instructions, because it is not especially user-friendly, so we'll give some pointers here:

**PARITY**            If your UNIX system is set up for even parity, you will need to set up Kermit that way too. To do so on startup, tell UNIX:

```
kermit -p e
```

Or, once Kermit is running, you can give it the command:

```
set parity even
```

Finally, it is possible to put the "set parity even" command in a file named ".kermrc" in your user home directory.

#### TEXT vs. BINARY

UNIX Kermit must know whether files being transferred are TEXT (which translates between **linefeed** for end-of-line on UNIX and **return-linefeed** for end-of-line on Windows) or BINARY (which does no translation). To see its current setting, tell UNIX Kermit:

```
sh
```

To set it , tell UNIX Kermit:

```
set file type x
```

where x is text or binary.

#### SERVER MODE

For less confusion, we recommend putting UNIX Kermit in "server" mode, by telling it:


```
server
```

Then, give it commands using Anzio's Kermit commands.

To terminate the server mode, tell Anzio:

  **F** KCOMMAND FINISH 

To terminate Kermit itself, tell Kermit:

quit 

---


### 8.3 Using Zmodem

---

Zmodem, like Kermit, is available for most host systems, and may be on your system already. It is usually in the form of a sending program (“sz”) and a receiving program (“rz”).

- Anzio can respond to an incoming Zmodem file download request automatically, if Transfer/Zmodem/Auto Reception is checked on (default is off).

To download a file using Zmodem, give the host the command

```
sz filename 
```

To upload a file using Zmodem, first make sure the host system is at a shell prompt. Then go to Transfer/Zmodem/Send, and choose a file to transfer.

For more tips on Zmodem file transfer, see our web site.

---

### 8.4 C332 File Transfer

---

Anzio, when in C332 emulation, also supports a special file transfer protocol used by some Versyss systems. These transfers are controlled by the host system.

---

### 8.5 FTP

---

AnzioWin includes two different FTP client capabilities, one interactive and one command-driven.

- An FTP transfer requires you to have a network connection to the host system. It will not work over a serial or dial-up modem connection.



### **8.5.1 Interactive FTP**

The Transfer/FTP menu item, or the “FTP” Anzio command, will initiate an FTP connection to the same server that Anzio is currently connected to, with your current username and password (if Anzio knows them).

The Transfer/FTP (different host) menu item, or the “FTP NEW” Anzio command, will initiate an FTP connection to an arbitrary host system. It presents first a dialog box that asks for the host name (or IP), the port, the username, and the password. Then it connects.

In either case, you will then see a dialog box that allows you to interactively move files between the host system and the PC.

### **8.5.2 Programmed FTP**

Anzio provides commands to programmatically perform FTP operations. These commands can be typed into Anzio’s Function prompt, or they can be programmed into a macro or sent from the host as an Anzio command. That means an FTP transfer can be done under program control (or macro control), without user intervention.

The commands for this are FTPGET, FTPPUT, and FTPDELETE, and also their “/S” variations. See the reference section.

## **8.6 SFTP**

---

AnzioWin includes two different SFTP client capabilities, one interactive and one command-driven.

➤ An SFTP transfer requires you to have an SSH network connection to the host system. It will not work over a serial or dial-up modem connection, or another type of network connection.

### **8.6.1 Interactive SFTP**

The Transfer/SFTP menu item, or the “SFTP” Anzio command, will initiate an SFTP connection to the same server that Anzio is currently connected to, as the same user.

You will then see a dialog box that allows you to interactively move files between the host system and the PC.

### **8.6.2 Programmed SFTP**

Anzio provides commands to programmatically perform SFTP operations. These commands can be typed into Anzio's Function prompt, or they can be programmed into a macro or sent from the host as an Anzio command. That means an SFTP transfer can be done under program control (or macro control), without user intervention.

The commands for this are SFTPGET, SFTPPUT, and SFTPDELETE, and also their "/S" variations. See the reference section.

## **Part III Reference Guide**

## 9 Command Reference

---

This section lists all Anzio text commands, first summarized by type, then described in alphabetical order. Anzio commands can come from three sources:

- Anzio's command line, where you typed the command – see *Entering Commands*, page 34.
- A defined key macro – see *Defined Keys and Macros*, page 38.
- From the host computer – see *Sending Anzio Commands From The Host*, page 192.

---

➤ Anzio Lite does not include some commands.

---

The following sections describe:

- *Commands By Type*
- *Alphabetical List of Commands*

### 9.1 Commands By Type

---

This section lists Anzio's commands by type:

- *Operator Preference Items*
- *Communication Parameters*
- *File Transfer*
- *Local Processing*
- *Printing*
- *Macro Processing & Key Mapping*
- *Interaction with Host and User*

- *Miscellaneous*

Commands are described in alphabetical order beginning on page 98.


### 9.1.1 **Operator Preference Items**

ALLOW-QUIT	Specifies whether Anzio can be terminated locally
BEEP	Turns beep on or off
BEEP IDLE	Beeps when idle
BMP	Sets a background bitmap on the screen
COLOR	Sets operating colors/attributes (old method)
COLOREDIT	Sets operating colors/attributes (current method)
CURSOR BLINK	Allows non-blinking cursor
FONT	Sets the screen font and size
FONT-ALT	Sets alternate fonts to be used for international characters
GAUGE	Shows gauge line at bottom of screen
JUMP	Sets jump scrolling
MENULANGUAGE	Sets the language for Anzio's menus, prompts, etc.
PALETTE	Sets the group of available colors and attributes
PITCH	Sets beep pitch, sound, speakers
REGTELNET	Registers or unregisters Anzio as the PC's standard telnet client
SCREENMODE	Sets various screen modes
SCREENMODE/S	Reports screen info to host
STATUS LINE	Turns status line on or off
TITLE	Sets window title

TRACK-WINDOW Should Anzio save its window position?

### 9.1.2 Communication Parameters

7E1, 8N1, etc.	Shortcuts to set standard data bits, parity, stop bits
AUTO-LF	Sets auto-linefeed
AUTO-CONNECT	Specifies whether to connect on startup
AUX	Turns on/off aux input
AUX-BIDI	Puts the AUX port into bidirectional mode
AUX-DEBUG	Turns on/off debugging of aux communication
AUX-READ	Read aux input until a certain string
AUX-SETUP	Configures aux input
AUX-WRITE	Write output to aux port
BACKSPACE	Configures the Backspace key
BAUD	Sets the baud rate
CHARSET[S]	Sets character set translation
COMMTYPE	Sets communication type, serial or network
DATA BITS	Sets the number of data bits
DELAY	Sets the delay between characters
DELAY/S	Reports DELAY settings
FULL DUP	Sets full-duplex mode
HALF DUP	Set half-duplex mode
HOLD	Suspends host output
IGNULL	Ignores null characters
INTERPRET	Calls communication diagnostics
LINE DELAY	Sets the delay for line turnaround

LOCK	Sets the keyboard locking protocol
LOGIN-WIZARD	Specifies whether Anzio will attempt to log in automatically to the host system
MONITOR	Communication diagnostics
PARITY	Sets the parity type
PORT	Sets the communication port number
RECONNECT	Reconnect on line drop?
RESET	Resets the Anzio terminal
SCROLL	Saves erased data
SCROLL-LOCK	Does the  key suspend host output?
STOP BITS	Sets the number of stop bits
TAB	Defines tab stops and the character sent for Tab
TERM	Sets the terminal emulation type, such as VT100
TERMNAME	Sets the terminal type that is reported to host
UPPERCASE	Changes keystrokes to upper case

### 9.1.3 File Transfer

ASEND	ASCII mode upload
CAPTURE	Captures incoming screen data to a PC file
CLOSEI	Closes input file, optionally reporting to host
CLOSEO	Closes output file, optionally reporting to host
DELETE/S	Deletes a file and reports to host
DOWNLOAD-DIR	Sets the preferred directory for downloads
DOWNLOAD-LOCK	Restricts downloads to a certain directory
FIND/S	Finds a PC file name and sends it to host

FINDNEXT/S	Finds next PC file name and sends it to host
FTP	Does an interactive FTP session
FTPDELETE[/S]	Delete a file via FTP
FTPGET[/S]	Download a file via FTP
FTPPUT[/S]	Upload a file via FTP
KCOMMAND	Sends a Kermit command
KEEP	Copies part of screen to a file
KRECEIVE	Receives a file with Kermit
KSEND	Sends a file with Kermit
OPENE	Opens a PC file in extend mode for output from host
OPENI	Opens a PC file for input to host, optionally reporting to host
OPENO	Opens a PC file for output from host, optionally reporting to host
PICK	Picks data from screen columns for spreadsheet
PURGE	Clears the file transfer buffer
RECEIVE CODED	Receives a file coded for PC-to-PC transfer
RECEIVE QUIET	Doesn't display file transfer
RETRANSMIT	Retransmits last element
SFTP	Does an interactive SFTP file transfer session
SFTPDELETE	Deletes a file on the host via SFTP
SFTPRECEIVE	Receives a file from the host via SFTP
SFTPSEND	Sends a file to the host using SFTP
TIMEOUT	Sets the file transfer timeout



TRANSFER-OPT	Brings up the Transfer Options dialog box
TRANSMIT	Provides simple file upload to host
WRITELN	Write text with CR/LF to output file
XN	Sends next TRANSMIT record
XRECEIVE	Receives a file using Xmodem
XSEND	Sends a file using Xmodem
YRECEIVE	Receives a file using Ymodem
YSEND	Sends a file using Ymodem
ZRECEIVE	Receives a file using Zmodem
ZSEND	Sends a file using Zmodem

#### **9.1.4 Local Processing**

CALC	Invokes the built-in calculator
CD	Changes the logged disk/directory
CLONE	Create a duplicate Anzio session
COPY	Copies one PC file to another, optionally sending result to host
DDE (various)	Interact with other programs
DELETE	Deletes a file from PC, optionally ignoring "file not found"
DIR	Lists a PC disk directory, optionally sending to host
KILL	Terminates another program that Anzio started
LAUNCH	Starts another program in Windows
LOG	Changes logged disk/directory
MAPISENDFAX	Send a fax

MKDIR	Makes a directory, optionally sending result to host
OLE (various)	Interact with other programs
RENAME	Renames a PC file
REVIEW	Scrolls back data from top of screen
RUN	Runs a PC program and then returns to Anzio, optionally sending result to host
SAVE	Saves a key macro file
SCRIPT	Runs a Windows script
SWITCHTO	Switch focus to a different Windows program
WINSTART	Starts a Windows program
WRITE	Writes a string to the current output file

**9.1.5 Printing**

CHOOSEPRINTER	Opens the Printer Setup dialog box
EJECT	Ejects a printer page
FLUSH	Releases a print job to the printer
FLUSHTIMER	Sets timed release of print jobs
PASSTHRU	Initiates passthrough print
PASSTHRU-CP[/S]	Sets character type for passthrough print
PREVIEW	Turns on or off Print Preview
PRINT	Prints all or part of screen
PRINTER	Sets the printer to use
PRINTER-SETUP	Configures PC's printer
PRINTFILE	Prints a text file
PRINTFONT	Sets printer font and size

PRINTFONT-ALT	Sets alternate fonts for printing international characters
PRINTINIT	Specifies print initialization file
PRINTLINE	Print text with CR/LF
PRINTLOW	Sets low-level print
PRINTMACRO	Specifies macro(s) for code-activated switch
PRINTTOFILE	Controls printing to a file
RAW-SETUP	Configures Windows raw printing
SPOOLER	Specifies printer to use for print level of spooler
WINPRINT	Asks Windows to print a file using the associated Windows application

### **9.1.6 Macro Processing & Key Mapping**

BASENAME	Returns filename without path
CALL	Calls a macro key as a subroutine
DEFINE	Defines a macro
EVENT	Initiates macros when certain things happen
EXPECT	Test strings for equality
INVOKE	Stops current macro and starts another macro
KEYS	Shows special and macro keys
MERGE	Merges a macro file with macros in memory
ONERROR	Macro error handling
ONSTARTUP	Run a macro when starting Anzio
READ	Reads a key macro file
RECORD	Record a macro

REPLACE	Regular expression replacements
SET	Set a variable
WAIT	Waits a specified time
WAITFOR (WF)	Waits for specific incoming text until timeout
WAITQUIET	Wait until host stops sending

**9.1.7 Interaction with Host and User**

ADD128	Process certain special keystrokes
ANSWERBACK	Sets the Anzio terminal's answerback string
BLOCKTYPE	Sets shape of mouse highlight
BOX	Draws a box on the screen
BREAK	Sends a Break character
C332_NEXT	Sends code to host for next C332 terminal session
C332_PREV	Sends code to host for previous C332 terminal session
CLIP	Copies screen to Windows clipboard
COMPOSE/S	Create a diacritic/character combination
CONNECT	Initiate a connection with host
DATE/S	Sends the date to the host
DIAL	Dials a modem
ENV/S	Sends an environment variable to host
ENVIRONMENT	Sets Anzio internal environment variables
FILL	Fills a screen area with a character
HANGUP	Hangs up a modem
MCISENDSTRING	Exercise Windows' multimedia functions

MENUBAR	Creates a menu
MESSAGE	Displays a message box
PASSWORD	Sets the password for the host, or sends it to the host
PASTE	Pastes Windows clipboard text
PLAYMEDIA	Play various media files
PLAYSOUND	Plays a WAV file
PROMPT-ID	Prompts the user for username and password
PROMPT/S	Prompts the user for a string
SCREENSIZE/S	Tell the host your screen dimensions
SEND	Transmit (part) of the screen to the host
SETCOLOR	Reset current color
SPEAK	Text-to-speech
TIME/S	Sends PC time to host
TYPE	Shows a PC file on screen
UNICODE/S	Create a Unicode character
USERNAME	Sets the username for the host, optionally sending the username to host
VERSION	Displays Anzio version number, optionally sending result to host
WINDOW	Opens a text window on the screen
WINDOWCLOSE	Closes a text window

**9.1.8 Miscellaneous**

DEFAULTS	Saves settings
END	Quits Anzio

HELP	Gets help on Anzio
HOSTNAME/S	Sends host name to host
SLEEP	Wait until certain time
STOP	Quits Anzio
WINDOW ZOOM	Maximize the window
WINDOW MINIMIZE	Make the window an icon
WINDOW RESTORE	Bring the window out of zoomed or iconic mode

---

## 9.2 Alphabetical List of Commands

---

This section describes each command in Anzio, in alphabetical order. Each entry begins with the command name and syntax, followed by a description.

### 9.2.1 Syntax Conventions

- A vertical bar | indicates alternate commands, for example:

BACKSPACE 8 | 127

- means that you can enter BACKSPACE 8 or BACKSPACE 127.

- Square brackets [ ] indicate optional parameters:

AUTO-LF [ON]

- where AUTO-LF is equivalent to AUTO-LF ON.

- *Italics* indicate a user-specified string or number:

ANSWERBACK *string*

### 9.2.2 Commands

7E1

Shortcut for DATA BITS 7, PARITY EVEN, and STOP BITS 1.

7E2

Shortcut for DATA BITS 7, PARITY EVEN, and STOP BITS 2.

7M1

Shortcut for DATA BITS 7, PARITY MARK, and STOP BITS 1.

7M2

Shortcut for DATA BITS 7, PARITY MARK, and STOP BITS 2.

7N1

Shortcut for DATA BITS 7, PARITY OFF, and STOP BITS 1.

7N2

Shortcut for DATA BITS 7, PARITY OFF, and STOP BITS 2.

7O1

Shortcut for DATA BITS 7, PARITY ODD, and STOP BITS 1.

7O2

Shortcut for DATA BITS 7, PARITY ODD, and STOP BITS 2.

7S1

Shortcut for DATA BITS 7, PARITY SPACE, and STOP BITS 1.

7S2

Shortcut for DATA BITS 7, PARITY SPACE, and STOP BITS 2.

8E1

Shortcut for DATA BITS 8, PARITY EVEN, and STOP BITS 1.

8E2

Shortcut for DATA BITS 8, PARITY EVEN, and STOP BITS 2.

8M1

Shortcut for DATA BITS 8, PARITY MARK, and STOP BITS 1.

8M2

Shortcut for DATA BITS 8, PARITY MARK, and STOP BITS 2.

8N1

Shortcut for DATA BITS 8, PARITY OFF, and STOP BITS 1.

8N2

Shortcut for DATA BITS 8, PARITY OFF, and STOP BITS 2.





8O1

Shortcut for DATA BITS 8, PARITY ODD, and STOP BITS 1.

8O2

Shortcut for DATA BITS 8, PARITY ODD, and STOP BITS 2.

ADD128

Used with T160E emulation. Takes the next character entered, adds 128 to set the 8<sup>th</sup> bit, and sends the result to the host. By default, assigned to  and all variations (, , ).


ALLOW-QUIT [ON] | OFF

If ON, then Anzio will prevent the user from quitting at the PC end. Instead, the user would have to tell the host system to logoff. This would cause the host machine to drop its end of the network or modem connection. Anzio would detect this, and then quit (unless RECONNECT is on).

Using ALLOW-QUIT OFF can force users to terminate host-end programs properly.



ANSWERBACK *string*

Sets the ANSWERBACK to *string*. The ANSWERBACK can be sent to the host, on host request, in certain terminal emulations. To include an  in *string*, use a vertical bar "|".


ASEND *filename*

Causes Anzio to send then file *filename* to the host, using the ASCII file transfer "protocol", which consists of simply sending the file a line at a time.

AUTO-CONNECT [ON] | OFF | TOGGLE

AUTO-CONNECT ON causes Anzio to immediately try to connect to the host when it is started up.

AUTO-LF [ON] | OFF | TOGGLE

AUTO-LF ON causes Anzio to send a line-feed character after every carriage-return (). It also adds a linefeed to any carriage-return received from the host.

AUX [ON] | OFF | TOGGLE

AUX ON causes Anzio to accept input from a serial device such as a barcode reader, on a communication port configured by AUX-SETUP. AUX OFF turns off that input.

AUX-BIDI

Puts the AUX port into bidirectional mode. Data received from the host is sent out the AUX port (instead of to the screen), and data received from the AUX port is sent to the host.

This mode is terminated when the host sends a sequence consisting of an Escape character followed by "[2{".

AUX-DEBUG [ON] | OFF | TOGGLE

Turns on or off the AUX-DEBUG feature. When this feature is on, and you do input from or output to an AUX port, such as for a serial barcode reader, a window will pop up to show data on the aux channel.

AUX-READ/S *string*

Starts reading from the AUX port, sending received data to the host, until the terminator *string* is received on the AUX port.

AUX-SETUP

AUX-SETUP configures a comm port (NOT the one connected to your host system) to accept serial data from an external device such as a barcode reader. You will be prompted for a port name, and then for baud rate, etc. Input will not begin until you do AUX ON.

AUX-WRITE *string*

Sends *string* to the AUX device. This does NOT alter the state of AUX input - if it was allowed before, it is allowed after this command, etc. If *string* contains spaces, use quotes around entire string. This command could be used to send a trigger code to an electronic scale, for instance.

BACKSPACE 8 | 127

Sets the ASCII character sent for the Backspace key, either a BS (decimal 08, hex 08, ctrl-H) or a DEL (decimal 127, hex 7F).

BASENAME/S *filepath*

Sends to the host the filename, and extension if it exists, of the *filepath*. That is, it strips off the disk and directory part.

BAUD *nnnn*

Sets the connection's baud rate, in bits per second. Available rates are determined by Windows.

BEEP [ON] | SLOW | OFF

Controls the PC's beeper. BEEP ON sets the beeper to sound when a beep character (Ctrl-G, ASCII 7) is received. BEEP SLOW is the default, sounding the beep only when 1) the host issues a beep and unlocks the keyboard, and 2) there is nothing to be sent (e.g. a defined key). BEEP OFF silences the PC.

BEEP IDLE [ON] | OFF | TOGGLE

BEEP IDLE ON sounds the PC beep once per minute, when the host is idle and expecting input, for example, when online and at the command level. BEEP IDLE OFF turns this feature off.

BLOCKTYPE COLUMN | STREAM | WRAP

Sets the "shape" of the selection (highlight) area that is created when you drag the mouse. A COLUMN block is rectangular, and will be copied to the clipboard as several separate lines. A STREAM block is similar to that used in word processing programs, and will also be copied to the clipboard as separate lines. A WRAP block is similar to a STREAM, but lines are joined with an intervening space, except where there is a blank line.

