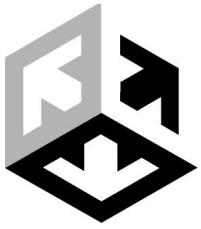
ANZIO

Personal Computer Communications

User Manual

Version 12.0 – November 1998



Rasmussen Software, Inc.

10240 SW Nimbus Ave. Suite L9 Portland OR 97223 USA
(503) 624-0360 www.anzio.com RSI@anzio.com

Copyright Notice

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 1998

Table Of Contents

Introduction

Anzio is a family of personal computer (PC) communications programs. Anzio provides *terminal emulation* 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 7.
- To stop and exit Anzio, press the ALT key and the X key at the same time:

  a X

  This will return you to the PC operating system.

➢ If you are migrating from an earlier version of Anzio, check Appendix E, *Migration From Earlier Releases*, page 113.

Anzio makes your PC function as a conversational, asynchronous terminal for many host operating systems, including:

- UNIX and variations
- VMS
- NCR IMOS (II, III, IV, V), IRX, ITX
- Ryan-McFarland RMCOS

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.

➢ Anzio does not provide "page mode" or "polling". If these are necessary for your system, contact us for assistance.

Pressing a keyboard key sends one or more characters to the host. Function keys (q) 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 terminal emulation. 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 23.

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

The Anzio family consists of six members, some running under DOS, some under Windows; some communicating via serial, some via networks such as TCP/IP:

Name	DOS or Windows	Serial or Network	Notes
AnzioWin Anzio32.exe	Windows	both	Uses either a serial or a network connection (TCP/IP, PicLAN, or Novell's LAN Workplace). Anzio for Windows 9x and NT provides extended Unicode character support.
Anzio Lite Anzio32r.exe	Windows	both	A limited-function version of Anzio for Windows.
Anzio for DOS Anzio.exe	DOS	Serial	The classic.
Anzio Small AnzioS.exe	DOS	Serial	A small version of Anzio for DOS, optimized for memory-resident operation.
Anzio for DOS Networks Anzio14.exe AnzioNet.exe	DOS	Network	Anzio for DOS, using either interrupt 14 (INT 14) to communicate through third-party network software, or INT 6B to communicate with Novell's LAN Workplace.

Key Features

Anzio is optimized for very fast local connection to host systems. Anzio goes beyond simple terminal emulation, using the power of your PC to create a "smart" terminal. Anzio's key features include:

Multiple emulations — Anzio emulates many popular terminal types: VT220, SCO ANSI, AT386, Wyse 50, Wyse 60, Versyss C332, ADDS Viewpoint/NCR 7901, and NCR 7900, among others.

Online help — Anzio supplies online help information for all commands and functions.

Language support — Anzio can display and send characters for many languages. Anzio for Windows 9x and NT uses 16-bit Unicode characters. Unicode defines about 39,000 characters used in European, Hebrew, Arabic, Chinese, Japanese, and Korean languages. If your host provides language support, Anzio can display and print your language.

Kiosk mode — Anzio can be run with a limited menu system, disabling unneeded user commands, for situations such as public-access library PCs.

Screen review — Anzio buffers data that scrolls off the top of the screen, including cleared screens. You can later see this data with Diagnose/Review or the REVIEW command.

Defined keys — You can define almost any key to send text or call Anzio commands. For example, a defined key can let the user select and transmit a file with one keystroke.

Host control — Programs running on the host system can control and configure Anzio and its PC, including PC file operations, DOS commands, 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.

Printing — Anzio can print the live screen, or incoming files. You can select any attached PC printer, and send it printer setup commands. Anzio for Windows lets you set the printer font and size manually, or automatically with the Anzio Print Wizard.

File transfer — Anzio provides a variety of file transfer protocols for different host systems and file types.

Copy-and-paste — Anzio for Windows 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.

You can select columns of data on the screen and write them to disk in comma-separated-values format, for later use with spreadsheets and word processing programs.

Configuration — You can configure many “comfort” features, such as colors, mouse click actions, non-blinking cursor, status/gauge line, beep pitch, etc.

Screen fonts — You can select any fixed-width font to use on the Windows screen, including such character sets as Cyrillic and Greek. You can set the font size and have Anzio resize the terminal window, or "zoom" the window to take over the whole screen.

80- and 132-column screens — Anzio for Windows provides 80- and 132-column screens, and custom screen sizes. Anzio for DOS supports 80/132-column hardware capabilities, or can maintain a 132-column virtual screen on an 80-column actual screen.

Calculator — A built-in calculator allows 4-function arithmetic in decimal or hex.

Memory-resident operation (DOS) — Anzio for DOS can be pushed into the background, and then can pop up at a keystroke. Anzio can also call the DOS command interpreter (shell).

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 Anzio for DOS*

- Appendix C, *Notes On Particular Host Systems*
- Appendix D, *Additional Programs*
- Appendix E, *Migration From Earlier Releases*
- Appendix F, *Serial Communication Problems*
- Appendix G, *Distribution Information*
- Appendix H, *Sample Defined Keys*
- Appendix I, *Anzio On a Network*

Conventions

- "Anzio" refers to any version of Anzio. Specific versions are noted as (Windows), (DOS), or (Network).
- "PC" is the personal computer running Anzio.
- "Host" is the computer to which you are connected. Host-specific comments are noted as (UNIX), (ITX), etc.
- f represents the "Carriage Return" character, generated by the "Enter" key on the PC keyboard.
- q 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*
- *Uninstalling Anzio*
- *Hooking Up The PC (Serial Connection Only)*
- *Starting Anzio*
- *Setting Default Anzio Communication Parameters*
- *Initial Anzio Terminal Screen*
- *Logging On to the Host System*
- *Setting User Preference Items*
- *Exiting from Anzio*
- *Configuring Your Host System*
- *Typical Terminal Types*

➢ The README.TXT file on the distribution disk provides 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 disk, because those program files are compressed.