BMP *filename* WATERMARK

BMP *filename* [CELL] [*window-rect* [*view-rect* [*bright* [*contrast*]]]]

Tells AnzioWin to open *filename*, which must be a BMP, GIF, or JPG file, and display it on the screen. The *filename* can also be a URL; Anzio will fetch the file from a web or FTP server.

One bitmap at a time can be used. A bitmap can be displayed on the entire Anzio screen, with text over it. This is known as a watermark bitmap. Or, a bitmap can be displayed in a particular part of the screen, known as a *window-rect*, the corners of which can be specified in either pixel coordinates or cell coordinates. In addition to a *window-rect*, you can specify that only a portion of the original bitmap is displayed; this portion is called a *view-rect*. By changing the *window-rect* and the *view-rect*, various zooming, panning, and stretching effects can be created. It is also possible to alter the bitmap's *brightness* and *contrast*.

It is possible in any case to have text displayed over the bitmap. Anzio will check the background color of the text to be displayed. If that background color matches the background color of the “normal” color in Anzio’s Custom Color screen, that character cell will be transparent, and the bitmap will be visible. All other background colors will block out the bitmap.

If you print the screen when a bitmap is displayed, the bitmap is also printed, *unless* it is a WATERMARK bitmap.

Optional parameters indicate which part of the window is affected, which part of the bitmap to use, and brightness and contrast settings.

WATERMARK

Specifies that the image should be stretched or shrunk to completely fill the screen, including the border areas. Overrides any *window-rect* or *view-rect* settings.

CELL

States that coordinates in the *window-rect* are in character rows and columns, rather than pixels.

*window-rect*

Area of Anzio window, expressed as a rectangle in four numbers: *left top right bottom*. Each number represents a character offset from the upper left corner of the screen, for example, 1 1 80 24 (if CELL is specified). A value of -1 for *top* or *left* represents that edge of the screen. The *right* and *bottom* numbers may be 0, to represent the full window’s right and bottom edges.

The window rectangle may also be set with the BMP-WINDOW command.

*view-rect*

Rectangular area of the bitmap to be used, as *left top right bottom*. Always in pixels.

The *right* and *bottom* numbers can be 0, to

represent the bitmap's right and bottom edges.

The bitmap can be stretched if either *right* or *bottom* of *window-rect* is -1, or if *right* or *bottom* of both rectangles is specified.

The view rectangle can also be set with the BMP-VIEW command.

*bright* Brightness, between 0 and 2000, where 1000 is "normal". Can also be set with the BMP-ADJUST command.

*contrast* Contrast, between 0 and 2000, where 1000 is "normal". Can also be set with the BMP-ADJUST command.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

BMP-ADJUST *bright* [*contrast*]

Sets the background bitmap's brightness and contrast, as described above. If your video adapter is running in 256-color mode, Anzio will manipulate the palette in order to change brightness and contrast. In true-color modes, Anzio will internally adjust the bitmap, which will take longer.




BMP-CLOSE

Closes the current background bitmap, if any, and removes it from the screen. Use the BMP command to set the background bitmap.

BMP-VIEW *left top right bottom*

Sets the background bitmap's view rectangle as described under the BMP command.



	Exits the calculator, leaving the result on the status line.
	Exits the calculator and sends the result to the host computer.
H	Switches between decimal (default) and hexadecimal mode (indicated by 'HEX' on the status line). Changing modes converts the current result to the new base.
	sends the result to the host in the current mode.

*CALL macro*

This command allows one macro to start another macro. When the second macro finishes, the first (calling) macro will resume. See also INVOKE.

CAPTURE [ON] | LONG | OFF  
CAPTURE *filename* | CLOSE  
CAPTURE WPRN

The CAPTURE feature saves a copy of all displayable data received from the host. When CAPTURE is ON, each time the cursor moves to a new line, the previous screen line is added to the receive buffer. You cannot capture data and download a file at the same time, since the receive buffer is also used for file transfer.

CAPTURE LONG is used when records longer than the screen width (over 80 bytes) are being received. Each long record is terminated by a carriage-return.

If an output file is open (from the OPENO command), captured data is written to that file. Use CAPTURE ON and CAPTURE OFF to start and stop recording.

CAPTURE *filename* opens a file and begins capture. CAPTURE CLOSE turns off capture and closes the file.

CAPTURE WPRN sends the captured data to the Windows printer driver.

CD [*unit:*]*directory*

Sets the default ("logged") disk drive and directory. The logged disk and directory are used for local file operations (DIR, OPENO, etc.) when no drive or directory is specified.

The optional disk unit must be followed by a colon.

Synonymous with the LOG command.

CHARSET *name*

CHARSET/S *name*

Specifies which character set coding is used by the host system, which in turn causes Anzio to translate characters appropriately.

*Name* can be a language, representing a National Replacement Character (NRC) set. If *name* is INTERNATIONAL, no NRC translation is done.

<i>language</i>	CANADIAN, FINNISH, FRENCH, GERMAN, ITALIAN, NORWEGIAN, SPANISH, SWEDISH, SWISS, UK, SLOVENIAN, JAPANESE
-----------------	--

*Name* can also be a specific character set name. Character set names can only be used with ANSI emulations (VT-xxx, SCOANSI, AT386, and ANZIO).

ISO	Standard ANSI (Windows) character set
OEM	The DOS or "linedraw" character set.
ISO-8859-x	x is 2 through 9 or 15
DEC	Slightly different from standard ISO
CP-xxx	xxx is the number of a Windows or DOS <i>character codepage</i> , such as 850. Requires Windows support for the particular codepage.
T160E	Innovative Interface's T160E multilingual coding scheme



CCCII	Chinese, Japanese, and Korean combined character set (also known as EACC)
ASMO449	Arabic
KOI8-R	Russian
VISCII	Vietnamese
UTF8	Byte-encoded Unicode characters
USMARC	US Library of Congress scheme
CUSTOM	Allows use of a user-editable translation file, CUSTOM.UNI
<i>other</i>	Other encodings as provided by additional *.UNI files in Anzio's directory

CHARSET/S returns a status result to the host: "00" for success, and "01" for failure. That way a host program can determine whether this installation of Anzio has support for a particular encoding.

See also PASSTHRU-CP to set the printer code page.

CHOOSEPRINTER [*name* | DEFAULT]

CHOOSEPRINTER brings up the Windows Printer Setup dialog box, so the user can select or set up a printer.

The optional *name* is the name of a Windows printer, exactly as it appears in the Windows Printers box. Or, use CHOOSEPRINTER DEFAULT to choose the current default Windows printer.

CLIP [*c1 c2 r1 r2*]

Copies the screen text (or just the given region) to the Windows clipboard. If the columns and rows are not given, Anzio will copy the entire screen to the clipboard, unless a highlight is in place, in which case it will copy just the highlighted area.

If the coordinates are given, they can be cursor-relative – see BOX.

Anzio stores the screen in the clipboard in three ways: text, Unicode, and bitmap. Use your PC program's "Edit/Paste Special" command to get data from the clipboard in one of these formats.

Use PASTE to send the clipboard text to the host.

CLIP "*string*"

Sends the literal *string* to the clipboard. The quotes are necessary to differentiate this from the other form of CLIP.

CLONE

Creates another Anzio session, using the same parameters as the current session. That is, it clones the connection. If the parameters have been changed, you will be prompted to save them first.

CLOSEI

CLOSEI/S

Closes the input file, if any. See OPENI.

CLOSEI/S also sends a result code to the host:

00	Completed successfully
01	Error occurred

CLOSEO

CLOSEO/S

Closes the output file. See OPENO.

CLOSEO/S also sends a result code to the host:

00	Completed successfully
01	Error occurred

COLOR

Starts a dialog for specifying which colors/attributes are used for normal text, highlights, etc. Depending on the terminal being emulated, there are up to four video attributes: reverse video, half intensity, underline, and blink. These may be used in any combination, for 2<sup>4</sup>, or 16, possibilities.

Anzio has an internal translation table to change each combination of emulated attributes into a character-level parameter for the PC's video driver. The COLOR dialog is used to change that translation table.

Press the key for the attribute combination to change (0 through 9, A through F), then enter the two-byte color/attribute code from the table shown at the right.

The **P** key (for *palette*) cycles through several combinations of blink/underline and bright/dim attributes.

Press **ESC** to leave the COLOR dialog without changes, or **SAVE** to save the changes.

You can also reset all colors to our defaults, as indicated.

COLOREDIT

Brings up the standard color mapping dialog.

COMMTYPE *name*

Tells Anzio which communication method it should use to communicate with the host system. Changing the COMMTYPE causes an existing connection to be dropped.

Possible values for *name* are

SERIAL  
TELNET or WINSOCK  
RLOGIN  
MODEM  
SSH

COMPOSE/S x[y]

Composes a special character, usually a combination of a base character and a diacritic, as represented by the two-character sequence *xy*. If *y* is not given, the user will be prompted for it. For instance,

COMPOSE/S :A

will generate an A-umlaut, and send it to the host.

The first character in the sequence usually represents a diacritic, as follows:

<u>Character</u>	<u>Meaning</u>
`	grave
'	acute
^	circumflex
>	circumflex
~	tilde
-	macron
(	breve
*	dot above
:	umlaut
"	umlaut
?	hook above
<	hacek
,	cedilla
.	dot below
_	underline
=	double underline

All standard Unicode character/diacritic combinations using the above diacritics (as the first character) are supported. In addition, Anzio defines the following list of sequences and the generated characters:

`a	à
`e	è
`i	ì
`o	ò
`u	ù
`A	À

`E	È
`I	Ì
`O	Ò
`U	Ù
'a	á
'e	é

'l	í
'o	ó
'u	ú
'A	Á
'E	É
'l	Í
'O	Ó
'U	Ú
'd	ð
'D	Ð
-D	Ð
'Y	Ý
'y	ý
^a	â
^e	ê
^i	î
^o	ô
^u	û
^A	Â
^E	Ê
^l	Î
^O	Ô
^U	Û
:a	À
"a	À
:e	È
"e	È
:i	Ì
"i	Ì
:o	Ò
"o	Ò
:u	Ù
"u	Ù
:A	Ä
"A	Ä
:E	Ë
"E	Ë
:l	Ï
"l	Ï
:O	Ö

"O	Ö
:U	Ü
"U	Ü
:y	ÿ
"y	ÿ
:Y	ÿ
"Y	ÿ
:<space>	"
"<space>	"
,c	ç
c,	ç
,C	Ç
C,	Ç
,<space>	,
~a	ã
~n	ñ
~o	õ
~A	Ã
~N	Ñ
~O	Õ
&a	æ
Ae	æ
&s	ß
Ss	ß
*A	Ä
Th	þ
*a	å
&A	Æ
AE	Æ
&o	œ
Oe	œ
&O	Œ
OE	Œ
/o	ø
o/	ø
/O	Ø
O/	Ø
C/	ç
C	ç
!!	¡

??	¿
L-	£
L=	£
Y-	¥
Y=	¥
SO	§
S0	§
S!	§
XO	¤
X0	¤
CO	©
C0	©
A_	ª
<<	«
0^	°
O^	°
+-	±
2^	²

3^	³
/U	µ
P!	¶
.^	·
1^	¹
O_	º
>>	»
14	¼
12	½
RO	®
-^	-
34	¾
-:	÷
-d	Ð
TH	Þ
TM	™
e=	€

CONNECT

[*commtyp*://][*username*[:*password*]@]*hostname*[:*port*]

Causes Anzio to drop its current connection (if it has one), and establish a new connection. If just CONNECT is done, Anzio will reconnect according to all its current settings. Or, the command can specify a new *hostname*, a new *commtyp*, etc. See the COMMTYPE command for possible values for *commtyp*. For instance, the following command will establish a username of "sally" and establish a modem connection to the indicated phone number:

```
CONNECT modem://sally@555-4321
```

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

COPY *filename newfile*  
COPY/S *filename newfile*

Copies *filename* to *newfile*. If *newfile* exists, it is overwritten without warning.

COPY/S also sends a result code to the host:

00	Completed successfully
01	Error occurred

If *newfile* is "WPRN", the file identified by *filename* is sent to Anzio's printer driver.

If *filename* and/or *newfile* contains the string "://", that name is interpreted as a SUPERFILE name.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

CURSOR [BLINK] [ON] | OFF

Tells Anzio whether you want the standard blinking cursor, or the non-blinking pseudo-cursor.

CURSOR BLOCK | UNDERLINE

Tells Anzio whether you want a block cursor or an underline cursor.

DATE/S [*format*]

Sends today's date, as known by the Windows clock, to the host. The default format is "mm/dd/yy". If *format* is given, it is a string of tokens representing the elements of the date to be sent, as follows:

d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).

ddd	Displays the day as an abbreviation (Sun-Sat).
dddd	Displays the day as a full name (Sunday-Saturday).
m	Displays the month as a number without a leading zero (1-12).
mm	Displays the month as a number with a leading zero (01-12).
mmm	Displays the month as an abbreviation (Jan-Dec).
mmmm	Displays the month as a full name (January-December).
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).

Note that day and month names and abbreviations will vary with the Windows locale.

DATA [BITS] 5 | 6 | 7 | 8

Sets the number of data bits in the serial communication protocol, usually 7. Note that this bit count does not include the parity bit, if any.

DDEEXECUTE *command*

Executes a DDE *command*. A DDE conversation must have been established with another program, with the DDEINITIATE command.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.



DDEINITIATE *program topic*

Causes Anzio to act as a DDE client, and begin a DDE conversation with *program*, about *topic*. Only one DDE conversation can be in effect at one time. It is terminated with a DDETERMINATE command. Note that *program* is a special name used for DDE by another program; it might NOT be the same as the executable's name.

Arguments can also contain *\${variable}* where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

DDEPOKE *item data*

Sends data into another program, using DDE. A DDE conversation must have been established with another program, with the DDEINITIATE command.

Arguments can also contain *\${variable}* where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

DDEREQUEST/S *item*

Requests data from another program using DDE. A DDE conversation must have been established with another program, with the DDEINITIATE command. The data returned is sent by Anzio to the host.

Arguments can also contain *\${variable}* where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.


DDETERMINATE

Terminates a DDE conversation started with DDEINITIATE.

DEFAULTS

Saves the current parameter settings to a file.

DEFINE *x text*

Defines key *x* as *text*, so that when you enter  *x*, the *text* is sent to the host. See *Defined Keys and Macros*, page 46.

DELAY *n*

DELAY/S

DELAY sets the time delay between characters sent to the host. This is necessary on some systems to prevent overloading the communication hardware on the host. *n* is a number from 0 to 65535 in units of 10 microseconds.

DELAY/S sends the current DELAY and LINE DELAY settings to the host.

See also LINE DELAY.

DELETE *filename*

DELETE/N *filename*

DELETE/S *filename*

Deletes a PC file. Anzio will display an error message if the file does not exist.

DELETE/N does not display an error message if the file is not found.

DELETE/S also sends a result code to the host:

00	Completed successfully
01	Error occurred

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

DIAL *phone\_number* [*wait retries*]


Tells the modem to call *phone\_number* to initiate a new connection.

The recommended way for Anzio to communicate with a modem is with the communication type set to "modem". However, for historical reasons, it is still possible to communicate with a modem (although not some newer modems) as serial devices. For that approach, Anzio allows you to specify additional parameters, so that if the connection is not successful on the first try, it will keep trying *retries* times, waiting *wait* tenths of a second between retries.

DIR [*pathname*]

DIR/S [*pathname*]

Displays all file names in the current LOG directory, or the *pathname* directory if specified.

You can move the highlight bar up or down to see a file's size and creation date at the bottom of the display. If you move the highlight to a subdirectory name and press , that subdirectory's files are displayed.

Note that you can print the displayed directory information using the PRINT command.

DIR/S also sends the selected file name either to the host or to another function. See *DIR/S Example*, page 56.

DOWNLOAD-DIR [*pathname*]

Sets *pathname* as the default directory used for downloads and file captures.

If *pathname* is not specified, cancels any previous DOWNLOAD-DIR or DOWNLOAD-LOCK.

DOWNLOAD-DIR does not prevent files from being created explicitly in other directories. See DOWNLOAD-LOCK.

DOWNLOAD-LOCK [*pathname*]

Forces all downloads and captures to be in *pathname*.

If *pathname* is not specified, cancels any previous DOWNLOAD-DIR or DOWNLOAD-LOCK.

EJECT

Ejects a page from the printer and closes the print job. See also FLUSH.

END | E

Stops Anzio. Same as STOP. Can be abbreviated as E.

ENV/S *variable*

Sends the value of the DOS/Windows environment *variable* to the host, terminated by a RETURN. Or, this variable could be an internal Anzio environment variable — see ENVIRONMENT.

ENVIRONMENT *string*

Sets Anzio's internal environment *string*. This string is sent to the host system at the beginning of a telnet or SSH session. Typically used with UNIX systems. Note that some telnet daemons or SSH daemons will pass on only a limited set of environment variables, such as DISPLAY, LPDEST, and/or PRINTER.

*string* is a list of environment variables and their values, separated by semicolons. The format for each variable is:

*name=value*

or

*name=\${env}*

---

➤ The curly-brace ("{" ) character has a special meaning inside a macro. So if you wish to include an environment variable inside a macro, it must be in form:  
    \${value}

---

Where *value* is a number, string, or nothing (null).

The second format is used to access the current value of the PC environment variable named *env*.

Anzio also provides several of its own internal environment variables, which can be used as well:

ANZ_COMMTYPE	The kind of connection, such as "SSH://hostname", "TELNET://hostname", "RLOGIN://hostname", "MODEM://phonenum", or "SERIAL://".
ANZ_MAC_ADDRESS	The MAC address of the network card which Anzio is using.
ANZ_HOSTNAME	Host name
ANZ_IP	PC's IP address
ANZ_PASSWORD	Value set by PASSWORD
ANZ_USERNAME	Value set by USERNAME
ANZ_LAST_REC'D	Name of the last file Anzio received
ANZ_TITLE	Title of the Anzio window
ANZ_WINDIR	The directory that contains Windows
ANZ_SYSDIR	The "System" directory
ANZ_CURDIR	Anzio's current working directory
ANZ_DOWNDIR	Anzio's download directory
ANZ_PROGRAM	"ANZIOWIN" or "ANZIOLITE"
ANZ_MY_DOCS	The folder name equivalent to "My Documents".

ANZ_COMMON_DOCS	The folder name equivalent to "Shared Documents", such as "c:\Documents and Settings\All Users\Documents". Not available on all Windows installations.
ANZ_DESKTOP	The folder name equivalent to the Windows desktop.
ANZ_PROGRAM_FILES	The folder name where programs are normally stored, such as "c:\Program files".
ANZ_PROGRAM_DIR	The folder that contains the currently running Anzio program.
ANZ_PROGRAM_EXE	The full pathname of the Anzio executable program.
ANZ_COMPUTERNAME	The NETBIOS name of this PC
ANZ_WINNAME	The username by which the user logged into Windows
ANZ_TCPNAME	The TCP/IP name of this PC
ANZ_LAST_LAUNCH	The status code of the last program launched by Anzio:  "running", "error", or the numeric code given when the program quit.
ANZ_HIGHLIGHT	The text that the user has highlighted with the mouse, on the Anzio screen.
ANZ_CURSOR_X	The X location of Anzio's cursor (starting at 1).
ANZ_CURSOR_Y	The Y location of Anzio's cursor (starting at 1).

ANZ_DROP_NAME	The name of the file dropped onto Anzio's Window.
ANZ_DROP_X	The X location of the point where a filename was dropped onto Anzio's window.
ANZ_DROP_Y	The Y location of the point where a filename was dropped onto Anzio's Window.
\$	A literal "\$". This "\${}\$" will return an actual dollar sign.
^c	Where <i>c</i> is a character, this returns the control-equivalent of that character. So for instance "\${^a}" will return a control-a.
<i>nn</i>	Where <i>nn</i> is numeric, this will return a character with the decimal value of <i>nn</i> . Values above 127 are treated as Unicode.
<i>Hnn</i> or <i>Xnn</i>	Where <i>nn</i> is a hex number, this will return the character with the hex value of <i>nn</i> . Values above 7F are treated as Unicode.
<i>Onn</i>	Where <i>nn</i> is an octal number, this will return the character with the octal value of <i>nn</i> . Values above 177 are treated as Unicode.
<i>Name as SET</i>	Any user-defined variable that has been set with the SET command will return that variable's value.

EVENT *eventname macro*

The EVENT command tells Anzio to execute a certain macro when a certain event occurs. The *macro* can be an alphanumeric key, or the name of the key, such as "F10". The *eventname* can be one of the following:

ONSTARTUP	Executes when Anzio starts up. This is equivalent to the STARTUP-MACRO command.
ONERROR	Executes when an error occurs in command processing. Equivalent to the ONERROR command.
ONDROP	Executes when the user does a drag-and-drop operation and drops a filename on the Anzio window. The variable ANZ_DROP_NAME will contain the filename dropped. The variables ANZ_DROP_X and ANZ_DROP_Y will contain the x and y coordinates (in character units, based on 1) where the drop occurred.

The *macro* parameter can be an alphanumeric key, or it can be the key name, such as "sf10" for shift-F10.

EXPECT *value1 value2*

Anzio will evaluate *value1* and *value2*, expecting them to be equal strings. If they are not, an error condition exists; if an ONERROR macro is in effect, it will be executed, otherwise a message will be displayed. Note that either value might be derived from an embedded function.



FILL *c1 c2 r1 r2 [char]*

Fills a rectangular area of the screen with spaces, or the optional character *char*. The offsets may be absolute or cursor-relative, for example, FILL 1 1 .+5 .+5.

FIND/S *filespec*

Finds the first file or directory name matching the name or wildcard entry in *filespec* and sends it to the host. If it is a directory, Anzio will add a backslash ("\") to the end.

If there are no matching files, Anzio sends two backslashes ("\\"). However, if an ONERROR macro is in effect, that will be triggered instead. See also FINDNEXT/S.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

FINDNEXT/S

Finds the next file or directory name matching the wildcard entry in the previous FIND/S command, then sends the name to the host. If it is a directory, Anzio will add a backslash ("\") to the end.

If there are no matching files, Anzio sends two backslashes ("\\"). However, if an ONERROR macro is in effect, that will be triggered instead. See also FINDNEXT/S.

FLUSH

Sends any remaining data in the print buffer to the printer, and closes the print job. If spooling is enabled on the printer (network or Windows local), this will release the print job for printing. See also FLUSHTIMER.

This command will eject the page only if PRINT LEVEL is High. See also EJECT.

FLUSHTIMER *time*

Sets the number of seconds between FLUSHes (default 5 seconds). After *time* seconds have elapsed since Anzio sent something to the printer, Anzio calls the FLUSH command to allow the data to be printed.

To disable this feature, set *time* to 0. For example, you may need to print two screen dumps per page, or your passthrough print data may take longer than *time* to arrive.

FONT [*name*] [*size*] | LARGER | SMALLER | BOLD

Sets the displayed Windows screen font name and size. Anzio will change the window size to follow the font size, unless the window is zoomed.

<i>name</i>	Font name, enclosed in quotes if it contains spaces, such as "Courier New".
<i>size</i>	Font size, as either <i>height</i> or <i>heightxwidth</i> .
LARGER	The next larger font size, for example, from 10 to 12.
SMALLER	The next smaller font size.
BOLD	Selects a bold font.

Note that your PC may not have the indicated font and size.

FONT-ALT *fontnames*

Tells Anzio what alternate fonts should be used for the screen, to display characters not found in the primary font. The *fontnames* can be a list of fonts by their Windows names, separated by commas. Fonts will be tested in the order indicated. To tell Anzio NOT to use a particular font, put a minus sign in front of its name. The last entry should generally be "auto", telling Anzio to look through any other fonts on the Windows system. The default is simply "auto".

FTP [PASSIVE]

FTP NEW

These commands initiate an *interactive* File Transfer Protocol (FTP) session over a network connection. They require WININET.DLL, which comes with Internet Explorer and some versions of Windows. The dialog box provides an easy point-and-click interface that allows you to move files easily between the PC and a host system, which might or might not be the same host as you are telnetting to.

The command

FTP

will initiate an FTP connection to the same host, the same username, with the same password as you are currently telnetting. If you include the word "PASSIVE", then the connection will be established in FTP passive (PASV) mode, which might be required by a firewall at your site.

The command

FTP NEW

will prompt you first for the hostname, port, username, and password for the connection. It also allows you to indicate that you need a passive (PASV) connection, via a check box.

FTPDELETE *hostfile*

FTPDELETE/S *hostfile*

Causes Anzio to use FTP commands to delete a file on an FTP server.

*hostfile* can be a name, a pathname (with either kind of slash), or a URL matching:

*username[:password]@hostname[:port]/filepath*

(note that you can't do *username* and *filepath* but not *hostname*).

The "/S" causes a status result to be sent to the host ("00" means success).

```
FTPGET [ASCII] hostfile [pcfile]  
FTPGET/S [ASCII] hostfile [pcfile]  
FTPPUT [ASCII] hostfile [pcfile]  
FTPPUT/S [ASCII] hostfile [pcfile]
```

These commands do *programmed* File Transfer Protocol (FTP) transfers over a network connection. They require WININET.DLL, which comes with Internet Explorer and some versions of Windows. They allow AnzioWin to work as an FTP client, although commands can be issued from the host (through Anzio's usual command protocol). Commands assume that the host name, username, and password for the FTP session are the same as for the Anzio telnet connection. Thus the command can be as simple as

```
FTPGET <filename>
```

*hostfile* can be a name, a pathname (with either kind of slash), or a URL matching:

```
username[:password]@hostname[:port]/filepath
```

(note that you can't do username and filepath but not hostname). The "ASCII" modifier causes an ASCII (text mode) transfer. If *pcfile* is not given, the PC file name will be derived from the basename (name without directories) of the *hostname*.

The "/S" causes a status result to be sent to the host ("00" means success). Anzio's DOWNLOAD DIRECTORY will affect downloads (FTPGET[/S]).

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
FULL [DUP]  
HALF [DUP]
```

Sets the duplex mode:

```
FULL DUP Anzio expects the host to echo back each  
keystroke to the screen.
```

**HALF DUP** Anzio displays each keystroke sent to the host.

If you are getting double characters, you need to set FULL DUP. If you are not getting characters on the screen at all, you need to set HALF DUP.

**GAUGE** [ON] | OFF | TOGGLE

Controls the gauge line on the bottom of the screen. The gauge line displays '. . . . | . . \* . 1 . \* . . | . . . . 2 . . . .' to help you locate columns. The gauge line also shows the cursor line and position.


**HALF** [DUP]


Sets Half Duplex mode. See FULL DUP.

**HANGUP**

Hangs up and terminates a modem connection.

**HELP**

The HELP command displays an old DOS style help screen with many available commands, any open file names, and the current option settings. Press  to exit the HELP screen and return to your "working" screen.

In the HELP screen, you can enter a command without pressing . After this command is processed, either the HELP screen is redisplayed, or you will return to the working screen.

Note that the PRINT command will print the HELP screen, not the working screen.

**HEXPATCH** *address bytes*

No longer used.

HOLD [ON] | OFF | TOGGLE

Suspends (holds) output from the host to the screen. The HOLD TOGGLE command can be attached to a macro key. See also SCROLL-LOCK.

HOSTNAME/S

Causes Anzio to send its current host name to the host. The host name can come from a command line parameter, a telnet address, or from the network parameters.

IGNULL [ON] | OFF | TOGGLE

Tells Anzio whether to ignore nulls (hex 00) in the incoming data stream. Should be IGNULL OFF unless instructed otherwise by Rasmussen Software.

INTERPRET

Displays the last 2048 characters received from the host. Non-printable characters appear as reverse-video hex numbers. This display is useful when something strange has occurred at the terminal, since it shows the input conditions.

You can view high non-ASCII characters (between hex A0 and FF) as characters in the screen's character set, as 7-bit characters, or as hex codes. Use the **[M]** key (for *mask*) to cycle through these view options.

Use PRINT to print the display.

INVOKE *macro*

Starts the indicated *macro*. If used inside a macro, the calling macro stops and the invoked macro continues. See also CALL.

JUMP OFF | MEDIUM | FAST

Tells Anzio how to scroll the screen for rapid incoming data:

OFF                      Shows every line.

MEDIUM	Repaints the screen when the display is a half-screenful behind.
FAST	Repaints only when the display is a complete screenful behind.

KCOMMAND *command* [*parameters*]

Sends a Kermit *command* to a host. The results of the *command* will depend on the host Kermit implementation. Anzio supports the following commands:

KERMIT *command*  
CWD *newdir*  
DIRECTORY *spec*  
ERASE *spec*  
FINISH *spec*  
HELP *spec*  
LOGIN *spec*  
JOURNAL *spec*  
COPY *file1 file2*  
LOGOUT *spec*  
MESSAGE *spec*  
PROGRAM *spec*  
QUERY  
RENAME *file1 file2*  
TYPE *filename*  
USAGE *param*  
VARIABLE  
WHO

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

KEEP [F7] [c1 c2 r1 r2]

KEEP/N [F7] [c1 c2 r1 r2]

KEEP sends the screen contents to the output file opened by OPENO. If you OPENO LST:, the printer will be your output file.

KEEP/N is the same, but does not append CR/LF to the output lines.

KEEP [F7] Starts a stretchable KEEP rectangle in the center of the screen. Use the arrow keys to move one corner of the highlighted rectangle to include the data you want to keep. Press [F7] to nail that end down. Move the other corner to cover the desired KEEP area. Press [F7] to display the selection coordinates on the command line.

c1 c2 r1 r2 Optional screen coordinates used to specify a part of the screen, either absolute or cursor-relative (as in BOX).

KEYS

Shows all currently defined keys. The first section is system defined (hard coded) keys. The second is key definitions either contained in the KEYS file or downloaded from the host.

Use PRINT to print the display.

KILL

Terminates the last program started by Anzio with a LAUNCH, LAUNCH/S, RUN, or RUN/S command.

KRECEIVE AUTO-OFF

KRECEIVE AUTO-ON

KRECEIVE [options] [filespec]

KRECEIVE/S [options] [filespec]

Receives one or more files from a Kermit host system, or controls Kermit auto-reception.




When Kermit auto-reception is on, a program running on the host system can send (download) a file to the PC without the user doing anything. This capability can be turned on or off with

```
KRECEIVE AUTO-ON
```


and

```
KRECEIVE AUTO-OFF
```

A file transfer is specified as:

```
KRECEIVE filename 
```

or

```
KRECEIVE wildcard 
```

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”.

---

➤ Options for Kermit, Xmodem, Ymodem, and Zmodem are shown here. Some options are not applicable to some transfer protocols.

---

Possible options are:

-A	ASCII mode transfer
-B	Binary mode transfer
-C	XmodemCRC & Kermit Checksum
-D	Debug
-G	Xmodem1KG & YmodemG
-K	Xmodem1K & Kermit CRcK
-O	Zmodem WriteClobber
-U	Uppercase
-W <i>xxxx</i>	Windowing and window size
-1	Kermit CRC
-2	Kermit Checksum2
-3	Kermit CRC32

- 8 Zmodem 8K
- V Debug

If “S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. Possible result codes are:

- 00 Success
- 01 at least 1 file transferred, but at least one was not (not implemented yet)
- 02 file not found
- 03 file already exists
- 04 file skipped
- 05 cancelled by user
- 06 cancelled by host (not implemented yet)
- 07 transfer failed (started ok, but then failed for some reason)
- 08 protocol error
- 09 protocol timeout (not implemented yet)
- 10 critical unknown (not implemented yet)

See also KCOMMAND.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

KSEND [*options*] *filespec*  
KSEND/S [*options*] *filespec*

Sends one or more files to a Kermit host system. The *filespec* can be either a single filename or a wildcard.

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For a description of available options, see KRECEIVE, above.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes see KRECEIVE, above.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
LAUNCH [options] program [parameters]  
LAUNCH/S [options] program [parameters]
```

Starts another Windows program to run alongside Anzio. Similar to RUN, but Anzio does not wait for the other program to complete. See also WINSTART.

Possible *options* are:

/MIN	Minimize the program's window
/MAX	Maximize the program's window
/HIDE	Hide the program's window
/nnn	The numeric value <i>nnn</i> is

equivalent to that used in VBScript, etc.

Both *program* and *parameters* can contain environment variables, coded as

$\${var}$

Anzio will replace each environment variable with its contents. These can be DOS/Windows environment variables, or Anzio pseudo environment variables such as ANZ\_IP, as explained under ENVIRONMENT.

If LAUNCH/S is used, Anzio will send a status return code back to the host, indicating whether the *program* started successfully. See RUN/S.

LINE [DELAY] *n*

Sets the line turnaround delay. After an “unlock” is received from the host, Anzio delays its next transmission for the specified amount of time. *n* is a number from 0 to 65535 in units of 10 microseconds.


The “unlock” protocol depends on the host operating system.

If you are experiencing lockups, or problems with file transfer, particularly with NCR I-systems, you may need to experiment with LINE DELAY and DELAY settings.

See also DELAY.

LOCK [ON] | OFF | TOGGLE



LOCK ON is used for the normal NCR I-system protocol, which ‘locks’ the keyboard when the host is not expecting entry. To temporarily unlock the keyboard, press .

LOCK OFF sends all characters as they are entered, never locking the keyboard. LOCK OFF is required by some programs, such as free-standing utilities and modems.

**(UNIX)** Use LOCK OFF with UNIX systems.

LOG [*drive:*] *directory*

Sets the default (“logged”) disk drive and directory. The logged disk and directory are used for local file operations (DIR, OPENO, etc.) when no drive or directory is specified.

The optional disk unit must be followed by a colon.

Synonym for the CD command.

LOGIN-WIZARD [ON] | OFF | TOGGLE

Turns on or off the Login Wizard. When Login Wizard is ON, and a USERNAME and PASSWORD are set, Anzio will attempt to log you in as soon as the connection is made.

MAPISENDFAX *phone filename [profile]*

Uses Windows' messaging system (MAPI) to send a fax, to the phone number contained in *phone*. Windows decides how to "print" *filename* to the fax driver based on its file extension.

If a *profile* is included, and it matches a profile configured in MAPI, then MAPI will not prompt the user for any details, but will send the fax directly.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

MCISENDSTRING *command*




MCISENDSTRING/S *command*

These commands provide an interface to Windows' Multimedia Control Interface, via the Windows command "mciSendString". The MCISENDSTRING Anzio command gives NO feedback on success; on an error, it will trigger an ONERROR macro if one is defined, or display "Function Error". The MCISENDSTRING/S version will send either "OK" or an error message back to the host.

Possible values of *command* are (somewhat) documented by Microsoft. Here is an example that records sound to a file. To initiate recording, do

```
MCISENDSTRING open new type waveaudio alias
capture
MCISENDSTRING set capture time format ms
bitspersample 16 channels 2 samplespersec
44100 format tag pcm
MCISENDSTRING record capture
```

To terminate recording and save the sound to a file, do:

```
MCISENDSTRING stop capture   
MCISENDSTRING save capture test.wav   
MCISENDSTRING close capture 
```

```
MENUBAR c1 c2 r1 r2 c3 c4 off len [cols]
```

Superimposes a sliding menu bar system over the data presently on the screen, in one or more rectangles: (*c1, r1*) to (*c2, r2*) and (*c3, r1*) to (*c4, r2*). If a single-column menu is needed, just make *c3* equal *r1* and *c4* equal *c2*. If more than two columns are needed, use the *cols* parameters.

When the user has selected an entry, Anzio returns the contents of the screen at offset *off* and length *len* of the item selected.

See *MENUBAR Example*, page 57.

```
MENULANGUAGE xxx
```

Sets the language that Anzio uses for its dialog boxes, menus, and error messages. Support for languages is provided via files (in the same directory as Anzio) named ANZIOzzz.DLL, where “zzz” is an abbreviation of the language, such as ANZIOSPA.DLL for Spanish. These files are provided by Rasmussen Software and partner companies, and can not generally be created by users.

The *xxx* language identifier can be a) the native name of the language, such as “español”; b) the name in the language of the Windows system, such as “Spanish”; or c) the Windows standard number for the language, such as “10” for Spanish. Entering zero, or encountering any error, will switch Anzio back to its built-in English mode.

```
MERGE filename
```

Merges the key macros in *filename* with the macros in memory. Like READ, but does not clear existing macros, nor change the current key file name. If you SAVE after the merge, the merged set of macros is written to the current key file.

Macros in *filename* will overwrite macros in memory only if they have the same name.

MESSAGE *string*

Pops up a message box containing *string* and an OK button – useful in a defined key or a host command.

MKDIR *dirname*

MKDIR/S *dirname*

Creates a new directory.

MKDIR/S also sends a result code to the host:

00	Completed successfully
01	Error occurred

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

MONITOR [ON] | OFF | TOGGLE

Sets monitor (diagnostic) mode. MONITOR ON displays all characters received from the host, including screen control codes. Non-printing characters are displayed as ASCII hex values.

MONITOR OFF returns to normal Anzio terminal emulation.

See also INTERPRET.

```
OLE INIT program.title
OLE GET program.property
OLE program.property=value
OLE program.method [args]
OLE/S program.method [args]
OLE TERM
```

These commands provide an interface into the WINDOWS OLE system for calling and manipulating other programs. In Anzio, only one OLE object can be in use at a time. Note also that property names and method names must include the *program* part of the object name.

Begin an OLE session with

```
OLE INIT program.title
```

To get a property value, and return it to the host, do

```
OLE GET program.property
```

To set a property, do

```
OLE program.property=value
```

To call a method when you are not expecting a return value, do

```
OLE program.method [args]
```

where *args* are any arguments expected by the method.

To call a method that returns a value, and send that value to the host, do

```
OLE/S program.method [args]
```

where *args* are any arguments expected by the method.

To terminate the session, do

```
OLE TERM
```



ONERROR *macro*  
ONERROR

This command tells Anzio how to handle errors in macro processing. Note that these are operational errors, not syntax errors. When an error occurs, the current macro (and any macros nested with the CALL command) is terminated, and *macro* is executed. For instance, if a macro:

```
[F6] onerror E | [F6] WAITFOR Login 5 | guest |
```

is running, and the WAITFOR times out, the macro will be terminated, and the "E" macro will be run. Setting the ONERROR macro lasts only for the current program run; it is NOT retained in the parameters file. To return to normal default error handling, do just

```
ONERROR
```

ONSTARTUP *macro*  
ONSTARTUP

Specifies a *macro* to be run on startup. The *macro* specified can be an alphanumeric name, or the keyname of the key such as "F12". This information is stored in the settings file, and will take effect the next time Anzio is run. To clear the setting, so NO macro will be run, do:

```
ONSTARTUP
```

A startup macro can also be specified in the command line, and that will take precedence over the ONSTARTUP macro, even if it doesn't exist.

If a startup macro is specified (by either method), the Login Wizard will not function. Your macro must then do whatever logging in is necessary.

OPENE *filename*  
OPENE/N *filename*  
OPENE/S *filename*

Opens a PC file for output from the host, like OPENO (and variations) below, except in EXTEND mode. That is, new data will be added to the end of the existing data in the file.

If *filename* does not already exist, and OPENE was used, that constitutes an error; an ONERROR macro will be triggered if one has been specified, otherwise an error message is presented.

If *filename* does not already exist, and OPENE/N was used, a new file will be created.

OPENE/S will send the following result codes to the host:

00	Completed successfully
01	File not found
02	Output file already open
03	File location is not in DOWNLOAD- LOCK directory, or other file error

See OPENO for more information.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

OPENI *filename*  
OPENI/S *filename*

Opens a PC file for input to the host. If *filename* does not include a drive or directory name, the default drive and directory names are used (see LOG).

Only one input (and one output) file can be open at a time.

The file name mentioned can be a serial port, such as "COM2". Note that in some versions of Windows, if *any* printer driver is assigned to this serial port, it will interfere with Anzio accessing the serial port in this way.

OPENI/S also sends a result code to the host:

00	Completed successfully
01	File not found
02	Input file already open

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

OPENO *filename*  
OPENO/N *filename*  
OPENO/S *filename*

Opens a PC file for output from the host. If *filename* does not include a drive or directory name, the DOWNLOAD-DIR directory will be used if defined, otherwise the default drive and directory names are used (see LOG).

Only one output (and one input) file can be open at a time.

The output file is used for:

- CAPTURE – see *Data Capture*, page 80
- KEEP
- PICK
- WRITE
- File transfer – see *File Transfer*, page 83
- Passthrough printing – see *Passthrough Printing*, page 72

*filename* may be a DOS device name, such as COM2, or LPT2. If *filename* is LST:, output is sent to the printer set with the PRINTER command. Use OPENO LST: for file transfer from a host computer directly to the PC's printer.

A message is displayed if the file exists. Use DELETE to delete that file, or OPENO/N.

OPENO/N overwrites the file if it exists, without displaying a message.