Installing Anzio consists of:

- Transferring the program files from the distribution disk to your PC, either Windows or DOS.
- Optionally, transferring certain files to your host system, depending on the host system type and your file transfer requirements.

For more information on the contents of the distribution disk, and the system requirements for Anzio, see Appendix G, *Distribution Information*, page 117.

➢ 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 20.

➢ Anzio for Networks has special installation requirements. It comes in two versions, one configured for use with Novell software (AnzioNet.exe), and the other for use with redirectors such as TCP/IP drivers from FTP, Wollongong, or Lantastic (Anzio14.exe). See the installation file README.TXT for more information.

1.1.1 Installing Anzio on the PC—Windows

1. Insert the distribution disk (disk 1) into any available disk drive, typically the A: drive.
2. Start an Explorer or File Manager window and navigate to the disk drive.
3. Double-click on the SETUP.EXE program.

1.1.2 Installing Anzio on the PC—DOS

1. Insert the distribution disk (disk 1) into any available disk drive, typically the A: drive.
2. Change the current directory to the disk drive, for example:

   DOS> **A:** f
3. Use the supplied INSTALL command to transfer Anzio to the destination directory for example:

   A:\> **INSTALL C:\Anzio** f
4. The INSTALL batch file will take you through the whole process. If something goes wrong in the installation process, there is no harm in trying it again. If you continue to have problems, please call us.

➢ Anzio for DOS is capable of terminate-and-stay-resident (TSR) operation – see *Stay In Memory – STAY*, page 106.. You may wish to use the Anzio Small program for this purpose (AnzioS.exe).

1.1.3 Installing Files on the Host

For normal connection and terminal emulation, you will generally not need to do anything to your host system. However, the Anzio distribution disk includes some files for use on the host computer, to a) better support the 'Anzio' terminal type, and b) provide some kinds of file transfer. Because these files are generally **not** needed, we'll address them later.

1.1.4 Uninstalling Anzio

To uninstall Anzio, just go to the installation directory that contains Anzio, then delete all files. Finally, delete the installation directory itself. Anzio does not install any components outside its installation directory.

If you put Anzio transfer programs on the host, delete those files.

1.2 Hooking Up The PC (Serial Connection Only)

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 or 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.

1.2.1 Configuring the Serial Port

Your PC must have a *serial port* to talk to the outside world of host computers, modems, serial printers, and other devices. Some PC serial ports are built in, some are on add-in boards, and some are part of an internal ("integrated") modem.

All serial ports have a unique *address* and an *interrupt vector* ("IRQ"). These are generally set by jumpers or switches near the communications chip itself. Serial ports on the PC's motherboard may be configurable from the BIOS setup program. Internal modems may be "plug-and-play", and be set automatically by Windows.

(Windows) Windows maintains configuration settings for all serial ports. Anzio requests a port number. Thus, if AnzioWin is set for PORT 2, it will tell Windows to open communication on port 2, and Windows must know how your port 2 is configured.

(DOS) Anzio accesses the serial port directly through the specified PORT and IRQ. Ports 1 and 2 are defined by the PC industry standard. Anzio defines ports 3-6 for other typical configurations.

Port	Addresses	IRQ
1	3F8-3FF	4
2	2F8-2FF	3
3	3E8-3EF	4
4	2E8-2EF	3
5	3220-3227	4
6	3228-322F	3

- In rare cases, your serial port is at an unusual location. You can store the port address in the reserved BIOS location referenced by Anzio. Please contact us for more information.

(DOS) If your IRQ does not correspond to the PORT as above, Anzio lets you set it explicitly - see the IRQ command in the reference section.

- The IRQ used by a communication port must not be shared by any other device, including another communication board or modem. This is a PC design restriction, not an Anzio limitation.

1.2.2 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.

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.

If you are using a modem, connect a standard phone line between the modem input and the phone jack. You may also be able to plug your telephone into the modem output.

- You are welcome to call us for assistance in setting up Anzio. However, some communication problems are the result of misbehaving PC hardware. Any reputable PC dealer can install and test a serial port or internal modem.

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)

➢ Anzio can operate with only 3 wires: ground, SD, and RD. To use this minimal wiring, the host system may need some jumper wires on its side. Call us for details.

1.3 Starting Anzio

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

(DOS) Change to the appropriate directory, and run the Anzio program:

```
DOS> C: ↵
C:\> CD \AnzioDir ↵
C:\AnzioDir> Anzio ↵
```

➢ The Anzio program name could be Anzio, AnzioNet, AnzioS, or Anzio14.

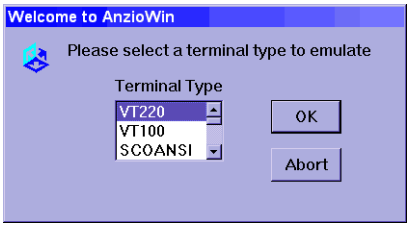
Anzio will prompt for communication parameters, described in the next section.

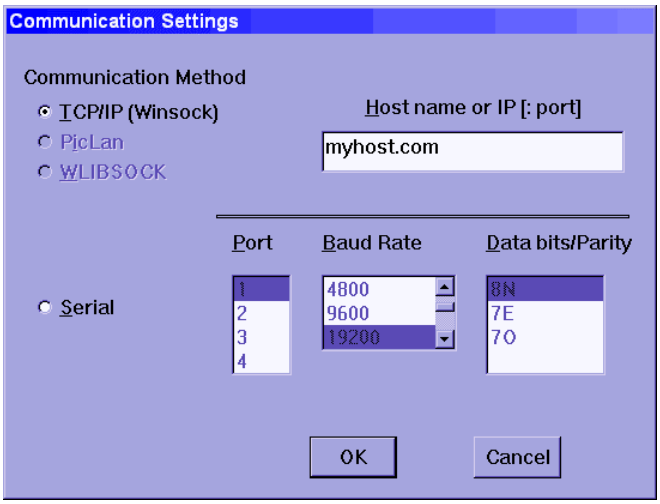
1.4 Setting Default Anzio Communication Parameters

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 for Windows 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:





These parameters will be saved in a *defaults file*, either AnzioWin.def for Windows, or Anzio.def for DOS. The defaults file can also contain system settings such as tab stops, printer font, etc.

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

➢ If you get the message UNABLE TO INITIALIZE COMMUNICATION, you have probably told Anzio to use a PORT that does not exist on your machine. Double-check your PORT setting – see *Configuring the Serial Port*, page 8.

1.4.1 Terminal Type

Anzio can emulate many different terminal types. The host is probably expecting a particular terminal type. Select or enter 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 15, 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 (TCP/IP, PicLan, or Novell's WLIBSOCK).
- Enter either the host name, or an IP (*Internet protocol*) address with an optional host port number.

1.4.2.2 Serial Parameters

A serial connection requires a communications port (attached directly to the host system), or a previously configured modem – see *Using A Modem to Call the Host*, page 33.

Port — Anzio needs to know which serial port is connected to your host computer – see *Configuring the Serial Port*, page 8.

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 or enter the baud rate at the prompt.

Modem connection rates can be from 1200 to 56000 baud. Direct serial connections are probably 9600 or 19200 baud.

If the PC-to-modem baud rate is not the same as the modem-to-modem baud rate, your modem must be configured for a baud rate shift – see *Baud Rate Shift*, page 34.

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 or even, referring to the sum of data bits in each byte. 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 or enter these parameters to match the host.

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

1.5 Initial Anzio Terminal Screen

By now you should be looking at a blank screen with a *status line* at the bottom. Anzio for Windows 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, a U, or the LOCK OFF command.

HOLD — The output from the host to Anzio has been suspended, with either the HOLD command or the [key.

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 is also used to enter Anzio commands, called with b j F. We'll get into that later. For now, let's see if we can communicate.

1.6 Logging On to the Host System

(UNIX) If you are connected to a UNIX system, but haven't seen a login prompt, press f. You should see the UNIX "login:" prompt. Enter your login name and password just as you would with a terminal.

(I-series) If you are using an NCR I-system, your terminal must be *logically attached* before the host system can "see" it. On IRX/ITX, this must be done from another terminal. On IMOS, you may be able to use b ~ to self-attach, depending on how your system's SYSGEN settings; if not, use another terminal. To attach your terminal from another terminal enter ATTACH *terminal_number*:

AT (*n*) f

(IRX, ITX) You must log on like a terminal, using either b C and f, or b ~. If your login doesn't "take" the first time, try it again – press o first to unlock your keyboard. Anzio provides some defined "login" keys – see *Defined Keys for NCR I-Series*, page 120.

(Other hosts) On other systems, log on as from a normal terminal.

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, *Troubleshooting*.

To use Anzio menus and enter commands, see *Talking with Anzio*, page 20. The *Command Reference* section, starting on page 55, describes all commands.

1.6.1 Troubleshooting

This section describes some common problems and typical solutions:

- 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, or a M to get the "menu" screen and enter:

  BAUD 9600 f

  or whatever rate is appropriate.
- If you can't see the characters you're typing, turn off Communicate/Full Duplex, or enter:

  q HALF DUP f
- If you see two of every character you type, turn on Communicate/Full Duplex, or enter:

  q FULL DUP f

- 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 or the command q TERM.

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

1.7 Setting User Preference Items

After Anzio is communicating, you may want to change some of its display behavior. *User preference items* are parameters that affect the way the Anzio "terminal" looks and sounds.

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.

Anzio for Windows provides many of these items on the View menu. You can also use the corresponding commands, listed under *Operator Preference Items*, page 55.

(DOS) You may want to set Anzio for DOS to use an 80-column or 132-column display, depending on your PC screen hardware. See *Setting the Screen Size*, page 107.

1.8 Exiting from Anzio

To exit Anzio, press the a 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 (key file), you will be prompted for the save file names. For now, just press f to accept each default file name.

Each time Anzio starts up, it looks for a defaults file named AnzioWin.def (Windows) or Anzio.def (DOS). If the file is found, Anzio loads those settings. A defaults file may include the name of a key file – if so, those defined keys are read in (see *Loading Key Definitions*, page 28).

The first time you run Anzio, or if the defaults file is not found, Anzio will ask you for the basics: terminal type and connection parameters.