OPENO/S sends a result code to the host, without displaying a message on the PC:

00	Operation successful
01	File already exists
02	An output file is already open
03	File or subdirectory error (including violation of DOWNLOAD-LOCK)

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

PALETTE BG | BLBRIGHT | BLDIM | ULBRIGHT | ULDIM | FULL

Sets the palette. At any one time, Anzio can use 8-bit attributes or "full" (10-bit) attributes. With full attributes, Anzio can control 16 foreground colors, 16 background colors, blink, and underline. With 8-bit attributes, Anzio can display characters in a palette of 256 color/attribute combinations. For instance, it is possible to display blinking characters, or underlined characters, but not at the same time. This is often necessary for compatibility with earlier hardware designs. This command specifies which palette is in effect:

BG	Bright and dim backgrounds
BLBRIGHT	Blinking, with bright backgrounds
BLDIM	Blinking, with dim backgrounds
ULBRIGHT	Underlines, with bright backgrounds
ULDIM	Underlines, with dim backgrounds
FULL	Full palette of 16 foreground colors, 16 background colors, blink, and underline.

PARITY EVEN | ODD | MARK | SPACE | OFF

Sets the parity for communication.

PASSTHRU

Puts Anzio into passthrough print mode, just as though it had received the appropriate escape sequence from the host.

PASSTHRU-CP *name*

PASSTHRU-CP/S *name*

Sets the encoding scheme used for passthrough printing. Options for *name* are the same as for the CHARSET command, plus the following option:

NONE	Results in ISO or OEM depending on the terminal type.
------	---


This setting has an effect only when printing with Print Level set to High, or when using Print Wizard.

PASSTHRU/S sends a result code indicating whether it succeeded: 00 for success, 01 for failure. A failure could indicate that a particular codepage was not installed on this PC, for instance.

See also CHARSET.

PASSWORD *text*

PASSWORD/S

Sets or sends the host password. *text* is stored internally in an encoded format. Use a pipe character (“|”) to represent a password of .

The stored password is used by the Login Wizard, and is sent to the host with PASSWORD/S.

PASSWORD/S sends the password to the host. The password was either set by the PASSWORD command, or as part of a command line parameter containing a URL.

See also USERNAME.

PASTE

Pastes text data on the Windows clipboard to the host. The CLIP command clips a region of screen text and puts it on the clipboard.

PICK *c1 c2 r1 r2 type* [...]

Picks data off the screen and puts in the output file (see OPENO) for use by PC programs that understand comma-separated-values format. After the host displays a table, PICK can capture one or more columns of numbers or labels off the screen. The picked data is written to the current output file in CSV format.

Offsets can be cursor-relative – see BOX.

PITCH *n* | -1

Sets the type of sound that the PC will make when it receives a beep signal (control-G, hex 07) from the host.

➤ If your PC has a sound card but does not have speakers connected, Windows does not know that. Your only option is to force the beep to come from the PC motherboard's speaker.

PITCH -1 causes the beep to come from the PC motherboard speaker, rather than an installed sound card.

A value of *n* higher than 99 tells Anzio to try beeping at the indicated pitch. This will NOT work in Windows 95/98/ME.

Values between 0 and 64 cause Anzio and Windows to generate certain sounds, identified by name, as configured in the "Sounds" control panel applet. These values are:

0	Default sound
16	Critical stop
32	Question
48	Exclamation
64	Asterisk

PLAYMEDIA

PLAYMEDIA *filename*

PLAYMEDIA *filename* [CELL] *x1 y1 x2 y2 [options]*

Plays a multimedia file, such as an MP3, WAV, MID, or AVI file named *filename*, which can be a local file or a URL.

Only one media file can be played at a time. If a PLAYMEDIA is executed while a previous file is playing, the first playback is terminated. To stop the playback, use the command without a filename:

PLAYMEDIA

If the media file is video, such as an AVI file, you can optionally specify a location in the Anzio window for it to display, with

PLAYMEDIA *filename* [CELL] *x1 y1 x2 y2*

If CELL is given, then coordinates are in character rows and columns, starting at 1. If CELL is not given, coordinates are in pixels, relative to the top left corner of the top left character. For *x1* and *y1*, a value of -1 means the left or top edge. For *x2* and *y2*, a value of 0 means the right or bottom edge. If all four parameters are greater than zero, the video will be stretched, while maintaining aspect ratio.

If any text follows the *y2* above (that is, *options* are specified), that text will be passed on to the Windows `mciSendString` command. Examples are "repeat", "at time xxx", "from position yyy", etc. See Windows documentation of `mciSendString` for details.

This command is aborted by Grand Abort (Alt-A).

PLAYSOUND *filename*

Plays a WAV sound file named *filename*, which can be a local file or a URL.

Arguments can also contain `${variable}` where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

PORT *n*

Specifies the hardware communication port number – see *Configuring the Serial Port*, page 12.

PREVIEW [ON] | OFF | TOGGLE

Turns on or off Print Preview, which is part of the Print Wizard feature.

PRINT [*c1 c2 r1 r2*]


PRINT/N [*c1 c2 r1 r2*]

Prints the lines visible on the screen, ignoring video attributes. PRINT/N prints each line without adding carriage return-linefeed characters.

The optional column and row offsets define the screen area to be printed, from column *c1*, row *r1* to column *c2*, row *r2*. Offsets can be cursor-relative – see BOX.

If columns and rows are not given, Anzio will print the entire screen, unless the screen has a section highlighted, in which case it will print only the highlighted text.

To print the screen during a REVIEW, enter P.

To print displays generated by DIR, HELP, INTERPRET, KEYS, or TYPE, enter PRINT .

PRINTER *name*

PRINTER WPRN

Identifies the printer or output driver used for the PRINT and PRINTER-SETUP commands, and for passthrough print.

PRINTER WPRN tells Anzio to use its Windows printer driver.



If *name* is a file, that file is opened for output. If the file exists, it is overwritten.






If *name* contains the string "://", it is treated as a SUPERFILE designation, such as "PDF://" to specify creation of an Adobe PDF file.

PRINTER-SETUP  
PRINTER-SETUP *text*






Sends characters directly to the printer.

If no *text* is specified, the keyboard is used, so that anything you type is sent to the printer (like a typewriter). You will need to follow each  with a linefeed character (Ctrl-J). You may also enter printer-specific control codes to set compressed pitch, etc. Non-printable characters are displayed as ASCII hex. Press  to end keyboard input and exit this command.



*text* may also include control characters. To embed a control character in *text*, preface it with  . For example, to send  Ctrl-N to your printer:

PRINTER-SETUP       N 

As another example, suppose you want to define the F12 key to print the screen, and then eject a page. Use the PRINT command to send the screen to the printer, and then send the printer a form-feed character (Ctrl-L):

```
DEFINE F12  PRINT |  
 PRINTER-SETUP   | 
```

Note that a vertical bar | ends each command.

To embed any hex character in *text*, enter   followed by hex 10 and the two-character ASCII hex value of the character. The hex 10 represents a second Ctrl-P, used internally for quoting ASCII characters. For example, to embed a Return character (hex 0D), enter:

```
  100D
```

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

PRINTFILE [/charset] filename

Copies *filename* to the current PRINTER. *filename* can be a local file or a URL, causing Anzio to fetch the file from a web or FTP server.

The file is assumed to be in the OEM ("DOS") character set. This can be overridden with a *charset* value, following a slash, as follows:

/I	ISO character set
/U	Unicode UTF-8
/O	OEM

If you are using the Windows printer driver (PRINTER WPRN) and Anzio's Print Wizard is enabled, the Print Wizard examines the data in *filename* and sets the line spacing, character size, and margins to make the document fit on the page.

If the *filename* has an extension indicating it is a bitmap (.BMP, .JPG, .JPEG, or .GIF), the file will be printed as a fullpage bitmap, as large as possible on the page.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

PRINTFONT [*name*] [*size*] [BOLD]

Used with the Windows printer driver (PRINTER WPRN) to set the printer font name and size. Enclose *name* in quotes if it contains one or more spaces. *size* is either *height* or *heightxwidth* (specified in points). Specifying BOLD selects a bold font.

PRINTFONT-ALT *fontnames*

Tells Anzio what alternate fonts should be used for the printer, to print characters not found in the primary font. This setting applies only when printing with Print Wizard. The *fontnames* can be a list of fonts by their Windows names, separated by commas. Fonts will be tested in the order indicated. To tell Anzio NOT to use a particular font, put a minus sign in front of its name. The last entry should generally be "auto", telling Anzio to look through any other fonts on the Windows system. The default is simply "auto".

PRINTINIT *filename*

Specifies a print-initialization file. When Anzio receives a passthrough print job, if this file exists, it is inserted at the beginning of the print job. The file can be built with a text editor, such as Notepad, and can contain text to print, escape sequences for printer control, or Print Wizard Markup Language. If the named file does not exist, no error will occur, but it will be noted in the Debug Printing window, if enabled.

PRINTLEVEL WIZARD | HIGH | LOW | SPOOLER | RAW

Used with the Windows printer driver (PRINTER WPRN) to set the print level.

PRINTLEVEL WIZARD turns on the Print Wizard

PRINTLEVEL HIGH indicates high-level print.

PRINTLEVEL LOW indicates low-level print.

PRINTLEVEL SPOOLER indicated spooler-level print.

PRINTLEVEL RAW indicates raw-level print.

PRINTLINE [*text*]

PRINTLN [*text*]

Writes *text*, if specified, to the printer. Then writes a return and a linefeed to the printer.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

PRINTLOW [ON] | OFF | RAW | SPOOLER

Used with the Windows printer driver (PRINTER WPRN) to set the print level. See also PRINTLEVEL.

PRINTLOW ON indicates low-level print.  
PRINTLOW OFF indicates high-level print.  
PRINTLOW SPOOLER indicates spooler-level print  
PRINTLOW RAW indicates raw-level print.

PRINTMACRO [*macro*]

Specifies a code-activated-switch macro. The stream of print data coming from the host in passthrough print can contain control characters control-A (hex 01) through control-F (hex 06), which can trigger Anzio to initiate one of six macros starting at *macro*. For instance, if you do

```
PRINTMACRO a
```

then when the passthrough print data contains a control-A, Anzio will CALL the “a” macro; when it contains a control-B, Anzio will call the “b” macro, etc., up to control-F. When Anzio calls the macro, it exits from the passthrough print routine.

Typically, these macros will each switch to a different printer, and then use the PASSTHRU command to return to doing passthrough print. This way, control codes embedded in the data stream can effectively switch from one printer to another.

PRINTTOFILE [ON] | OFF | *filename*

Tells Anzio whether to print to file. This is equivalent to checking the “Print to file” box in the Printer Setup dialog. However, with this command you also have the option of specifying the *filename* to be printed to, so the operator will not be prompted for that.

Note that this file will contain printer control codes as included by the Windows printer driver.

PROMPT-ID

Displays a dialog box to prompt the user for login and password, with no default values. Values entered are stored in Anzio's internal username and password variables, and are stored in the settings file. Values can be used by the Login Wizard at the time of the next connection, or they can be used in macro-programmed logins, using the commands USERNAME/S and PASSWORD/S.

If the user clicks Cancel, and an ONERROR macro is defined, it will be started.

PROMPT/S [*caption* [*text*]]

Prompts the user for a string of alphanumeric input, using a Windows dialog box. The result is sent to the host, or nested into another command.

If a *caption* is given, it will appear in the top line of the dialog box. If *text* is also given, it will appear above the user input prompt.

PURGE

Clears Anzio's "received file" buffer. This buffer gets data from either CAPTURE or a file transfer.

Data in this buffer is waiting to be stored in an output file. If there is no output file (see OPENO), the buffer will contain a backlog of data. The HELP screen displays the buffer status.

RAW-SETUP [DEFAULT]

This command allows you to specify a particular port (parallel or serial) to be used whenever Anzio prints at the RAW level, and to configure that port.

RAW-SETUP first prompts for the name of the port. If this field is empty, Anzio will derive the port name from what is shown under File/Printer Setup. Or, you can enter a port name such as "LPT2" or "COM1".

Then, if the port as specified or derived is a comm port, Anzio will display a dialog box allowing you to specify baud rate, parity, etc.

To tell Anzio to use the printer specified in Printer Setup for Raw printing also, do

```
Raw-setup default
```

READ *filename*

Reads in a file of defined keys and macros, replacing those in memory. *filename* becomes the new default key file name used by SAVE.

See also MERGE.

RECEIVE CODED

Used when transmitting coded files between PCs.

RECEIVE QUIET [ON] | OFF | TOGGLE

Controls whether file transfer data and host commands are displayed.

RECEIVE QUIET ON does not display transfer data, and so is slightly faster for file transfer. This setting also affects functions sent from the host.

RECEIVE QUIET OFF is the default, and all transferred data is displayed on the screen.

RECONNECT [ON] | OFF | STAY

Specifies reconnection behavior when the host system drops its end of the connection, for example, after you log off from a UNIX host. Does not apply to direct serial connections, but does apply to modem (TAPI) connections.

RECONNECT ON causes Anzio to try a new connection, leading to a new login prompt.

RECONNECT OFF causes Anzio to exit when the connection ends.

RECONNECT STAY causes Anzio to keep running, but in a disconnected state.

RECORD [ON]  
RECORD CANCEL  
RECORD OFF  
RECORD *keyname*

These commands control Anzio's macro recording capability. To begin recording a macro, do simply RECORD. Anzio will start recording and saving your keystrokes. To cancel recording, and discard saved keystrokes, do RECORD CANCEL.

If you do RECORD OFF, Anzio will stop recording. It will prompt you for the key you wish to assign the saved keystrokes to.

If you do RECORD *keyname*, Anzio will stop recording and assign the macro to the key identified by *keyname*.

REGTELNET [RESET]

Anzio can register itself to Windows as the official telnet client. Then, whenever the user launches a URL that starts with telnet://, such as from a web page, Windows will start Anzio. This is also known as "browser integration".

REGTELNET sets this version of Anzio as the official telnet.

REGTELNET RESET sets Windows back to its default state.

RENAME *oldname newname*

Renames a PC file from *oldname* to *newname*.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

REPLACE/S *search-pattern replace-pattern content*

Does a "regular expression" replacement on *content*, and returns the result to the host or to a nesting command.

RESET

Resets some terminal emulation parameters: character sets, attributes, and wrap mode.

RETRANSMIT

Resends the last function response code, or, during file transfer, resends the last file record transmitted.

REVIEW

Allows you to see old screen data that has scrolled off the top of the screen. See *Using Review Mode to See Previous Data*, page 40.

RUN [*options*] [*program* [*parameters*]]

RUN/N [*options*] [*program* [*parameters*]]

RUN/S [*options*] [*program* [*parameters*]]

Runs another Windows or DOS *program*. When *program* ends, control returns to Anzio, with the screen intact. For example, you can change from terminal emulation to word processing and then back.

Before restoring the screen, Anzio prompts you to press any key. This is so you are able to see *program's* output if necessary.



Anzio starts *program* and then goes into an inactive state until *program* finishes, for compatibility with DOS versions of Anzio. Use LAUNCH or WINSTART to run a concurrent program with Anzio.

RUN/N does not require a keystroke to restart Anzio after *program* ends.

RUN/S also sends *program's* exit code to the host.

Possible *options* are:


/MIN	Minimize the program's window
/MAX	Maximize the program's window
/HIDE	Hide the program's window
/ <i>nnn</i>	The numeric value <i>nnn</i> is equivalent to that used in VBScript, etc.

If you do not specify *program*, the RUN commands call the DOS command interpreter (C:\COMMAND.COM). You can then enter any DOS commands. Use EXIT to exit DOS and return to Anzio.

Some programs must be started in their home directory. Use LOG to change to the required directory before using the RUN command.

*program* can be a Windows or DOS command or an application. For applications, enter the application name including the ".exe" or ".COM" suffix.

If *program* does not include a suffix, Anzio calls COMMAND.COM to find *program*, somewhere in the PATH. For example:

```
RUN COPY MYFILE PRN 
```

To allow *program* to access the results of a recent file transfer, the output file must first be closed (see CLOSEO). The HELP screen will show you if you have an output file open.

Both *program* and *parameters* can contain environment variables, coded as

```
#{var}
```

Anzio will replace each environment variable with its contents. These can be DOS/Windows environment variables, or Anzio internal environment variables such as ANZ\_IP, as explained under ENVIRONMENT.

RZ

An alias for ZRECEIVE. See ZRECEIVE.

SAVE [*filename*]

Saves all currently defined keys (macros) to a PC file, overwriting the file's previous contents. If *filename* is not specified, Anzio will use the file which was read at startup, if any.

SCREENMODE *option*

Sets the screen driver. SCREENMODE/S, below, reports screen mode parameters to the host.

*option* is one of:

132	132 columns wide
80	80 columns wide
CxR	Sets the window to the indicated size, for example, 132x25. In a network connection, if the new size is supportable, sends the new size to the host.

SCREENMODE/S

Reports several screen parameters to the host, formatted as one line:

TMLLLCCCWWW*screenmode*

T (type)	C for color, M for mono, B for B&W, blank if unknown
MM (mode)	BIOS hex mode
LLL (lines)	Number of lines on screen
CCC (cols)	Number of physical columns

WWW (width)  
Number of logical columns

*screenmode*  
The SCREENMODE as displayed on the  
HELP screen

See also SCREENMODE above.

SCREENSIZE/S *param*

This causes Anzio to respond to the host with a numeric value containing information on the screen, according to *param* (all units are in pixels):

CLIENT.X sends width of client (interior of window)  
CLIENT.Y sends height of client  
WINDOW.X sends width of window (including borders)  
WINDOW.Y sends height of window (including borders,  
menu, etc.)  
MAX.X sends monitor width  
MAX.Y sends monitor height

SCRIPT

SCRIPT *filename* [E:*lang*][M:*module*][T:*timeout*][*args*]

Runs an external script, such as in VBScript or JavaScript. If the command is simply

SCRIPT

then Anzio will provide an interactive approach to debugging your script. Ordinarily, though, you will run a command with at least

SCRIPT *filename*

This runs the module "Main" in the script file *filename*, in the VBScript language, with no timeout and no arguments. Optional parameters are:

E:*lang* Specifies the language of the script.  
Options are "vbscript" and "javascript".

M: <i>module</i>	Specifies the module (subroutine) within the script to run.
T: <i>timeout</i>	Specifies a timeout value
<i>Args</i>	One or more arguments that are passed to the script module

If the script returns anything (a string), Anzio will process it as an Anzio command.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.



SCROLL [ON] | OFF | TOGGLE

Controls how a host “clear screen” request is interpreted.

SCROLL ON ignores “clear screen”, instead scrolling the current lines up and off the live screen. Use REVIEW to see the scrolled-up data.

SCROLL OFF clears the screen when requested, erasing all data from the previous screen. REVIEW is not available.

SCROLL-LOCK [ON] | OFF | TOGGLE

Enables or disables the  key. The  key suspends screen display of data from the host, useful when data is arriving too rapidly to read. See also HOLD.

SEND


Sends an image of the screen to the host.

SEND *c1 c2 r1 r2*

Sends a line of data from the screen to the host, from column *c1* to column *c2* on line *r1* through *r2*.

Positions can be cursor-relative – see BOX.

SEND *"string"*  
SEND/N *"string"*

Sends the *string* to the host. SEND adds an , SEND/N does not.

The quote marks are necessary to distinguish this command from other forms of SEND.

SET *variable=value*

Sets a user-defined string *variable* to the string *value*. Variable names can be any string not containing "=", and are case independent. The *value* can be any string.

Variables set in this way can be dereferenced with "\${*variable*}" in most commands. See ENVIRONMENT.

SETCOLOR *n* | NORMAL

Sets the working color to color index *n*, where *n* is either a decimal number or a hex value represented as 'xxH'. Does not affect the attribute/color table (see COLOR).

Use SETCOLOR NORMAL if you accidentally set a strange color.

SFTP

Initiates an *interactive* Secure File Transfer Protocol (SFTP) session over an existing SSH connection. The dialog box provides an easy point-and-click interface that allows you to move files easily between the PC and the host system. Unlike the FTP command, the SFTP command allows transfer only with the same host you are already connected to.

SFTPDELETE *hostfile*  
SFTPDELETE/S *hostfile*

Causes Anzio to use SFTP commands to delete a file on the server you are currently connected to.

*hostfile* can be a name or a pathname (with either kind of slash).