You can use multiple defaults files, and multiple key files, for different Anzio configurations (for different users or hosts) on the same PC. See *More On Starting Anzio*, page 86.

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 C, *Notes On Particular Host Systems*, page 109.

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 and VT220*
- *WYSE 50 and WYSE 60*
- *ADDS Viewpoint (NCR 7901)*
- *SCOANSI*
- *Terminal Type "Anzio" (UNIX only)*

Your host system or application software may only support one particular terminal type.

➢ If you have a choice, set Anzio for VT220 mode, and set your host system for VT100 or VT220. UNIX hosts may support "Anzio" mode – see *Terminal Type "Anzio" (UNIX only)*.

Some terminals support various *character attributes*, which affect the appearance of characters on the screen. Character attributes include reverse video, blinking, underline, intensity, etc.

➢ The function keys on your PC can be mapped to terminal function keys – see *Using Function Keys in Terminal Emulation*, page 17.

1.10.1 VT100 and VT220

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 VT100 has only four function keys and a few special keys. The VT220 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. This is Anzio's most powerful operating mode.

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. Local key definitions are not saved. The KEYS display (a M KEYS) will show both definitions, with the local definition on top. See *Defining a Key Macro*, page 26.

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 (NCR 7901)

The ADDS Viewpoint (NCR 7901) 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.

1.10.5 Terminal Type "Anzio" (UNIX only)

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:

Note that the Anzio.TIC file assumes a color capability on your PC. **If you are running a monochrome PC** , use "Anzio-M" below instead of "Anzio".

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 50. 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   f
   ```

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   f
   ```

6. Finally, tell UNIX to initialize the terminal:

   ```
   tput init   f
   ```

- 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.

1.10.6 Using Function Keys in Terminal Emulation

Who owns the function keys? That is, if Anzio is emulating a WYSE 60 (which has function keys), and you press p, does that cause a certain keycode to go to the host, or does it invoke Anzio's HELP screen?

To use function keys with your host software, you can either load in one of the supplied key definition files, or you can define the keys yourself (see *Defined Keys and Macros*, page 23).

To load a key definition file, use File/Read Keys or the READ command.

Supplied key definition files, such as VT220.kys, are listed in *Appendix H, Sample Defined Keys*, page 119. 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.

- Even though you have defined the terminal function keys, you can still access Anzio's default function keys. For example, if you define p to send a certain sequence to the host, you can still get Anzio's HELP screen. You can use a p, j p, or b p if you haven't yet defined them. You can also use Anzio's shortcut key a M, if you haven't defined it.

  See *Overdefining Anzio's Default Function Keys*, page 28.

With some host software, this special key business can get quite confusing. For instance, consider the spreadsheet package 20/20. The documentation for 20/20 refers to the PAGE DOWN } key. If Anzio is emulating a VT100, for instance, a PAGE DOWN should be PF1 y. If you have loaded VT100.KYS, l on your keyboard will generate the control code equivalent to the VT100's PF1 key. So, you could get a PAGE DOWN by pressing l y. If you use 20/20 a lot, you might want to define your } key to send the entire combination – see *Defined Keys and Macros* page 23.

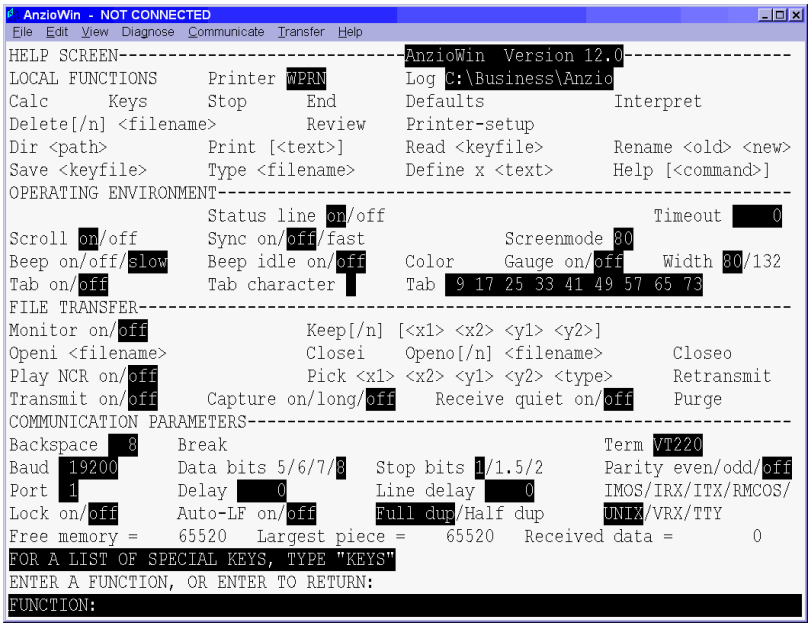
2 Help When You Need It

Anzio provides two kinds of online help:

- Press a M or the p help key to get a basic Help screen, showing current Anzio settings.
- Press a H to call up the Help menu (Windows) or Help index display (DOS).

2.1 The p Help Key And Screen

Press a M or the p key to bring up the HELP screen, which shows Anzio settings and commands:



At the “FUNCTION:” prompt, you may enter:

HELP *command*
: For example, HELP BAUD will bring up information on the BAUD command.

HELP INDEX
: Displays all Anzio commands. Use the arrow keys to move the highlight to a command, then press f to see it. Press ^ to exit.

HELP ASCII
: Displays ASCII keys and hex codes. Press f to exit.

KEYS
: Displays the current defined keys, if any.

command
: Any Anzio command.

f or ^
: Exit from this HELP screen.

Help information comes from a file named Anzio.hlp, which must bein the same directory as the Anzio program file (Anzio.exe).

2.2 Help Topics and Index – a H

(Windows) Press a H to call up the Help menu, which provides access to on-line help, and support information. The Help/Contents item brings up a typical Windows help display, showing Anzio commands and help topics. Click on a topic for more information.

(Windows) You can also go straight to help on any Anzio menu item by highlighting that menu item and hitting l .

(DOS) Press a H to call up the help index page, which lists all Anzio commands. Use the arrow keys to move the highlight to a command, then press f to see it. Press ^ to exit.

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 for Windows provides menus to access most Anzio commands. The following sections describe:

- *Anzio Menus*
- *Entering Commands*
- *Editing the Command Line*
- *Predefined Function Keys*

(Windows) You can use the mouse to select words and regions on the screen, and to read the word under the cursor.

- To select, move the mouse cursor to the start or upper left corner of the desired region. Hold the left mouse button down while dragging the cursor to the end of the region.
  - Print the region with File/Print, or save it to a file with the KEEP command.
  - If you hold the a key down when you release the button at the end, the 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.
- The default action for a left button double-click is to send the word under the cursor to the host, followed by a f . You can customize the actions performed by mouse button clicks, for example, to turn off the added f , or to start a Web browser with the URL under the cursor – see*Customizing Anzio*, page 98.

3.1 Anzio Menus

Anzio for Windows 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 I

File Menu

- Open input and output files, change directories
- Save defaults 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 lineat the bottom of the screen. Anzio displays “FUNCTION:” on the status line when it is expecting a command. There are two ways to get into command entry mode:

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

p or a M — As above, but also displays the HELP screen, showing Anzio commands and the 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 f 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 23.
- The host computer – see *Sending Anzio Commands From The Host*, page 96.

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.

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.

x , z
: Moves the cursor right or left.

h
: Switches between *insert* and *overtype* mode. In insert mode, displays INS at the right end of the command line.

c
: Deletes the character at the cursor.

g
: Moves the cursor to the beginning of the command.

d
: Moves the cursor to the end of the line.

l
: Delete-to-end-of-line – erases all characters from the cursor to the right end of the line.

r
: Used with the KEEP, PRINT, and PICK functions to define a rectangular area (column) on the live screen – see *Data Capture*, page 43.

b P
: *Prefix key* 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 P ^ . Without the prefix, ^ is interpreted by Anzio as an editing command (next).

^
: Throws away the edited command and returns to the live screen.

Function keys entered as part of a command are shown in reverse video with the function key name, such as F5. If a prefix key is used with the function key, the prefix keyis shown by:

s	j
^ (caret)	b
a	a

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

The command line entry can be up to 255 bytes long. The line pans over if you type past the right margin.

3.4 Predefined Function Keys

Anzio predefines several function keys for its own use. For instance, t is the "grand abort" key. 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 a M KEYS to see the current function key assignments.

If a function key has been overdefined, you can still access the Anzio function using a *modifier key*. For example, if your configuration uses n, you access Anzio's default n with j n, a n, b n, or b j n. 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 28.

Some of Anzio's default function keys also have an a key, including b j F for Function, a M for help Menu, and a X for eXit.

l	"Send defined key". Press l, then a defined key – that key is INVOKEd and the results are sent to the host. See *Executing Key Macros*, page 27.
n	*Function prefix key*, used for FUNCTION key. 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 f. Press f alone to exit the Func: prompt without sending anything.
o or a U	Temporarily unlocks the keyboard.
p or a 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.).
q or a F	"Anzio function" – displays the FUNCTION prompt for you to enter Anzio commands. Note that in Windows, a F will pull down the File menu. So instead use b j F
t or a A	PANIC BUTTON (Grand abort) – aborts a defined key in progress, turns TRANSMIT OFF, exits any endless loops, and empties the keyboard buffer.
u or b ~	BREAK key – sends the BREAK signal to the host computer. b ~ can also be pressed at any time to clear the type-ahead buffer and unlock the keyboard.
a H	Displays the Help menu (Windows) or Anzio's help index (DOS).
a X	Exits the Anzio program.

3.5 Using Anzio's Language Support

Anzio supports many different character encodings. 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, using more than one byte per character.

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

- ISO/ANSI Western, equivalent to ISO 8859-1.
- DEC – very similar to ISO.
- OEM – sometimes called "linedraw" or "DOS". Corresponds to the DOS codepage on your Windows installation.
- ISO-8859-x, where "x" is 2 to 9. These are used in Europe and elsewhere.
- DOS and Windows code pages, including Far East. 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

➢ To use these encodings, you must set the terminal emulation type to VT100, VT220, SCOANSI, or ANZIO.

There are three areas affected by character encodings, as described in the following sections:

- Characters sent by the host must be displayed properly.
- Characters typed on the keyboard 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.

➢ Anzio provides only incoming and outgoing character *encodings*. You may also need to obtain *fonts* that contain the characters you wish to see.

➢ For language independence and ease of translation, Anzio stores all characters internally in 16-bit Unicode.

3.5.1 Setting the Host Communication Code

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 the Keyboard Encoding

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 our Windows, and then you can switch between them with a special key combination. See your Windows documentation for more information.