The "/S" causes a status result to be sent to the host ("00" means success).

```
SFTPGET hostfile [pcfile]  
SFTPGET/S hostfile [pcfile]  
SFTPPUT hostfile [pcfile]  
SFTPPUT/S hostfile [pcfile]
```



These commands do *programmed* Secure File Transfer Protocol (SFTP) transfers over an existing SSH connection. They allow AnzioWin to work as an SFTP client, although commands can be issued from the host (through Anzio's usual command protocol).

The "/S" causes a status result to be sent to the host ("00" means success). Anzio's DOWNLOAD DIRECTORY will affect downloads (FTPGET[/S]).

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
SLEEP hh mm ss
```

Tells Anzio to sleep until the given PC clock time is reached. For example, you could set Anzio to wake up and dial a remote computer at a preset time.

Enter   to interrupt a SLEEP and awaken Anzio.

```
SPEAK  
SPEAK "text"  
SPEAK c1 c2 r1 r2
```

These commands do text-to-speech, using Windows' speech interface (SAPI). SAPI is included in some versions of Windows, and is downloadable from Microsoft's web site. The configuration of SAPI, including the audio devices to be used, the voice characteristics, and the language, are set in the Windows control panel.

The command SPEAK will read the entire screen. If the argument after SPEAK is in quotes, it is considered as literal *text*; that *text* will be spoken. If the arguments following the word SPEAK are not in quotes, they are assumed to be the coordinates of a rectangular area of the screen — that part of the screen will be read.

SPOOLER [*name*]

Tells Anzio what printer to use when printing with Print Level = Spooler. This can be a different printer from what is used with Print Level = High. Note that when Print Wizard is being used, and it detects that a print job contains escape codes, it will switch to Spooler mode, which will then use the printer identified here.

Use the Windows printer name for *name*. To tell Anzio that the SAME printer should be used for Spooler as for High, omit *name*.

STATUS [LINE] [ON] | OFF

Tells Anzio whether to put status information on the bottom line of the screen. Status includes CAPS lock, NUM lock, keyboard LOCK, and communication errors.

STAYALIVE *nn*

Sets the stayalive interval to a certain number (*nn*) of seconds. When Anzio is connected via telnet or SSH, and STAYALIVE is non-zero, Anzio will send a "do-nothing" packet every *nn* seconds. This can keep a router, firewall, etc. from disconnecting the session during periods of inactivity.

STOP

Stops the Anzio program and returns to the operating system. Same as END.

STOP [BITS] 1 | 1.5 | 2

Sets the number of stop bits in the communication protocol. Usually 1.

SWITCHTO /NEXT  
SWITCHTO /PREV  
SWITCHTO *title*

These commands tell Windows to switch "focus" to a different Windows program that is currently running, or another instance of Anzio. SWITCHTO /NEXT will switch to the "next" instance of Anzio (in order by process number). SWITCHTO /PREV will switch to the previous instance.

Any argument not starting with a slash is assumed to be the title of another Windows program, as shown in its title bar. If the title contains spaces, use quotes around the *title* argument.

SZ

An alias for ZSEND. See ZSEND.

TAB *i j k ...*

Sets tab stops at columns *i, j, k...* Tab column numbers are entered in ascending order, separated by any non-numeric character.

Tab stops can also be set with control codes from the host computer.

When the host system sends a TAB or BACKTAB control character, Anzio moves the cursor forward (backward) to the next (previous) tab stop.

See also TAB CHARACTER and TAB ON.

TAB RESET

Resets all tab stops to every 8 columns, starting at column 1.

TAB CHARACTER *x*

Sets the character sent for the Tab key.



**(I-systems)** With TAB ON, when you press the Tab key, the TAB CHARACTER (usually space) is sent enough times to move the cursor over to the next tab column.

TAB [ON] | OFF | RESET

TAB ON uses the TAB CHARACTER for the Tab key's code.

TAB OFF uses the standard ASCII TAB character for the Tab key.

TAB RESET resets all tab stops to standard settings; that is, every 8 columns beginning with column 1.

TERM *termtype*

Sets the terminal type to be emulated:

N7900	NCR 7900 Model 1
N7901	NCR 7901
VIEWPOINT	ADDS Viewpoint (same as N7901)
VT100	DEC VT100
VT220	DEC VT220
VT320	DEC VT320
VT420	DEC VT420
WYSE50	Wyse 50
WYSE60	Wyse 60
T160E	Innovative Software's multi-language extension to VT220
C332	Versyss C332
SCOANSI	Console of SCO UNIX
TV965	Televideo 965
AT386	Console of various AT&T and UnixWare UNIX systems
ANZIO	Anzio's own definition, based on VT220

IBM3161    IBM 3161  
ADDS200    ADDS Regent 200

➤ Be sure that Anzio's TERM setting agrees with the host computer's expected terminal type.

---

TERMNAME *name*

Sets a TERM variable that will be sent to the host as part of the telnet, SSH, or rlogin protocol, during the initial connection. If no TERMNAME is specified, Anzio will send a name based on the TERM type.

With some host systems, it is possible to specify several choices for TERMNAME, separated by semicolons, with the most desirable setting first. For instance

```
TERMNAME vt320;vt220;vt100
```

During the option negotiation that occurs at the beginning of a telnet session, Anzio and the host system will choose which option works best for both.

TIME/S [*format*]

Sends the PC clock time to the host, by default in the format HH:MM:SS<sup>1</sup>. If *format* is given, it is used as a character pattern to indicate the format which should be used. Possible characters are:

h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
m	Displays the minute without a leading zero (0-59).
mm	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).

z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
am/pm	Insert an "am" or "pm".

TIMEOUT *n*

Sets the file transfer timeout, where *n* is in tenths of a second.

If Anzio sits locked for the specified time, it will a) beep, b) unlock itself, and c) RETRANSMIT. This can get a file transfer going again after some types of communication errors

TITLE *string*

Sets Anzio's window title.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

TRANSFER-OPT

Brings up the Protocol Options dialog box for setting file transfer options.

TRACK-WINDOW [ON] | OFF | TOGGLE

Tells Anzio whether or not to restore its last window position the next time Anzio starts.

TRACK-WINDOW ON works only if you SAVE the current parameters during a session. Anzio will prompt you to save the parameters upon exit.

If you don't need this feature, TRACK-WINDOW OFF will not prompt to save parameters.

TRANSMIT *filename* [*param*] [TRAILER *string*]  
TRANSMIT [ON] | OFF

Controls file transmission according to the host protocol. See *File Transfer*, page 83.






TRANSMIT *filename* opens the file, transmits it to the host, and closes it. If *param* is not given, the file is transmitted one line at a time, with each line followed by a Return. Possible values of *param* are:

SINGLE	Sends one line, then suspends until an XN (transmit next) command happens.
CRC	Special mode for Rasmussen Software's UFT protocol.
LONG	Breaks the line into pieces with a length indicator.

TRANSMIT ON sends the last OPENI input file to the host, one line at a time.

TRANSMIT OFF ends transmission. TRANSMIT OFF is automatically called after sending the last line of the input file.

TRANSMIT TRAILER specifies an end-of-file *string* to be sent to the host after the last transmitted file record. For example, the host software may expect "END\$" or "\".

To specify a  character as the trailer, enter *string* as either    or  127.

TYPE *filename*

Displays a file on the screen. Non-printing characters are shown as hex codes in reverse video.

To verify a received file after transmission, use CLOSEO first, then TYPE the file.

You can PRINT the displayed information.

UNICODE/S [*num*]

Provides a way to enter a character by its hexadecimal Unicode value. If *num* is given, it is assumed to be in hex, representing a character. If *num* is NOT given, Anzio will prompt the user for the value.

The character value is converted according to the current character set, and sent to the host.

UPPERCASE [ON] | OFF | TOGGLE

Like the CAPS LOCK key, if UPPERCASE is on, alphabetic keys are changed to upper case before being sent to the host.

UPPERCASE TOGGLE changes between UPPERCASE ON and UPPERCASE OFF.

USERNAME *name*

USERNAME/S

Sets or sends Anzio's internal username. *name* is used by the Login Wizard.

A username can also be specified in Anzio's command line (see *Names – Settings File Name or Telnet Host Name*, page 183).

USERNAME/S sends the current username to the host system.

VERSION

VERSION/S

Displays (VERSION) or sends (VERSION/S) Anzio's version number.

WAIT *nnn*

This command simply tells Anzio to wait a certain amount of time before sending anything else to the host. The *nnn* parameter is measured in tenths of a second.

WAITFOR *string* [*timeout*]

WF *string* [*timeout*]

Causes Anzio to wait for the given *string* of characters to appear on the screen, or for *timeout* seconds to elapse. Enclose *string* in quotes if it contains a space. The default *timeout* is 86400, or 24 hours.

WF is an abbreviation for WAITFOR.

To interrupt a WAITFOR, press **ALT** **A**.

If a WAITFOR times out, and an ONERROR macro is in effect, that macro will be INVOKEd.

WAITQUIET *seconds*

This causes a macro to suspend operation until the host is "quiet" (not sending any data) for a certain number of *seconds*. This might be used, for instance, to have a macro CAPTURE data coming from the host. It could begin capturing, then do a WAITQUIET 3. When Anzio detected no host output for 3 seconds, the macro could continue and turn off CAPTURE.

WINDOW *c1 c2 r1 r2* [FILL] [BOX | DOUBLE]

Opens a character-mode "window" on the screen, from column *c1*, row *r1* to column *c2*, row *r2*. The original screen contents are saved in memory and later restored with the WINDOWCLOSE command.

Positions can be cursor-relative – see BOX.

The options are:

- |        |   |
|--------|---|
| FILL   | Fill the area with spaces.                    |
| BOX    | Draw a single-line box around the new window. |
| DOUBLE | Draw a double-line box.                       |

WINDOW ZOOM | MINIMIZE | RESTORE

Changes the Anzio window's position on the screen, by maximizing (zooming) it, minimizing it (into an icon), or restoring it, respectively.

WINDOW ROTATE [ 0 | 90 | 180 | 270 ]

Rotates the video display, on PCs that support this feature, especially the TabletPC.

WINDOWCLOSE

Following a WINDOW command, closes the window and restores the screen.

WINPRINT *filename* [*printer-name*]

Asks Windows to print *filename* using the program associated with its file extension, in the same way as Windows Explorer. This uses the "printto" shell command.

If *printer-name* is given (possibly surrounded by quotes), the file is printed on that printer; otherwise, the file is printed on the Windows default printer.

*filename* can contain environment variables, coded as

`${var}`

Anzio will replace each environment variable with its contents. These can be DOS/Windows environment variables, or Anzio pseudo environment variables such as ANZ\_IP, as explained under ENVIRONMENT.

WINSTART *filename* [*params*]

Asks Windows to start or open *filename*, which is either a program file or a data file. If *filename* is a data file, it is opened using the program associated with its file extension, in the same way as Windows Explorer.

*filename* can contain environment variables, coded as

`${var}`

If *filename* is a program, it can be started with command line parameters as in *params*.

Anzio will replace each environment variable with its contents. These can be DOS/Windows environment variables, or Anzio pseudo environment variables such as ANZ\_IP, as explained under ENVIRONMENT.

WRITE *text*

Writes *text* to the current output file, opened by OPENO.

*text* may include control characters. To embed a control character in *text*, preface it with `CTRL P`, for example `CTRL P ESC`.

To embed any ASCII character in *text*, enter `CTRL P` followed by hex 10 and the two-character ASCII hex value of the character. The hex 10 represents a second `CTRL P`, used internally for quoting ASCII characters. For example, to embed a Return character (hex 0D), enter:

`CTRL P 100D`

WRITELINE [*text*]

WRITELN [*text*]

If *text* is specified, it is written to the current output file. Then a return and linefeed are written to the current output file.

Arguments can also contain `$(variable)` where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

XN


Transmits the next line of a file to the host. Used with TRANSMIT SINGLE.




XRECEIVE [*options*] [*filespec*]]  
XRECEIVE/S [*options*] [*filespec*]]

Receives one or more files from the host system, using the Xmodem protocol.

A file transfer is specified as:

XRECEIVE *filename* 

or

XRECEIVE *wildcard* 

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For possible options, see KRECEIVE.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes, see XRECEIVE.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

XSEND [*options*] *filespec*  
XSEND/S [*options*] *filespec*

Sends one or more files to a host system using the Xmodem protocol. The *filespec* can be either a single filename or a wildcard.

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For a description of available options, see KRECEIVE, above.


If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes see KRECEIVE, above.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.


```
YRECEIVE [options] [filespec]]
YRECEIVE/S [options] [filespec]]
```

Receives one or more files from the host system, using the Ymodem protocol.

A file transfer is specified as:

```
YRECEIVE filename 
```

or

```
YRECEIVE wildcard 
```

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For possible options, see KRECEIVE.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes, see XRECEIVE.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
YSEND [options] filespec
YSEND/S [options] filespec
```

Sends one or more files to a host system, using the Ymodem protocol. The *filespec* can be either a single filename or a wildcard.

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For a description of available options, see KRECEIVE, above.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes see KRECEIVE, above.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
ZRECEIVE AUTO-OFF
ZRECEIVE AUTO-ON
ZRECEIVE [options] [filespec]]
ZRECEIVE/S [options] [filespec]]
```

Receives one or more files from a host system using the Zmodem protocol, or controls Zmodem auto-reception.

---

➤ The command “RZ” can be used as an alias for “ZRECEIVE”.

---


When Zmodem auto-reception is on, a program running on the host system can send (download) a file to the PC without the user doing anything. This capability can be turned on or off with

```
ZRECEIVE AUTO-ON
```


and

```
ZRECEIVE AUTO-OFF
```

A file transfer is specified as:

```
ZRECEIVE filename 
```

or

```
ZRECEIVE wildcard 
```

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For a description of options, see KRECEIVE.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes, see KRECEIVE.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

```
ZSEND [options] filespec  
ZSEND/S [options] filespec
```

Sends one or more files to a host system using the Zmodem protocol. The *filespec* can be either a single filename or a wildcard.

---

➤ The command “SZ” can be used as an alias for “ZSEND”.

---

If *options* are specified, they override the global options set in Transfer/Transfer Options for this transfer session only. Options start with “-” or “/”. For a description of available options, see KRECEIVE, above.

If “/S” is included in the command, a result code is sent to the host system (or macro). The result code is always 2 characters followed by a Return. For possible result codes see KRECEIVE, above.

Arguments can also contain  $\${variable}$  where *variable* is a DOS/Windows environment variable or an Anzio internal environment variable, which Anzio will translate. See ENVIRONMENT.

## 10 More On Starting Anzio

---

This section describes various ways of controlling Anzio's behavior when it starts up.

The following sections describe:

- *Command Line Parameters & Default Files*

### 10.1 Command Line Parameters & Default Files

---

You can call the Anzio program with one or more command line parameters, separated by spaces:

```
Anzio32 param1 /param2 /param3 name ...
```

Command line parameters can run a defined key, specify a settings file, connect to a telnet host, set communication settings, or restrict Anzio's memory usage. Strings used in parameters can be enclosed in "double quotes" if they contain spaces or tabs.

Command line parameters are coded into the program icon.

Command line parameters can also be added by another program starting up Anzio. For example, if a web browser is running, and the user clicks on a cross-reference that contains a "telnet://" link, Anzio will be called with that link name.

If there are no command line parameters, Anzio will try to load its standard settings file (AnzioWin.def). This file contains all user settings from the previous Anzio session. The settings file can also specify a key definition file to be loaded. Anzio looks for a settings file in up to three places until found:

1. The starting directory.
2. The parent directory of the starting directory.
3. Anzio's home directory.

If there are one or more command line parameters, Anzio looks at each parameter's length and first character:

- NONE means “start Anzio without reading any settings file”, that is, as if Anzio was just installed and being run for the first time. See the following section.
- A single character *X* means “call defined key *X*” – see *Single Character – Start Defined Key*, next page.
- Multiple characters that do not start with a slash “/” indicate a name. The name can be a settings file name to be loaded, or a telnet URL with a username and host name. See *Names – Settings File Name or Telnet Host Name*, page 183.
- A parameter starting with a slash and a C, D, H, K, M, or T sets an Anzio option – these are described in alphabetical order, beginning on page 184.

### 10.1.1 **‘NEW’ for New Connection**


Use the command line parameter NEW to start Anzio without loading the settings file. Anzio will start up just like after initial installation.

### 10.1.2 **‘NONE’ For Clean Startup**

Use the command line parameter NONE to start Anzio without loading the settings file **or any key file**.

### 10.1.3 **Single Character – Start Defined Key**

A command line parameter consisting of only one character will start that defined key (macro) upon startup. The key must be a printable character (not a function key), for example:

```
Anzio S 
```

will start Anzio and load the standard settings file. In the settings file is an entry for a defined key file, which will then be loaded. At this point, the “S” defined key is called.

Note that defined key names are case-sensitive – “s” and “S” are two different keys.

You can also define a key with a command line parameter – see */D Define Key*, page 184.

#### 10.1.4 **Names – Settings File Name or Telnet Host Name**

Any parameter more than one character long that does not start with a slash ("/") is taken to be a settings file name, or a URL. For example:

```
MyAnzio.def
telnet://my.host.com
user123:password777@my.host.com:80
```

A complete name parameter can contain a network address, including a username and password, and a host name and optional port:

```
[urctype][username[:password]@]name[:port]
```

That is, all parts are optional except *name*.

The *urctype* specifies what kind of connection will be made. Possible values are

```
telnet:
telnet://
ssh://
rlogin://
modem://
```

Anzio reads a name parameter as follows:

1. Anzio strips off an initial *urctype* if it exists, and uses it to determine the connection type.
2. Anzio strips a slash, if any.
3. If the parameter contains an at-sign @, Anzio takes the characters before the @ as a *username* and possibly a *:password*. The username and password (if any) are stored for later use by the Anzio Login Wizard.
4. Anzio first tries to use *name* as a settings file name.

If the file exists, in the current directory, in the parent directory, or in Anzio's home directory, that file is used instead of AnzioWin.def.

5. If there is no file by that *name*, the name is considered to be a host name or IP address, with an optional *:port*. Anzio loads the standard settings file (AnzioWin.def), but then connects to the given host name rather than the host named in the settings file.

---

➤ A special case is a URL representing a modem. Its format is  
modem://[*username*[:*password*]@]*phonenumber*  
Anzio will be configured to do a modem dial-up connection to the indicated *phonenumber*.

---

For example, if you need to connect to three host systems, you can create three Windows icons, each naming a different settings file, which will contain all the settings appropriate to that host system.

For Internet telnet connections, you can set up AnzioWin.def to contain the most common set of parameters. Then when you activate a telnet cross reference in your browser, the browser will start up Anzio, passing the host name as a parameter. Anzio will start up, read AnzioWin.def, and connect to the indicated host name.

---

➤ Note that if you have a telnet host requiring a special setting, you can create a settings file having the same name as the host. Anzio will find and read that settings file, and so will not read the parameter as a host name. Inside the settings file, specify the desired host name, and Anzio will connect to that host.

---

#### 10.1.5 **/C Choose Settings File**

The parameter

*/C*

causes Anzio to prompt the user for the name of a settings file to use.

#### 10.1.6 **/D Define Key**

Use */D* to specify a key definition (macro):

*/Dk string*



where *k* is the key to be defined (any printable key), and *string* is the definition. The space separating *k* and *string* is optional.

See *Defined Keys and Macros*, page 46, for more information.

You can use this command line option with the “call key” single-character option. For example, you can define the D key (and save it in the default key file) as:

```
F8 DIAL F8 CALL N | |
```

This definition dials the number returned by another defined key N (“CALL N”). You can then specify the number as a command line option by defining key N:

```
anzio D /DN555-1212
```

This command first defines N (“555-1212”), then starts Anzio and runs key D.

### 10.1.7 /H Hostname

Use /H to specify a network host system to connect to, overriding the host name in the settings file:

```
/H:hostname
```

You can specify a host port number as well:

```
/H:hostname:port
```

A numeric parameter following the /H option is interpreted as a port number, for use with a Netscape browser:

```
/H:hostname port
```

### 10.1.8 /K Kiosk Mode

Use /K to put Anzio into *kiosk mode*. In kiosk mode, Anzio limits the options available to the user, such as printing, dialing, or exiting. Kiosk mode is designed for libraries, and other places where public users are expected.

The /K option disallows all shortcut keys and the Anzio help information.

Use `/K:nnn` to allow one or more of the following menu items. `nnn` is the sum of the options:

- 0 = none
- 1 = Print Screen
- 2 = Eject
- 4 = Quit
- 8 = Capture to printer
- 16 = Capture to file
- 32 = Copy (to clipboard)
- 64 = Paste
- 128 = Dial
- 256 = Hangup

### 10.1.9 **/T Communication Type**

Anzio uses `/T` to specify which type of communication to use:

- `/Ts` serial
- `/Tt` TCP/IP
- `/Tr` rlogin
- `/Tp` PicLan
- `/Tm` modem
- `/Th` SSH

### 10.1.10 **/V Verbose**

This switch causes Anzio to present various diagnostic information about its establishing its connection to the host. For instance, for a modem connection Anzio will give various status messages, and also tell you at what baud rates the two modems connected. For a telnet connection, Anzio will show the results of telnet option negotiation.

## **Part IV Technical Reference Guide**

## 11 File Transfer Protocols

---

This section describes the byte-level behavior of Anzio during file transfer: TRANSMIT, RECEIVE, CAPTURE, and passthrough printing.

➤ This section is intended for programmers. See *File Transfer*, page 83, for the operator view of file transfer.

Several source programs that demonstrate the host side of file transfer are included with Anzio – see Appendix E, *Distribution Information*, page 221.

The following sections describe:

- *Simple Upload – TRANSMIT*
- *TRANSMIT LONG*
- *TRANSMIT CRC*
- *RECEIVE*
- *RECEIVE WITH CRC*
- *Simple Receive – CAPTURE*
- *CAPTURE LONG*
- *Passthrough Printing*

### 11.1 Simple Upload – TRANSMIT

---

The TRANSMIT command (Transfer/Transmit) reads each line/record from the input file, then sends it to the host. Each input line is terminated by a carriage return (ASCII 13).

Your host operating system may have an 80-byte maximum for input lines (ACCEPT). In this case, you can use TRANSMIT LONG (see next section) to break long records into 80-byte chunks.

If LOCK ON is in effect, Anzio uses *handshaking* for each line transmitted. That is, Anzio waits for an unlock, then sends a line and

locks itself. This is repeated until the end of the file is reached. The local file is then closed.

## 11.2 TRANSMIT LONG

---

The TRANSMIT LONG command sends records of any length to the host, in small pieces. The host must read each piece and reassemble the record on its end – see the “RECV-PC” example programs in Appendix E, *Distribution Information*, page 221.

Anzio reads each line from the PC file and breaks it into one or more pieces of up to 77 bytes. Each piece is sent to the host as one line, with three bytes of record info:

```
nnYddddddddd . . .
```

where *nn* is two ASCII characters (01-77) giving the number of data bytes *dddd*. The third character indicates whether this is the last piece of the record, an ASCII “Y” or “N”.

## 11.3 TRANSMIT CRC

---

TRANSMIT CRC uses a special protocol for error checking, and is intended only for use with the Universal File Transfer (UFT) utility program, formerly available from Rasmussen Software, Inc.

## 11.4 RECEIVE

---

Anzio uses incoming codes from the host to RECEIVE a file. Each record sent by the host is bracketed with a start code DC2 and an end code DC4. These ASCII codes are the same codes used to drive a printer “slaved” to a standard terminal.

In contrast, CAPTURE simply puts all incoming characters into the current output file – see *Data Capture*, page 80.

After Anzio sees a DC2, it puts all succeeding characters into the *receive buffer*, until a DC4. The receive buffer is expanded as needed, up to the available dynamic memory.

- The receive buffer is used by both RECEIVE and CAPTURE, only one of which may be active.
- 

Incoming data is added to the receive buffer until the host sends an “unlock” to Anzio. Then, if an output file is open:

- Anzio writes the receive buffer to the PC output file, then clears the buffer.
- Anzio sends a carriage-return CR to the host, indicating that the host can send the next record.

- The host must send the appropriate end-of-record marker, if any, inside the DC2/DC4 codes. For example, to end up with text lines in a PC file, the host must send CR and LF codes at the end of each record.
- 

### 11.5 RECEIVE WITH CRC

---

RECEIVE CRC receives files using a special protocol for error checking, and is intended only for use with Universal File Transfer (UFT) utility program, formerly available from Rasmussen Software, Inc.

### 11.6 Simple Receive – CAPTURE

---

CAPTURE stores each incoming line as the line is displayed on the screen. When the cursor moves to the next line, Anzio writes the stored line to the current output file.

### 11.7 CAPTURE LONG

---

CAPTURE LONG captures long records from the host, using the piece-wise protocol described with TRANSMIT LONG, above.

### 11.8 Passthrough Printing


---

Some terminal types provide *passthrough printing*, where data from the host passes through the terminal to the printer, without being displayed. The host sends a special escape code to start pass-through print mode,

then some printer data, and then a termination code. Anzio recognizes these pass-through codes for each supported terminal type.

Anzio supports pass-through printing to either the printer or a file. If there is an open output file (from OPENO), data is sent there. Otherwise, data is sent to the printer set by PRINTER.


---

➤ If the host does not send a termination code, press  to exit pass-through print mode and return to normal display mode.

---

## 12 Sending Anzio Commands From The Host

---

The host can send Anzio commands by using special control codes. Any  command or function can be called from the host.

For example, Anzio includes file transfer programs for the host which use host control to automate file transfer operations. See Appendix E, *Distribution Information*, page 221.

The host sends commands to Anzio enclosed in command-start and command-end codes. When Anzio receives a command start code (hex 1C, octal 34, decimal 28), all characters until the stop code (hex 1D, octal 35, decimal 29) are processed as Anzio commands. For example, if the host sends:

```
hex-1C OPENO DOWNFILE hex-1D
```



- 
- Anzio can be told whether received commands should be displayed or not. This is governed by the RECEIVE QUIET setting, in the Transfer menu. If RECEIVE QUIET is OFF, commands from the host will be shown on the screen.
- 

Anzio executes the command:

```
OPENO DOWNFILE 
```

The local file DOWNFILE is opened for output. However, if that file already exists, an error message is displayed on the screen for the operator.

In other words, even though a function is called by the host, all displays, error messages, and input will be to and from the PC screen and keyboard.

- 
- You can even send Anzio the  command itself, for example, to define a key from the host.  is represented by NULL @ (hex 00 followed by @). Contact Rasmussen Software for codes for other special keys.
-



➤ The host has no way to know if the Anzio command OPENO succeeded. Some commands provide an option "/S" for "send result to host".

- For example, OPENO/S sends a two-digit result code back to the host, indicating whether it was successful, or the file exists, or another output file is already open. *Commands*, page 102, describes all commands having the /S option.

**(UNIX)** You can send Anzio commands with the UNIX shell command "echo":

```
echo "\0034OPENO DOWNFILE\0035" 
```

Where "\0034" indicates octal 34 (hex 1C), and "\0035" is octal 23 (hex 1D). Depending on your UNIX system, you may need to add a parameter "-e" to your "echo" command in order to have it translate embedded octal. Or, you may be able to use the "printf" command.

➤ Early software from Rasmussen Software used hex-11 and hex-13, respectively, as command start and command end. These can be re-enabled in Edit/Advanced Options.

## 13 Security Options

---

Anzio offers two methods of protecting your communication data (including your password) while it moves over a network such as the Internet: SRP and SSH. Both offer *authentication* and *encryption*.

Authentication is the process of the two players proving who they are to the other. This means that you, the user, can be sure that the host system Anzio is connecting to is actually the system you think it is. It also means you the user have to verify to the host system who you are, the equivalent of supplying your login and password. But in these secure systems, your password never goes across the network in cleartext (unencoded).

Encryption means to encode the data. Although encryption is used during the authentication process, the term is used here as separate from that. That is, encryption refers to encode all data (going both ways) *after* authentication has occurred.

➤ SRP and SSH are supported only in AnzioWin, not in Anzio Lite. Also, your copy of AnzioWin may not support these protocols, because of US export restrictions.

The following sections describe:

- *SRP*
- *SSH*

### 13.1 SRP

---

SRP was developed at Stanford University. It is an encryption/authentication *option* under the telnet protocol.

SRP support on the host (server) involves replacing the standard telnetd, password, login, and possibly some other programs on the server.

When SRP is enabled on the host, and Anzio is connecting using the telnet protocol, and Anzio is configured to **allow** or **require** SRP authentication (in Edit/Advanced Options/Security), then Anzio and the

server will negotiate and establish an SRP connection. Then if encryption is allowed or required, the main communication of the telnet session will be encrypted.

---

➤ Anzio does not have support for FTP using SRP.

---

### **13.2 SSH**

---

The SSH protocol is a separate protocol from telnet, using a different socket (normally 22). On the host (server) end, it requires an SSH daemon (“sshd”) to be running. SSH daemons are available from several sources.

In Anzio, you must choose SSH as your means of connection in the Communicate/Setup dialog box. Further options for SSH are available in the Setup button next to “SSH”.

In order to prove who it is, the server will send out a “fingerprint”. If Anzio on this PC has not connected to this particular server before, Anzio will display this fingerprint and ask you to verify it. Ideally, for the most security, you will have been given this fingerprint information already by some secure means, and you can confirm to Anzio that the fingerprint is correct, meaning you’re connected to the right host. If you have NOT been given this information, and you confirm to Anzio, you will in effect be trusting this first connection as being to the correct host.

After you confirm, Anzio stores the fingerprint in the registry, and future connections are confirmed automatically.

After confirmation of the host, you will be prompted for your login and password. These will be encrypted during transmission. Assuming they are correct, the main session will begin, and it will look just like a normal telnet session.

## 14 Advanced Options

---

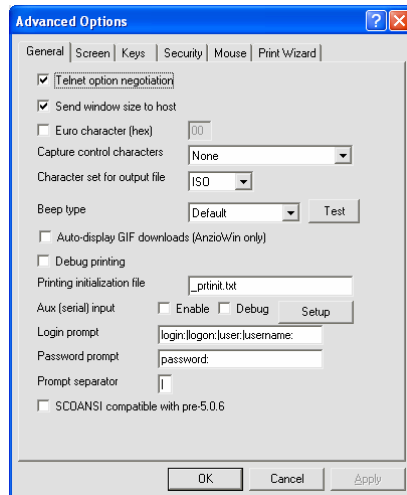
This section describes some more advanced settings within Anzio, accessible under Edit/Advanced Options. They are arranged here by tab.

The following sections describe:

- *General*
- *Screen*
- *Keys*
- *Security*
- *Mouse*
- *Print Wizard*

### 14.1 General

---



#### Telnet option negotiation

When connecting using the telnet protocol, should Anzio negotiate various options with the host? Normally ON.

Send window size to host

Should Anzio send its window size (rows and columns) to the host, when connected using various protocols such as telnet. Normally ON. Try turning this off if your host system reacts strangely when Anzio switches between 132-column and 80-column modes.

Euro character (hex)

Allows you to specify a custom hex value that your host system uses for the Euro symbol (€). This setting takes precedence over Character Set settings. Normally disabled. To enable, check the left box, and enter a hex value in the right box.

Capture control characters

When Anzio is capturing incoming characters, this setting specifies how it should treat control characters, including escape sequences. Options are:

- None Only plain text is included in the captured data.
- All All control characters are captured
- All except Escape sequences  
Escape sequences (such as for screen positioning) are NOT captured.

Character set for output file

When Anzio captures to a file, and the incoming data contains non-ASCII characters, this setting specifies by what character set they are stored in the output file.

Options are:

- ISO The Windows character set.
- OEM The OEM (DOS) character set.
- UTF-8 Unicode UTF-8 form.

Beep type

When a host program sends a code (hex 07) to Anzio to “beep”, how should Anzio make a sound. Your choice depends on whether your PC has a sound card, and whether it has speakers. Options are:

PC Speaker

Uses the PC motherboard's speaker. This is the only option that will be audible if the PC does not have BOTH a sound card and speakers. Will produce a brief "chirp", that may still be hard to hear.

Other settings

Other options are names of Windows sounds. In your Windows configuration (via the Control Panel) you can associate a sound file with each of these names.

Auto-display GIF downloads (AnzioWin only)

If enabled, then whenever a GIF file is transferred into Anzio from the host (such as with Zmodem), Anzio will display it on the screen.

Debug printing

If turned on, Anzio will pop up an informational window whenever it prints. This can be useful in diagnosing problems in printing, especially when using AnzioWin's Print Wizard feature.

Printing initialization file

Tells Anzio the name of a file to be used to initialize the printer, if that file exists. See *Print Initialization Files*, page 75.

Aux (serial) input

Allows you to configure and enable input from an "aux" serial device, such as a barcode reader. Items are:

Enable Turns it on

Debug Causes a debug box to pop up with information on serial data flow

Setup Allows you to configure the serial connection

Login prompt

Tells Anzio's Login Wizard what prompt(s) the host system might display asking for a login name. Multiple entries are separated by the *prompt separator*, below. Default is "login:|logon:user:|username:". Case INsensitive.

**Password prompt**

Tells Anzio's Login Wizard what prompt(s) the host system might display asking for a password. Multiple entries are separated by the *prompt separator*, below. Default is "password:". Case INsensitive.

**Prompt separator**

What character separates multiple entries in the *Login prompt* and *Password prompt* fields above.

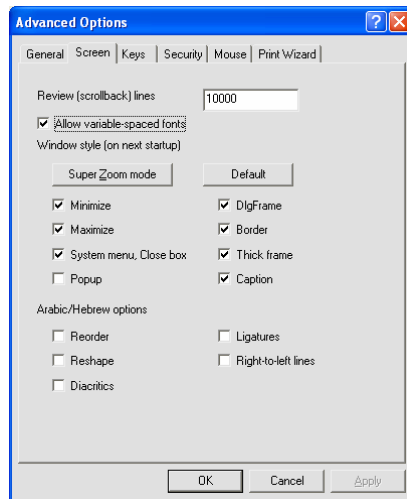
**SCOANSI compatible with pre-5.0.6**

Ordinarily, Anzio's SCOANSI emulation is compatible with changes introduced in SCO Open Server 5.0.6. Setting this option causes it to be compatible with earlier versions of that console.

---

**14.2 Screen**

---



**Review (scrollback) lines**

Sets the number of lines (at 80 columns) of data that Anzio will store in its scrollback (review) buffer.

**Allow variable-spaced fonts**

Indicates whether Anzio should present variable-spaced, along with mono-spaced, fonts in View/Screen Font.

Generally you'll want to use a mono-spaced font, such as Courier New. However, some Far East fonts are marked as variable-spaced.

Window style (on next startup)

By changing these settings, you can change the appearance of Anzio's window the next time Anzio is started. Some combinations may not be legal to Windows. Contact us for help if needed. Options are:

Minimize

Should the minimize button be included?

Maximize

Should the maximize (zoom) button be included?

System menu

Should the "system menu" (attached to the program icon) be included? Also governs the 'close' button.

Popup Sets the window to a "popup" style.

DlgFrame

A dialog-style frame has NO menu.

Border Should the window have a border?

Thick frame

Should the window have a thick frame?

Caption

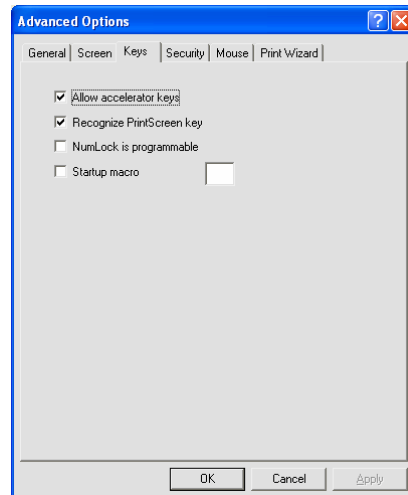
Should the window have a title bar?

Arabic/Hebrew options