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.

We are still investigating options for inputting Far East characters on non-Far East Windows systems.

3.5.3 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 labelled “Character set for passthrough print”, similar to what was described above.

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

3.6 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).

- To start REVIEW mode, press a R, or use Diagnose/Review or the REVIEW command. Exit REVIEW mode with f or ^.

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.

(DOS) If you have WIDTH set to 132 on an 80-column screen, you can use REVIEW to look to the right (or left) at characters that are out there on the virtual screen

3.6.1 REVIEW Mode Scroll Keys

REVIEW mode uses only the following keys:

f or ^	Restores the working screen and exits REVIEW mode.
w, y	Moves the view window up or down one line.
{ , }	Moves the view window up or down one screenful.
z, x	Moves the view window left or right one column, if possble.
g , d	Moves the view window to the far left or far right, if possible.
P	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.

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

q `DEFINE D DEALLOCATE` f

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

➢ q or l may be overdefined in your environment – see *Overdefining Anzio's Default Function Keys*, page 28.

Each defined key can contain up to 246 keystrokes. In addition, the available memory can limit the total number of keystrokes possible. Available memory is displayed in the HELP screen.

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. Anzio provides some key definition files for use with I-series and RM/COS operating systems – see Appendix H, Sample Defined Keys, page 119.

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

q `KEYS` f

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*).

➢ If you are having problems with sending key definition characters ('System Overload', or dropping characters), Anzio's character transmission rate may be too fast. See the DELAY command.

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).
- Special keyboard keys, including function keys, named keys such as g , left and right a keys, and the numeric pad period _ . The numeric pad keys are only accessible with] off, except that separate definitions are possible for numeric pad / * - + e with] on.
- ASCII control codes, hex 00 through hex 1F.

Function keys l through u can be defined four ways each: normal, with j , with b , and with a . Also, if your system supports keys F11 and F12, Anzio will support them also, both normal and shifted.

4.2 Defining a Key Macro

Key macros are defined with the DEFINE command:

q `DEFINE` *`x string`* f

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*.

➢ 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*:

q `DEFINE` *x* f

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, w, y, ^, b C, and so forth.

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

q `DEFINE` b C `{}` f

If you later press b C, nothing happens – the b C 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 30.

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 22.

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

q `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 f to finish editing.

4.3.1 Copying a Defined Key

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

q `DEFINE X`?

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

4.4 Executing Key Macros

To execute a key macro defined for a regular printable key, press l then the key, for example, l X.

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

4.5 Overdefining Anzio's Default Function Keys

Anzio's predefined function keys, such as p Help and q Function, can be *overdefined* with your own definition:

q DEFINE p . . .

Anzio predefines all variations on its function keys as well: a p , b p , and j p will all display the Help screen unless overdefined.

You can also use a F for p , and a M for q – see *Predefined Function Keys*, page 23.

➢ We strongly advise you not to redefine a A , a F , a M , a U , or a X .

4.6 Using Key Definition Files

When Anzio starts up, it can read a *key definition file*. The key filename is specified either on the command line or in the defaults file.

If you want Anzio to start up without reading any key file or defaults file, use the command line parameter NONE – see *'Anzio NONE' For Clean Startup*, page 87.

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 defaults file for use the next time.

➢ If you change a key definition, then exit from Anzio using File/Exit, Windows Close (X), END, STOP, STAY, or a X , 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:

q SAVE *filename* f

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:

q READ *filename* f

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

q MERGE *filename* f

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 f*
- # number sign – see *Waiting for User Input*
- {} braces – see *Displaying Text to the User*
- ~ tilde – see *Quoting Special Keys With '~'*
- q 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 v 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 f

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

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 TEXTF ILE(2),OW |
          AS LO (LP) |
          EX $EDIT,RE |    f
```

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

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

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 transmitting a DEFINE string, the Anzio program will stop transmission, accept your entry up to a f and send the entry (the f 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 |    f
```

Then, when you enter l 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 f . The "defined key" then takes over, and sends:

```
(3)  f
AS B
```

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

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

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 |   f
```

- You may include cursor movement characters 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}# |     f
```

When you enter l v, Anzio sends:

```
vi   k
```

and then displays on the PC screen:

```
Enter file to edit
```

After you enter a file name and press f , Anzio sends the filename and a f (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}   f
```

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.

4.8 Using Anzio Functions in a Defined Key

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

Two useful functions are DIR/S and MENUBAR – see *File and Screen Selection Functions*, page 31.

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

```
DEFINE S  q OPENI UPFILE |  q TRANSMIT |  f
```

- Each embedded q function must end with a vertical bar |, as shown, to represent the end-of-function f .
- q is recognized inside a defined key, even if that function key is overdefined – see *Overdefining Anzio's Default Function Keys*, page 28.

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  q OPENI # |  q TRANSMIT |  f
```

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:

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

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

```
q  DEFINE  {  q REVIEW |  {
```

defines the { key to start REVIEW mode ("q 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 (/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  q DIR/S *.DAT | # | |
          q TRANSMIT |  f
```

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 q 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
          q MENUBAR 1 13 1 23 0 0 1 13 | #| |
          f
```

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 q 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.

5 Using A Modem to Call the Host

If your PC is connected to the host over a telephone line, you will need to use a modem. The modem translates bytes to and from the host into serial bits for the phone line.

Modems are either inside the PC case (*internal*) or connected to the PC's serial port (*external*). They work the same in either case.

A modem connection is a serial connection, with two differences: 1) you have to tell the modem to call the host, and 2) a modem is often slower than a direct serial connection.

The typical use of a modem connection is:

1. Set modem parameters as necessary (host phone number, start bits, etc.).
2. Have the modem dial the host and return a modem response code.
3. Wait for the host to establish a connection.
4. Work on the host.
5. Sign off from the host, which hangs up the modem.

When Anzio starts up, it is talking to the modem, not the host computer. To call a host, you can either give commands directly to the modem, or use Anzio's DIAL command.

5.1 Using Direct Modem Commands

All commands to the modem start with AT for "ATTENTION". Modem commands are entered on the live screen, not on the status line. Remember that Anzio does not know it is talking to a modem instead of a computer.

A typical modem command to dial a number is:

```
ATDT5550101
```

where DT indicates *Dial Touchtone*. See the manual that came with your modem for instructions on how to use modem commands.

➢ With some older modems, any commands you give to the modem must be in UPPER CASE.

5.2 Calling the Host with the DIAL Command

Anzio provides a DIAL command to tell the modem to dial the host. If the host does not answer on the first try, the DIAL command can wait and retry several times. The DIAL command is also menu item Communicate/Dial, and its default key is Alt-D.

The syntax is:

```
DIAL string wait retries  f
DIAL 5551234 100 6    f
```

where *string* is a phone number, and *wait* is how long to wait before redialing, if necessary, up to *retries* times (max 32767). *string* can also contain modem commands, as explained below. *wait* is measured in tenths of a second (max 32767, or 52 minutes). *wait* and *retries* are optional, and are 0 if not specified.

The example above sends the string 5551234 to the modem as the phone number to dial (that is, preceded by "ATD"). The modem dials the number and returns a response code (see next section). If the response indicates that a connection was made, the DIAL command is complete. If a connection is not made, the PC waits 10 seconds each time before retrying, up to 6 more times. If the connection was still not made, Anzio displays an error message.

➢ You can stop the DIAL process at any time with the t panic button, or Transfer/Grand Abort.

The *string* parameter can contain any other modem setup characters necessary, such as T for touch-tone, before the phone number(s). For example, you can define a key to dial a phone number and then send a credit card number:

q DEFINE D DIAL T05552478,,,,123456789 f

When you press l D, Anzio tells the modem to dial touch-tone, the number 0-555-2478, wait for four counts (a delay to wait for the telephone company's computer to accept a credit card number), and then send the credit card number 1-2345-67-89.

5.2.1 Modem Response Codes

After each command sent to the modem, the modem returns a response message. The DIAL command reads the response message, expecting to see the word CONNECT for a successful connection. A connection failure is assumed if the response message contains NO CARRIER, NO DIALTONE, NO ANSWER, or BUSY.

Some modems use different response codes, which are not standardized. If Anzio displays "BAD MODEM STATUS", you must configure your modem to respond with the basic response codes. Refer to your modem manual.

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, displaying a modem response code. At the point, Anzio is once again talking to the modem, not the host.

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

5.4 Baud Rate Shift

Many modems will do a *baud rate shift*, meaning that the baud rate between PC and modem is different from the baud rate over the phone line (modem to modem).

For example, say you have a 28800 baud modem, and your Anzio is set to 38400 baud. You dial into a remote host that has a 2400 baud modem. Generally, your modem will "sync up" with the remote modem, and start communicating with it at 2400 baud. A properly configured modem will tell you that it connected at 2400 baud rather than 38400.

Another modem configuration parameter determines whether the modem will do a baud rate shift. That is, the modem will either continue to communicate with the PC at the prior baud rate (38400), and use handshaking to control the flow of data, or it will *shift* its baud rate and begin talking to the PC at the modem-to-modem rate (2400 baud).

If the modem shifts its PC-to-modem baud rate, there will be problems. Anzio has no way to recognize this change, and will no longer be able to communicate with the modem.

- For Anzio to work, your modem must be configured to maintain a constant baud rate for its connection to the PC. The alternative is to hang up and change Anzio's PC-to-modem baud rate to match the host (2400 baud in the example above), then redialing.

Note that the host modem has the same potential problem. If the host modem changes its baud rate, the host system may no longer be able to communicate.

- You will generally want to set Anzio's baud rate HIGHER than the rate at which the modems can connect, in order to take advantage of data compression that many modems can provide.

5.5 Flow Control

In most modem situations, some sort of flow control will be necessary. We suggest configuring your modem to use software (XON/XOFF) flow control (also called handshaking). Contact us for more information if you are having problems.

6 Printing

Anzio can do several kinds of printing:

- printing the current screen or window
- printing incoming text as it is displayed
- printing incoming text without displaying (*passthrough print*)
- sending an incoming file transfer directly to the printer

6.1 Choosing a Printer

To select the printer output device, use File/Printer Setup, or the PRINTER command. The printer may be attached to the PC (*local*), or located on a network. Anzio simply sends data to the selected printer name.

(DOS) Anzio for DOS provides output to any DOS device name, such as "PRN" or "LPT2".

(Windows) Anzio for Windows lets you use the current default Windows printer, or any installed printer. Anzio for Windows also supports DOS device names, and adds a virtual Windows printer named WPRN. The WPRN module uses the device-independent Windows printer interface, so that Anzio can change fonts and other settings. See *Windows Virtual Printer – WPRN*, page 38.