On a Windows system that supports Hebrew and/or Arabic, these items govern whether Anzio should tell Windows to do such things as reordering the text into right-to-left mode. If the data coming from the host has already been reordered, so it is in visual order already, then you would want to DISABLE the Reorder setting, for instance.




## 14.3 Keys




### Allow accelerator keys

Accelerator keys are those ALT-key combinations that trigger Windows menu events. Normally they are allowed. If you disable this setting, you will be able to use those keystrokes in other ways, such as to assign macros to them. Once assigned, you can turn this back on, because macro assignments have priority over accelerators.

### Recognize PrintScreen key

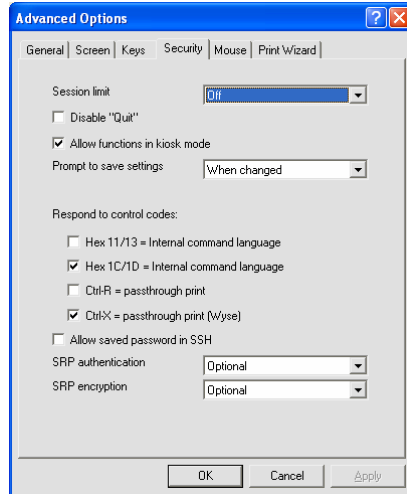
It's nice to be able to print the screen by pushing the  key. But you can disable this feature if necessary.

### NumLock is programmable

If enabled, you can assign a macro to the NumLock () key, such as to emulate the PF1 key on a VT keyboard. We don't recommend this, however, because it interferes with normal operation of the NumLock.

**Startup macro** You can specify a particular macro key to always run when you start Anzio. Check the left box, and type a key into the right box (normally an alphabetic or numeric key).

## 14.4 Security



### Single instance

Governs how Anzio will behave if a user starts a second “instance” or session. Options are:

**Off** This feature is disabled; user can run multiple sessions.

#### Keep existing

The new session will be terminated, and the user will be switched back to the already-running session.

#### Keep new

An already-running session of Anzio will be terminated. User will stay in the new session.

**Disable “Quit”** Disables quitting from Anzio at the PC end. This forces the user to terminate (log out of) the host session, which will then cause Anzio to terminate.

### Allow functions in kiosk mode

If Anzio is run in kiosk mode, should macros containing Anzio functions be allowed?

### Prompt to save parameters

When Anzio is terminated, it will generally check to see

whether any settings have changed, and if so, ask the user whether they should be saved. This setting can change that behavior. Options are:

When changed

As just described.

Save without prompting

Changed settings will be saved without asking the user first.

Don't ask, don't save

The user will not be asked, and the settings will not be saved.

#### Respond to control codes

Certain one-byte codes received from the host can cause certain actions in Anzio. However, if received in error, they can cause problems. This section allows you to control them. Codes are:

Hex-11 = Anzio command

In early versions of Anzio, a command sent from the host was bracketed between hex-11 and hex-13 (which are also XON and XOFF). This setting is normally disabled now.

HEX-1C = Anzio command

The current method for sending Anzio commands from the host is to bracket them with hex-1C and hex-1D. Disable this box for higher security, preventing any Anzio commands to be sent from the host system.

Ctrl-R = passthrough print

Normally disabled. Control-R is code DC2, which is sometimes used for "aux print".

Ctrl-X = passthrough print (Wyse)

A control-X starts passthrough print. This applies only when Anzio is emulating Wyse50 or Wyse60.

#### Allow saved password in SSH

When checked, a stored password can be used with an SSH connection, just as Login Wizard would otherwise run. Note that this is a weakening of SSH security.

### SRP authentication

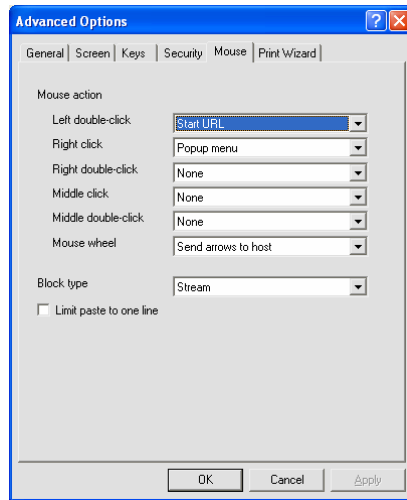
When connecting via telnet, the host system may have the SRP protocol. This setting tells whether SRP authentication (verifying your username and password) is to be done. Options are Off, Optional, and Required.

### SRP encryption

If an SRP connection is established with the host, and authenticated, should the remaining traffic (that is, the main communication of the session) be encrypted? Options are Off, Optional, and Required.

## 14.5 Mouse

---



### Mouse action (clicks)

Configures the actions Anzio will take on various mouse clicks. Options are:

None No action

Send word with CR

Sends the "word" pointed to on the screen, with a RETURN.

Send word

Sends the "word" pointed to on the screen.

Execute macro

Execute a user-defined macro. The particular macro to be run can be set only by editing the settings file; see *Editing the DEF File*, page 209.

Adjust Palette

Adjusts the brightness and contrast of the bitmap shown on the screen, according to the position on the screen where the mouse is clicked. The horizontal axis is contrast, and the vertical is brightness.

Send bracketed word with CR

Picks up a word from the screen, between brackets [ ], strips any leading spaces, and sends it to the host, followed by RETURN

Start URL

Picks a string from the screen, and does the equivalent of an Anzio command WINSTART (or a Windows command START) of that string. Usually used to launch a web browser to the URL pointed to on the screen.

Popup menu

Pops up a context menu.

Send word (allow underscore), CR

Sends the word pointed to on the screen, followed by RETURN. An underscore qualifies as part of a word.

Send word (allow underscore)

Sends the word pointed to on the screen. An underscore qualifies as part of a word.

Mouse action (mouse wheel)

Tells Anzio what to do when the user scrolls the mouse wheel. Options are:

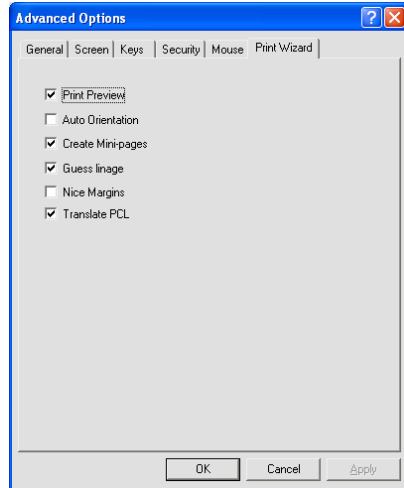
None Does nothing.

Scrollback

Scrolling up will take Anzio into scrollback (review) mode, where you can see information that has scrolled off the top. In scrollback mode, the mouse wheel can move up and down (regardless of this setting).

	Send arrows to host Sends up-arrow or down-arrow codes to the host.
Block type	Sets what shape block (or highlight) will be selected if you drag the left mouse button. Options are: Column A rectangular block will be selected. In this mode, when you copy a selection to the Windows clipboard, a bitmap will be copied as well as text. Stream Similar to selection in a word-processing program, but each line (or part) is sent with a RETURN. Wrap Similar to selection in a word processing program. Lines are assumed to wrap from one to another with one intervening space, so the contents of the highlight are copied as one line.
Limit paste to one line	If enabled, then when Anzio is pasting information from the clipboard to the host, it will be limited to pasting one line.

## 14.6 Print Wizard



### Print Preview

If checked, Print Wizard will present a Print Preview window before printing a job. This lets you see how the data will lay out on the page, and also adjust various margins, etc.

### Auto Orientation

If checked, Print Wizard will assume that print jobs should print in portrait orientation, but if they are wide format (more than 99 characters per line), it will switch to landscape.

### Create Mini-pages

If checked, Print Wizard will try to determine when a print job is intended for a short page, such as for a receipt or a single label printed on a continuous-feed printer.

### Guess linage

If a print job does not contain formfeed characters, Print Wizard must determine how many lines per page to assume. If this item is checked, it will attempt to guess the correct number of lines per page by using pattern analysis; otherwise, it will assume 66 lines per page.

Nice Margins

If this item is checked, Print Wizard will add up to 1/2" margins on the page, if they will fit given the amount of text to be printed. If **not** checked, Print Wizard will print the data as high and as far to the left as possible for the given printer.

Translate PCL

If checked, Print Wizard assumes that any print job containing escape characters is in PCL, and it will translate that PCL in order to render the output on any Windows-supported printer. If **not** checked, print jobs with escape codes are printed at Spooler level.



## 15 Customizing Anzio

---

Anzio can be configured by the user for different hosts and communication environments. You can also customize the Anzio program itself:

- See *Command Line Parameters & Default Files*, page 181, to start Anzio with a particular host, or to set *kiosk* (public use) mode, or to start or define a key macro, etc.
- To change the menus and menu items, use a Windows *resource editor* on the Anzio executable file (anzio32.exe or anzio32r.exe). You can also change or remove accelerator keys, such as **ALT X** to exit.
- You can edit the settings (.DEF) file with NOTEPAD, for instance.
- Anzio provides access to some internal settings with the HEXPATCH command.

The following sections describe:

- *Editing the DEF File*

### 15.1 Editing the DEF File

---

Anzio's settings file, such as ANZIOWIN.DEF, is stored in text format. It can be edited using a text editor such as Notepad. Normally, there is not a need to do this. However, some very seldom-used settings are accessible only in this way; they are not accessible through Anzio itself. Following are those items:

Cursor-color

Sets the color of Anzio's cursor, as a decimal value.

C332-pages

The number of concurrent sessions that can be opened in C332 emulation mode.

SFTP-timeout

The number of seconds to wait when expecting a response in SFTP.

Connecttime

The number of seconds to keep retrying for a network connection. Setting this higher allows time for a dial-up modem to establish a connection, for instance.

Flush-capture

Setting this to "1" tells Anzio to immediately flush to disk anything captured during a CAPTURE FILE operation.

Flush-spooler

Setting this to "1" tells Anzio to always flush output when printing in Spooler mode.

Flush-raw-print

Setting this to "1" tells Anzio to always flush output when printing in Raw mode.

Font-alt

A string of font names, separated by commas, telling Anzio which alternate fonts to use for the screen.

Printfont-alt

A string of font names, separated by commas, telling Anzio which alternate fonts to use for printing.

Left-click-macro

A number representing the macro to be run on a left mouse click, if so configured in Edit/Advanced Options/Mouse.

Left-dbl-click-macro

A number representing the macro to be run on a left mouse double-click, if so configured in Edit/Advanced Options/Mouse.

Mid-click-macro

A number representing the macro to be run on a middle mouse click, if so configured in Edit/Advanced Options/Mouse.

Mid-dbl-click-macro

A number representing the macro to be run on a middle

mouse double-click, if so configured in Edit/Advanced Options/Mouse.

Right-click-macro

A number representing the macro to be run on a right mouse click, if so configured in Edit/Advanced Options/Mouse.

Right-dbl-click-macro

A number representing the macro to be run on a right mouse double-click, if so configured in Edit/Advanced Options/Mouse.

Xlate-keystrokes

Set to "0" to tell Anzio NOT to translate keystrokes when sending them to the host, such as when using a Far East add-on program. Otherwise leave this line out. See next item.

Utf8-kbd

Set to "2" to tell Anzio that a Far East add-on program is sending it keystrokes in UTF-8. Set to "0", and set "xlate-keystrokes" to "0" if an add-on is sending keystrokes already in the character set of the host system, such as Big-5. Otherwise leave this line out.

## Appendix A Error Messages

---

Anzio displays error messages from two sources: hardware and software. Hardware errors, such as a PARITY error, are displayed on the status line. Software errors, such as "File Not Found", are displayed on the status line or in popup dialog boxes.

This section describes both types of error messages:

- *Communication Chip Errors*
- *Software and File Error Messages*

### A.1 Communication Chip Errors

---

If Anzio detects an error in the PC communications hardware, the error type (PARITY, FRAMING, or OVERRUN) is displayed on the status line.

PARITY	<p>A parity error has occurred. Verify that your PARITY and BAUD settings match the host's.</p> <p>Some parity errors are caused by telephone line noise over a modem. You may be able to install an error-correcting modem in your PC.</p>
FRAMING	<p>The PC is not receiving the correct number of bits per character. If this error is intermittent, it may be the result of phone noise, as described above. Verify that your BAUD, PARITY, DATA BITS, and STOP BITS are set to match the host's.</p> <p>Anzio's DATA BITS is the number of bits for data only. DATA BITS 7 and PARITY OFF is a seven-bit character, but DATA BITS 7 and PARITY ON is an 8-bit character.</p>
OVERRUN	<p>When a character arrives at the PC's serial port, it generates an <i>interrupt</i>. The PC interrupts whatever it is doing, fetches the character, and puts it into Anzio's buffer. An overrun is caused when another character comes in before the first is processed.</p>

This is almost always caused by poor hardware design choices. The chip that does most of the work in a serial port is called a UART. For serial communication to function well under Windows, at any but the lowest baud rates, the UART must be a type 16550 or equivalent. Earlier chips (8250, 16450) do not have enough buffering. If you get OVERRUN messages, ask your hardware supplier to determine what kind of UART you have, or contact us for assistance.

Some PCs that have a serial port as part of an integrated chipset, on the motherboard, claim to have a 16550, but don't behave that way. The best solution is to add another serial port, either inside the computer case or a USB-to-serial adapter.

## **A.2 Software and File Error Messages**

---

This section provides a list of error messages in alphabetical order.

### **AN INPUT FILE IS OPEN**

There is an input file already open. Use CLOSEI to close the current input file, then open the new input file.

### **AN OUTPUT FILE IS OPEN**

There is an output file already open. Use CLOSEO to close the current output file, then open the new output file.

### **BAD I/O STATUS: *nn* ON *filename***

A file operation returned an unexpected status.

### **BAD MODEM STATUS**

The DIAL command received an unexpected code from the modem.

### **BAD TAB FORMAT**

Anzio could not understand the format of your TAB command.

DEMO VERSION TERMINATED

You are running a demo version of Anzio, and your time is up.

DISK ERROR ON *command*

A disk error occurred while processing *command*.

DISK OR DIRECTORY FULL

There is no more room on the current output disk or directory.

FILE EXISTS

An OPENO command tried to open an existing file – either delete the file or use a different filename.

FILE OR SUBDIRECTORY ERROR

A file or subdirectory name does not exist or is in the wrong format.

FILE NOT FOUND: *filename*

The specified file was not found – verify that *filename* is the correct name.

INSUFFICIENT MEMORY

The dynamic memory space available to Anzio is full. Anzio uses this memory for both defined keys (macros) and received data.

INVALID FUNCTION

A requested function was not recognized by Anzio.

NO OUTPUT FILE OPEN

An operation requires an output file to be open, and there is none – use OPENO, then try again.

#### OVERFLOW ON RECEIVE

Data for an incoming file caused a communication chip overflow. The received file is incomplete.

#### RECEIVE BUFFER OVERFLOW

Data for an incoming file caused a receive buffer overflow. The received file is incomplete.

#### RESTRICTED COMMAND

You are using a restricted version of Anzio, such as Anzio Lite, which does not support the requested command.

#### UNABLE TO DELETE: *filename*

The indicated file does not exist, and so cannot be deleted.

#### UNABLE TO INITIALIZE COMMUNICATION

One or more of the current communication parameters is invalid. For example, the BAUD rate may be wrong, or the PORT number does not exist.

#### UNABLE TO READ KEY FILE: *filename*

The specified key file could not be found or was in the wrong format.

#### UNABLE TO TRANSMIT

Anzio is unable to transmit due to wiring, connection, or communication problems.

#### UNABLE TO WRITE TO OUTPUT FILE

Either the output file has filled its disk, or an output device (such as a printer) is not ready.

## Appendix B Notes On Particular Host Systems

---


Host-specific requirements are noted throughout this manual. This appendix collects some pointers for several host operating systems:

- *UNIX Hosts*


### B.1 UNIX Hosts

---

When working with UNIX, be sure its TERM setting agrees with Anzio's TERM setting. You can verify UNIX's setting by using the "echo" command at the Unix shell level:


```
echo $TERM 
```

You can set the TERM variable using the "set" command. To change UNIX's setting, tell UNIX:

```
TERM=VT220; export TERM 
```


for instance. Note that some application software on UNIX may assume you are using a certain TERM type, regardless of the current setting.

It is also important that UNIX be configured properly for backspace and "kill". If these are not set correctly, the characters '@' and/or '#' may not be processed correctly (especially during PC-to-UNIX file transfer). To make sure, tell UNIX that "erase" is *backspace* and "kill" is *ctrl-U*:

```
stty erase `^h`,kill `^u` 
```

This command can be made "permanent" by placing it where it will always be executed on startup. For the command to apply to an individual user account, place the above line in the file ".profile" in the user's home directory. For a command that will apply to every user, place the above line in the file "/etc/profile".

Anzio will issue XON/XOFF handshaking when necessary, and will also respond to it coming from UNIX. To make UNIX work correctly with XON/XOFF, do:

```
stty -ixany 
```



Finally, set Anzio for LOCK OFF, FULL DUP, BACKSPACE 8, and probably TAB OFF.

## Appendix C Migration From Earlier Releases


---

Anzio is constantly evolving to meet the needs of our users. If you are installing a new release, please read the README.TXT file on the distribution disk. This file contains specific instructions for migrating and details on release-to-release changes.

We try to maintain as much downward compatibility as possible. Old command forms will usually work, even when they have been superseded.

By default, Anzio will install in a directory dependent on its version. This means that a newer release of Anzio will not overwrite an older version. You may want to copy settings files (anything with a “.DEF” extension) from the old directory to the new directory, so you can take advantage of your prior configuration.

If you have problems with a new release, start Anzio with command line parameter NEW:




ANZIO NEW 

Anzio will start up in its “virgin” state, without reading a settings, just like the first time it was run a particular PC.

## Appendix D Serial Communication Problems

---

If Anzio does not want to communicate at all, it can be difficult to determine the cause. Try these suggestions first:

1. Verify that the cable from the host system is plugged into the PC *serial* port (I know it's obvious, but . . .).
2. make sure LOCK mode is off with Communicate/Lock Off, or enter  
LOCK OFF 
3. If the status line still shows LOCK, press   to unlock the keyboard.
4. Try setting Anzio for different BAUD rates.
5. Try setting Anzio for different comm ports.
6. Test the cable from the host: unplug the serial cable from the Anzio PC and try it on a CRT terminal. If the CRT doesn't work either, try a known good cable. If that doesn't work, check the line back to the host, and the host's terminal line settings.
7. Check the jumpers or switches on the PC serial port card. Make sure Windows' comm port setup agrees with the settings of the actual hardware.
8. Test the PC's serial port hardware with a "loopback test":
  - Disconnect the host cable from the PC. Configure the PC port so that it ends in a 25-pin female plug, using a 9-pin to 25-pin adapter and/or a female-to-female adapter as necessary.
  - Bend a small paper clip, and insert one end into hole 2 and one end into hole 3 on the exposed plug.
  - Now, if you type on Anzio's keyboard, the keystrokes should be echoed back to you (if you are in FULL DUP mode), or appear double (HALF DUP). If nothing shows, your serial port is not working properly.

- If the loopback test fails, you probably have a) two serial ports mapped to the same comm port number; b) two devices sharing an interrupt (IRQ); or c) a defective serial port.

---

➤ If you have not found the problem after trying these suggestions, contact Rasmussen Software for assistance.

---

## Appendix E Distribution Information

---

Anzio is distributed on CD-ROM, through a web download, or via email. The distribution contains a setup program (such as SETUP.EXE) which, when run, installs the Anzio executables and support files, including these types of files:

*.C	File transfer programs, written in C, for a Unix/Linux host system.
*.CAP	AcuCobol <i>termcap</i> file containing terminal settings.
*.DLL	Menu language files
*.DOC, *.TXT	Documentation files.
*.EXE	Executable (program) files.
*.HLP	Help information files.
*.KYS	Key definition files for various terminal emulation types. Most *.KYS files have an association *.DOC file. See <i>Defined Keys and Macros</i> , page 46.
*.PRT	Printcode files for FilePro to support Print Wizard.
*.TIC	<i>Terminfo compiler</i> files for UNIX host-side terminal settings.
*.TRS	Word Perfect support files.
*.UNI	Unicode mappings for various character sets – see <i>Using Anzio's Language Support</i> , page 38.

---


➤ The following list describes all distribution files in more or less alphabetical order, except the self-explanatory UNI (Unicode mapping) and KYS (key definition) files.

---

➤ See also the installation file README.TXT.

---

Anzio32.exe	AnzioWin program file. Might instead be Anzio32D.exe.
Anzio32r.exe	Anzio Lite program file.
ANZIO.CAP	A description of the ANZIO terminal type for Acucobol's A_TERM CAP file.
ANZIO.TIC	A UNIX terminfo file defining host terminal type TERM ANZIO – see <i>UNIX Host Configuration</i> , page 24. Used with ANZIO-TIC.KYS.
ANZIO-M.TIC	A UNIX terminfo file defining host terminal type TERM ANZIO-M for a monochrome PC – see <i>UNIX Host Configuration</i> , page 24. Used with ANZIO-TIC.KYS
ANZIO.TRS	A Word Perfect terminal definition file for the ANZIO terminal type.
ANZIO121.TRS	A later Word Perfect terminal definition file for the ANZIO terminal type. Provides better keymapping and more support for international characters.
ANZIOESP.DLL	The Spanish menu translations.
ANZIOFRA.DLL	The French menu translations.
AnzioTIC.KYS	Defined key file of function key definitions for use with Anzio.TIC and Anzio-M.TIC.
AnzioTIC.DOC	Documentation for AnzioTIC.KYS.
ANZIOWIN.HLP	The help file used by AnzioWin and Anzio Lite.
AT386.KYS	Defined key file for AT386 emulation.
C332.KYS	Defined key file for C332 emulation.
C332E.KYS	Also needed for C332 emulation.
DOWNLOAD	A UNIX shell script for downloading files to Anzio.

HOWTO.TXT	A text file describing how to implement some of the latest features in Anzio.
KERMIT.KYS	A file of function key definitions to work with Word Perfect on a UNIX system, when WPTERM is set for KERMIT.
PRINTWIZ.DOC	A Microsoft Word file containing instructions for AnzioWin's Print Wizard – see <i>AnzioWin's Print Wizard</i> , page 74.
PRINTWIZ.PRT	A printcode file for filePro to support Print Wizard, for monochrome printers.
PRINTWZC.PRT	A printcode file for filePro to support Print Wizard, for color printers.
README.TXT	A file of last-minute notes, etc. To see them, enter:  TYPE README.TXT 
RECV-PC.C	A UNIX source program that allows upload of longer records into UNIX.
SEND-PC.C	A C program used to download UNIX text files to a PC – see <i>UNIX Download Utility (Send-PC.C)</i> , page 85.
VT100.DOC	A file that explains the keys used in VT100.KYS
VT100.KYS	Sample function keys for VT100 emulation.
VT220.KYS	Sample function keys for VT220 emulation.
VT220.DOC	A file that explains the keys used in VT220.KYS
VT220N.KYS	A key definition file that more closely matches the "native" VT keyboard.
VT220S.KYS	Sample function keys for VT220 emulation. Uses a different approach to translating PC function keys to VT220 function keys than does VT220.KYS.
VT220S.DOC	A file that explains the keys used in VT220S.KYS.

VT-NUM.KYS	Defines the top four keys of the numeric pad to emulate the VT's PF1 through PF4 keys.
VT420PC.KYS	Key mapping for the VT420 with the PC keyboard.
VT8BIT.KYS	Key mapping for the VT, in which the initiator is hex 9B, instead of "ESC [".
VWPT.KYS	Key mapping for Viewpoint emulation.
WYSE60.KYS	Sample function keys for WYSE60 (and Wyse 50) emulation.
WYSE60.DOC	A file that explains the keys used in WYSE60.KYS
WYSE60N.KYS	Alternate keymap file for Wyse 60 emulation.



---

**Index**


---




exit Anzio 1  
 \$EDIT  
     defined key example 52  
 ^ (caret for Ctrl key) 36  
 <#>  
     pause in defined key 53  
     UNIX 216  
 <@>  
     UNIX 216  
 <{}>  
     display text in defined key  
         54  
 <|>  
     in defined key 52  
 <~>  
     quote character in defined  
         key 54  
 <1C> 192  
 <1D> 192  
 <alt-A> 174  
 <Alt-A> 50  
     end a SLEEP 166  
     panic button 38  
 <Alt-M> 50  
     Help key 37  
 <Alt-R> (review mode) 44  
 <Alt-U> 50  
     unlock keyboard 37  
 <Alt-X> 50  
 <BACKSPACE> 35  
 <BREAK> 18  
 <Ctrl-P>  
     editing 36  
 <DC2> 189  
 <DC4> 189  
 <DELETE> 35  
 <END> 36  
 <ESC>  
     editing 36  
 <F1>  
     editing 36  
     send defined key 37  
 <F10> 18  
     BREAK key 38  
 <F11> 47  
 <F12> 47  
 <F3>  
     function prefix key (NCR  
         7900) 37  
 <F4>  
     LOCK 140  
     unlock keyboard 37  
 <F5> 133  
     Help key 37  
 <F6>  
     call function in defined key  
         55  
     command entry 37  
 <F7  
     editing > 36  
 <F7>  
     with KEEP 136  
 <F9>  
     panic button 38  
 <HOME> 36  
 <INSERT> 35  
 <LEFT-ARROW>  
     editing 35  
 <RIGHT-ARROW>  
     editing 35  
 <TAB>  
     in defined key 52  
 132-column page  
     printing 71  
 7E1 102  
 7E2 103  
 7M1 103  
 7M2 103  
 7N1 103

7N2 103  
7O1 103  
7O2 103  
7S1 103  
7S2 103  
80- and 132-column screens 2  
8E1 103  
8E2 103  
8M1 104  
8M2 104  
8N1 104  
8N2 104  
8O1 104  
8O2 104  
add on programs for Far East  
    input 42  
ADD128 37, 104  
ADDS Viewpoint 26  
Allow Quit on/off 104  
alt key 36  
Alt-numeric Entry 41  
ANSWERBACK 105  
Anzio  
    /C choose settings file 184  
    /D define key 184  
    /H hostname 185  
    /K kiosk mode 185  
    /T communication type 186  
calculator 110  
    CALC command 110  
character encodings 38  
command line parameters  
    181  
commands 92  
    entering 34  
    from host 192  
    syntax 102  
communication parameters  
    14  
compatibility 218  
customizing 209  
error messages 212  
executables 2  
executing 13  
exit 23  
features 2  
files 221  
function keys 28  
host commands 192  
installing 9  
introduction 1  
kiosk mode 185  
languages 38  
menus 32  
NEW 182  
NONE 182  
Print Wizard 74  
quit 23  
settings file 183  
    choose 184  
SLEEP 166  
start defined key 182  
starting 13  
startup 181  
telnet parameter 183  
terminal type Anzio 27  
terminal types 25  
uninstalling 11  
upgrading 218  
VERSION 173  
versions 218  
WAIT 173  
ANZIO (terminal type) 169  
ANZIO.TIC 222  
ANZIO32.EXE 222  
ANZIO32D.EXE 222  
ANZIO32R.EXE 222  
ANZIO-M.TIC 222  
ANZIOTIC.DOC 222  
ANZIOTIC.KYS 222  
ANZIOWIN.DEF 24  
ASEND 105  
AT386 169  
AUTO-CONNECT 105

AUTO-LF 105  
automatic font selection 43  
AUX 105  
AUX serial devices 4  
AUX-DEBUG 106  
AUX-READ/S 106  
AUX-SETUP 106  
AUX-WRITE 106  
BACKSPACE  
    UNIX 216  
BACKSPACE 106  
BASENAME/S 106  
BAUD 106  
baud rate  
    checking 20  
    testing 219  
baud rate shift 63  
Baud Rate Shift 63  
beep  
    PITCH 150  
BEEP 107  
Beep Idle on/off 107  
Beep on/slow/off 107  
Bitmap  
    Cell 108  
    Watermark 108  
bitmaps  
    BMP commands 107  
blocking a defined key 49  
Blocktype 107  
BMP 107  
BMP-ADJUST 109  
BMP-CLOSE 109  
BMP-VIEW 109  
BMP-WINDOW 110  
BOX 110  
BREAK 38, 110  
browser integration 3  
buffer  
    PURGE 157  
buffer, keyboard 38  
C332 169  
C332 File Transfer 88  
cable  
    testing 219  
cable (serial port) 13  
    25-pin 13  
    9-pin 13  
CALC 110  
calculator  
    CALC command 110  
CALL  
    in defined key 57  
CALL 111  
CAPS (in status line) 18  
CAPTURE 111, 190  
    protocol 190  
capture display  
    to printer 72  
CAPTURE LONG  
    protocol 190  
caret ^ for Ctrl key 36  
CCII character codes 39  
CD 112  
changing menu language 44  
character encodings 38  
character set  
    languages 112  
Character Set of Output File 81  
characters  
    dynamic capture 81  
CHARSET 112  
CHOOSEPRINTER 113  
choosing a font 70  
choosing a printer 69  
CLIP 113  
clipboard (Windows)  
    copy screen to 29  
    paste from, in another  
        program 29  
CLOSEI 114  
CLOSEO 114  
Code Activated Switch Macros  
    59

- COLOR 115
- COLOREDIT 115
- command line parameters 181
- commands
  - about 92
  - alphabetical list 102
  - communication parameters 94
  - entering 34
  - file transfer 95
  - local processing 97
  - operator preference 93
  - sending from host 192
  - syntax 102
  - unicode 43
- Commands
  - compose 43
- COMMTYPE 115
- communication parameters 14
  - commands 94
- communications
  - command line parameter 186
  - hardware error 212
- compose key 40
- configuration
  - serial cable 13
- Configuration 5
- Configuring host system* 9, 24
- Connecting with host* 9, 11
  - Direct Serial Connection 12
  - Internet Dial-up 11
  - LAN 11
  - Modem 12
- connection
  - RECONNECT 159
- Connection methods 2
- connection problems 19
- control characters during capture 81
- control key 36
- Conventions 7
- COPY 119
- copy screen
  - to file 29
  - to printer 29
  - to Windows clipboard 29
- Copy-and-paste 5
- CURSOR [BLINK] 119
- customizing Anzio 209
- data bits 17
- DATA BITS 120
- Data Bits and Parity 62
- data capture 80
  - CAPTURE command 111
  - PURGE 157
- DDE commands 120
- dead key entry 41
- DEFAULTS 122
- DEFINE
  - usage 48
- DEFINE 122
- defined keys 46, 47
  - /D command line parameter 184
  - automatic execution 182
  - blocking 49
  - call on startup 182
  - calling another defined key 57
  - calling functions 55
  - copying 49
  - display text {} 54
  - editing 49
  - end-of-command | 52
  - executing 50
  - file selection 56
  - invoking another defined key 57
  - key definition files 50
  - keystrokes for functions 56
  - loading from file 51
  - menu selection 56
  - nesting 57

- overdefining 50
- pause # 53
- prompt user {} 54
- quote special characters ~ 54
- saving to file 51
- special characters 52
- startup 182
- tabs in 52
- undefining 48
- user input # 53
- DELAY 122
- DELETE 122
- Diacritic support 3
- DIAL 123
- DIR 123
- DIR/S
  - example 56
  - in defined key 56
- Direct Dial Modems 60
- direct serial connections 20
- display
  - capturing 80
  - COLOR 115
  - FONT 130
  - INTERPRET 134
  - MONITOR 143
  - PRINT 152
  - printing 71
  - SCREENMODE 162
  - SETCOLOR 165
  - STATUS LINE 167
  - TRACK-WINDOW 171
  - window TITLE 171
- DOWNLOAD 223
- download file transfer 83
- download script file (UNIX) 85
- DOWNLOAD-DIR 123
- DOWNLOAD-LOCK 124
- downstream file transfer 83
- duplex 20
- dynamic capture 81
- editing command line* 32, 35, 49
- editing defined keys 49
- EJECT 124
- eject paper 73
- END 124
- ending Anzio 23
- entering commands 34
- ENV/S 124
- ENVIRONMENT 124
- Error Control 62
- error diagnosis 219
- error messages 212
- Euro support 3
- EVENT 128
- execute defined key 50
- execute program
  - LAUNCH command 139
- exiting Anzio* 9, 23
- EXPECT 128
- expected data
  - WAITFOR 174
- faxing 3
- file transfer 5, 83
  - CAPTURE command 111
  - commands 95
  - download 83
    - protocol 189
    - UNIX 85
  - FIND/S 129
  - FINDNEXT/S 129
  - Kermit
    - UNIX 86
  - OPENI 146
  - OPENO 147
  - protocol 188
  - PURGE 157
  - RECV-PC.C 86
  - RETRANSMIT 160
  - SEND-PC.C 85
  - TIMEOUT 171
  - to printer 73

- TRANSMIT 172
- UNIX 84
  - upload 83
    - protocol 188
    - UNIX 84, 86
  - XN 176
- files
  - COPY 119
  - DELETE 122
  - DIRectory 123
  - FIND/S 129
  - FINDNEXT/S 129
  - MKDIR 143
  - OPENI 146
  - OPENO 147
  - PRINTFILE 154
  - READ 158
  - RENAME 160
  - TYPE 172
  - WINPRINT 175
  - WRITE 176
- FILL 129
- FIND/S 129
- FINDNEXT/S 129
- flow control 64
- FLUSH 129
- flush timer for printer 73
- FLUSHTIMER 130
- FONT 130
- font coverage 3
- FONT-ALT 130
- FRAMING error 212
- FTP 88
  - interactive 89
  - interactive FTP 5
  - new 131
  - Programmed 89
  - programmed FTP 5
- FTPDELETE 131
- FTPDELETE/S 131
- FULL DUP 132
- full duplex 20
- FUNCTION key (NCR 7900) 37
- function keys 28
  - defining 47
  - executing 50
  - overdefining 50
  - predefined 36
- functions in defined keys 55
- GAUGE 133
- gender converter 13
- Half DUP 133
- HALF DUP 132
- half duplex 20
- handshaking
  - UNIX 216
- hang up the phone line 61
- HANGUP 133
- help
  - online 31
- HELP 133
  - PRINT 133
- HOLD 134
- HOLD (in status line) 18
- host
  - /H command line parameter 185
  - in telnet address 183
  - name 134
  - sending Anzio commands 192
- Host and User Interaction* 92, 100
- host control 4
- HOSTNAME/S 134
- HTML
  - printing 75
- IGNULL 134
- initial Anzio terminal screen* 9, 17
- input method editors 42
- Installation 1
- installing Anzio 9
  - installing on the Host 10

PC 10  
 Interactive FTP 89  
 Interactive SFTP 89  
 internet connection 16  
 INTERPRET 134  
 INVOKE  
     in defined key 57  
 INVOKE 134  
 IP address 16  
 ISO/ANSI character codes 39  
 JUMP 134  
 KCOMMAND 135  
 KEEP 136  
     data capture 80  
 Kermit 88  
     commands 135  
     KCOMMAND 135  
     KRECEIVE 136  
     KSEND 138  
     UNIX 86  
 KERMIT.KYS 223  
 key definition file 50  
     loading 51  
     saving 51  
 key mapping 4  
*Key mapping* 92, 99  
 keyboard  
     encoding 40  
 Keys 201  
 KEYS  
     display defined keys 46  
 KEYS 136  
 keystroke entry 40  
 Kill 136  
 kill (under UNIX) 216  
 kiosk mode 4, 185  
 KRECEIVE 136  
 KSEND 138  
 Language support 2, 32, 38,  
     221  
 languages 38  
     CHARSET command 112  
     keyboard 40  
     printer  
         PASSTHRU-CP 149  
 Languages for menus 5  
 LAUNCH 139  
 library  
     kiosk mode 185  
 LINE DELAY 140  
 linedraw characters 3  
 local operations 4  
 LOCK 140  
 LOCK (in status line) 18  
 LOCK mode 219  
 lockdown 4  
 LOG 140  
 logging on 18  
 login  
     PASSWORD 149  
     RECONNECT 159  
     USERNAME 173  
 login wizard 19  
 Login wizard 2  
 LOGIN-WIZARD 141  
 loopback test 219  
 macro language 4  
 macros  
     code activated switch 59  
     executing 50  
     onerror 58  
     *processing* 92, 99  
     special uses 58  
     startup macros 58  
     unicode characters in  
         macros 43  
 macros (defined keys 46  
 MCISENDSTRING 141  
 MCISENDSTRING/S 141  
 menu  
     communicate menu 33  
     diagnose menu 33  
     edit menu 32  
     transfer menu 33

- view menu 33
- menu
  - file menu 32
- menu 33
- MENUBAR 142
  - example 57
  - in defined key 56
- Menulanguage 142
- menus 32
- MERGE
  - key definition file 51
- MERGE 142
- MESSAGE 143
- migration 218
- MKDIR 143
- modem
  - adapter 13
  - cable 13
  - hang up 61
  - signing off 61
- modem connections
  - connection speed 62
  - failure to establish
    - connection 63
  - making connection 61
  - setup 61
  - signing off 61
  - troubleshooting 63
  - types 60
  - using a modem to call the
    - host 60
- modem parameters 17
- MONITOR 143
- Mouse 204
- Mouse Action
  - mouse wheel 205
- mouse support 3, 9, 29
- Mouse wheel 205
- Multiple emulations 2
- NCR 7900 169
  - FUNCTION key 37
- NCR 7901 169
- Netscape
  - calling Anzio with host name 185
  - network connection 16
  - network connections 20
  - Network Modem Connections 60
  - non-ASCII characters 55
  - null modem 13
  - NUM (in status line) 18
  - numeric pad keys
    - defining 47
  - OLE 144
  - OLE/S 144
  - OPENI 146
  - OPENO 147
  - overdefining function keys 50
  - OVERRUN error 212
  - Palette 148
  - Panic button 38
  - panic button <F9> 38
  - parameters
    - names 183
  - parameters (Anzio command line) 181
  - parity 17
  - Parity 87
  - PARITY
    - error 212
  - PARITY 148
  - passthrough print 72
  - passthrough printing
    - PASSTHRU-CP 149
    - printer flush timer 73
    - protocol 190
  - PASSTHRU-CP 149
  - password
    - in telnet address 183
  - PASSWORD 149
  - PASTE 150
  - paste from clipboard 43
  - PICK



- data capture 80
- PICK 150
- PicLan 16
- pictures 3
- PITCH 150
- PLAYMEDIA 151
- PLAYSOUND 151
- port
  - checking 219
  - in telnet address 183
- PORT 152
- predefined function keys 36
- PREVIEW 152
- PRINT
  - data capture 80
- PRINT 152
- print capture 72
- Print Initialization Files 75
- print wizard
  - markup language 75
- Print Wizard 74
  - PRINTLEVEL 155
- printer
  - eject (flush) timer 73
  - FLUSH job 129
  - flush timer 73
  - FLUSHTIMER 130
  - font 71
  - open for output with OPENO 147
  - options 70
  - passthrough print 72
  - passthrough printing
    - protocol 190
  - PASSTHRU-CP 149
  - Print Wizard 74
  - PRINTER-SETUP 153
  - PRINTFILE 154
  - PRINTFONT 154
  - PRINTLEVEL 155
  - selecting 69
  - set with PRINTER 152
  - setup 70
  - slave print 72
- PRINTER 152
- PRINTER-SETUP 153
- PRINTFILE 154
- PRINTFONT 154
- PRINTFONT-ALT 155
- printing See printer
  - choosing a font 70
  - choosing a printer 69
  - end of print flush timer 73
  - HTML 75
  - print levels 66
  - printing the screen 71
  - setting printer options 70
  - troubleshooting 76
  - WINPRINT 175
- Printing
  - Auto-fit 6
  - barcodes 6
  - capture to printer 6
  - Forms overlays 6
  - levels 6
  - Markup Language 6
  - Mini-page 6
  - Multi-column 6
  - Multilingual 6
  - passthrough printing 6
  - print screen 6
  - printing HTML 75
  - printing the screen 71
  - PRINTLEVEL 155
  - PRINTLOW 156
  - Programmed FTP 89
  - Programmed SFTP 90
  - Prompt ID 157
  - PURGE 157
  - PWML 75
  - quitting Anzio 23
  - RAW-SETUP 157
- READ
  - key definition file 51

READ 158  
READ.ME 223  
Receive  
    CRC 190  
RECEIVE  
    protocol 189  
RECEIVE CODED 158  
RECEIVE QUIET 158  
RECONNECT 159  
RECORD 159  
RECV-PC.C 223  
    file transfer 86  
REGTELNET 159  
RENAME 160  
REPLACE 160  
RESET 160  
RETRANSMIT 160  
REVIEW  
    mode 44  
REVIEW 160  
RM/COBOL  
    under UNIX 216  
RUN 160  
run program  
    LAUNCH command 139  
    RUN 160  
    WINSTART 175  
running Anzio 13  
SAVE 162  
    defined keys 51  
saving a settings file 24  
saving settings 15  
SCOANSI 27, 169  
screen  
    capturing 80  
    PRINT 152  
    printing 71  
Screen 199  
screen capture 4  
Screen control 2  
screen review 4  
SCREENMODE 162  
SCREENMODE/S 162  
script 163  
Scripting 5  
SCROLL 164  
SCROLL ON  
    review mode 45  
scrolling  
    SCROLL 164  
    SCROLL-LOCK 164  
scrolling (Review mode) 45  
scrolling  
    JUMP 134  
SCROLL-LOCK 164  
Security 2, 202  
Security Options 194  
    Advanced 196  
    SRP 194  
    SSH 195  
send 164  
SEND 164  
SEND-PC.C 223  
    file transfer 85  
serial communication  
    problems 219  
*serial parameters* 14, 16  
serial port  
    cabling 13  
    troubleshooting 19  
serial port card 219  
Server mode 87  
SET 165  
SETCOLOR 165  
setting host's character set 40  
setting print encodings 40  
setting printer options 70  
settings file 15, 24  
    key definition file 50  
    startup 183  
    startup (ask) 184  
SETUP.EXE 10  
SFTP 89, 165  
    interactive 89

Programmed 90  
SFTPDELETE 165  
SFTPDELETE/S 165  
shift key 36  
signing off 61  
slave print 72  
SLEEP 166  
sound 3  
    PLAYMEDIA 151  
    PLAYSOUND 151  
special keys  
    defining 47  
    executing 50  
SRP 194  
SSH 195  
starting Anzio 13, 181  
startup settings file 183  
startup settings file (ask) 184  
status line 17  
    entering commands 34  
    error messages 212  
    GAUGE 133  
STATUS LINE 167  
STAYALIVE 167  
STOP 167  
STOP BITS 167  
stty (UNIX) 216  
Subdirectories 112  
SWITCHTO 168  
SZ 168  
T160E  
    emulation 104  
TAB  
    reset 168  
TAB 168  
TAB CHARACTER 168  
TAB ON 169  
TCP/IP 16  
Televideo 965 169  
telnet  
    command line parameter  
        183  
    host name 134  
    TERMNAME 170  
TERM  
    setting 216  
TERM 169  
TERM variable (UNIX) 27  
termcap 216  
terminal emulation  
    RESET 160  
Terminal features 2  
terminal types 25  
    function keys 28  
terminating Anzio 23  
terminfo (UNIX) 27  
TERMNAME 170  
Text vs. Binary 87  
TIME/S 170  
TIMEOUT 171  
TITLE 171  
TRACK-WINDOW 171  
TRANSFER-OPT 171  
TRANSMIT  
    protocol 188  
TRANSMIT 172  
TRANSMIT LONG  
    protocol 189  
TRANSMIT TRAILER 172  
transparent print 72  
troubleshooting 19  
    modem connections 63  
    printing 76  
TYPE 172  
UFT8 character codes 39  
undefining a defined key 48  
Unicode 173  
uninstalling Anzio 11  
UNIX  
    *download utility* 84, 85, 223  
    echo command 193  
    file transfer 84  
    handshaking 216  
    host configuration 24, 222

login 18  
notes 216  
Parity 87  
Server mode 87  
*simple download script* 84,  
85  
TERM setting 216  
terminal parameters 24  
terminal type Anzio 27  
Text vs. Binary 87  
upload utility 86  
upload file transfer 83  
UPPERCASE 173  
upstream file transfer 83  
User preference items 23  
username  
    in telnet address 183  
USERNAME 173  
USERNAME/S 173  
using Kermit 83, 84, 86  
using review mode 44, 160  
USMARC character codes 39  
VERSION 173  
video attributes  
    setting with COLOR 115  
VIEWPOINT 169  
VT100 25, 169  
VT100.DOC 223  
VT100.KYS 223  
VT220 25, 169  
VT220.DOC 223  
VT220.KYS 223  
VT220S.DOC 224  
VT220S.KYS 223  
wait  
    SLEEP 166  
WAIT 173  
WAITFOR 174  
WAITQUIET 174  
WF 174  
WINDOW 174  
WINDOWCLOSE 175  
WINPRINT 175  
WINSOCK  
    set with COMMTYPE 115  
WINSTART 175  
wiring (see cable) 13  
WLIBSOCK 16  
WRITE 176  
Wyse 60 26, 169  
Wyse50 169  
WYSE60.DOC 224  
WYSE60.KYS 224  
Xmodem  
    XRECEIVE 177  
    XSEND 177  
XN 176  
XON/XOFF  
    UNIX 216  
XRECEIVE 177  
XSEND 177  
Ymodem  
    YRECEIVE 178  
    YSEND 178  
YRECEIVE 178  
YSEND 178  
Zmodem 88  
    ZRECEIVE 179  
    ZSEND 180  
ZRECEIVE 179  
ZSEND 180