6.1.1 Setting Printer Options

(Windows) 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.

(DOS) DOS does not provide printer device information, so Anzio has no way to send printer-specific commands, to change fonts or line spacing. Each type of printer has its own command codes (escape sequences). The manual for your printer will list these command codes. Use the PRINTER-SETUP command to send command codes, for example, "compressed pitch".

The next time you print something, Anzio will use the current printer settings. Anzio saves changes to your printer settings in its defaults file (Anzio.def or AnzioWin.def).

6.1.2 Choosing A Font (Windows)

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, will line up in columns. TrueType fonts will give you more options for size. OEM fonts have more line-drawing characters.

We recommend "Courier New" for most applications. Anzio also supports various special character sets for use as printer fonts – see the PASSTHRU-CP command.

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*). For a 132-column page, you will probably want to change your font size down to 7 point.

➢ The *Anzio Print Wizard*, page 38, can be set to scan incoming printer text, then fit the text to the page automatically.

6.2 Printing The Screen

To print the currently active screen, use File/Print Screen, a P , or the PRINT command.

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

(Windows) File/Print Screen or a P always goes to the WPRN module, regardless of the setting of PRINTER. The output uses the current WPRN settings for printer selection, font, Print Wizard, etc. You can use the mouse to select the printed region (drag across the screen). The background bitmap, if any, will also be printed.

To print the screen during a REVIEW, press P .

To print displays generated by DIR, HELP, INTERPRET, KEYS, or TYPE, enter PRINT f .

- Printing the screen does not automatically eject the page – see *End-of-Print Flush Timer*, page 37.
- To print any buffered printer data and eject the page, use File/Eject, or the EJECT command.

6.3 Capture To Printer

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

(Windows) To start print capture, check menu item Transfer/Capture to Printer, or the command CAPTURE WPRN. To stop, uncheck menu item Transfer/Capture to Printer, or use CAPTURE OFF.

(DOS) Start print capture with the command OPENO LST: followed by CAPTURE ON. Stop with CAPTURE OFF and then CLOSEO.

6.4 Passthrough Printing

Anzio supports *passthrough print*, also called *transparent print* or *slave print*, to send incoming host data directly to the printer. During passthrough print, the host data is not displayed.

Passthrough print is controlled by the host system, using terminal-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.

6.5 File Transfer To Printer

To send incoming file data to the printer, using Kermit or similar transfer utilities, specify the PC output file as a printer device name (such as LPT1:, LST:, or WPRN).

6.6 End-of-Print Flush Timer

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

(Windows) 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, or the EJECT command. Any data left in the DOS printer buffer or Windows print job is printed, followed by a page feed.

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.

If passthrough print data arrives sporadically from the host, with pauses, you may need to increase the FLUSH TIMER.

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 command, or when you exit Anzio.

6.7 Windows Virtual Printer – WPRN

Anzio for Windows provides a virtual printer, WPRN, which sends print data directly to the standard Windows printer interface. Anzio uses the device-independent features of WPRN to set font types and sizes, and to provide "print levels".

6.7.1 Print Levels

The Windows model for printing, used in the WPRN module, is to "draw" text at various places on the page. Data sent to a Windows printer may be formatted by Windows in the current style (*high level*) or sent to the printer for formatting (*low level*). You can also bypass any Windows interpretation, using *raw* mode to send the data as-is.

To set the print level, use File/Print Level or the PRINTLOW command.

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.

If the data from the host contains escape codes or PostScript code, set the print level low. With print level low, the data is sent to the printer like writing to a DOS device, but still goes through the Windows print spooler.

6.8 Anzio Print Wizard

Anzio'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 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, providing such things as 80-column versus 132-column reports, backspace-underlining, embedded escape codes, and line wrap.

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.

6.8.1 Print Wizard Markup Language

The Print Wizard's internal markup language provides the programmer a way to control Anzio's printing in a device-independent manner. This ranges from job-level changes, such as setting the paper orientation and point size, to precisely-specified items such as bitmaps and rectangles.

If you would like to use this feature, contact Rasmussen Software to get documentation for the Print Wizard markup language.

Part II Data Capture and File Transfer

7 Data Capture

Anzio means more than just terminal emulation. You can capture data directly from the host, either right off of the live screen or through file transfer. This section describes screen capture.

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

You can capture screen data as characters (KEEP), as data columns (PICK), or as printed text (PRINT):

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 r 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. Use File Print Screen or a P to print the screen. You can also use PRINT r to select a part of the screen.

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 some cases the host will limit us to 80-byte records. In nearly all cases we are limited to passing ASCII data only (no packed numerics, etc.).

A *simple upload* involves sending sequential ASCII records, up to 80 characters, upstream. Most text files can be sent to the host with this method. See:

- *I-System Simple Upload*, next section
- *UNIX Simple Upload*, page 50

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:

- *ITX File Transfer*, page 44
- *UNIX File Transfer*, page 49

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 ITX* Kermit, page 49
- *Using UNIX* Kermit , page 51

➢ Rasmussen Software also provides an optional host program that can transfer almost all file types: the*Universal File Transfer Utility Program*, described on page 52.

8.1 ITX File Transfer

This section describes several file transfer methods for I-series operating systems:

- *I-System Simple Upload*
- *I-System Download Utility (SEND-PC.CBL)*
- *I-System Upload Utility (RECV-PC.CBL)*
- *I-System Upload Spool File Utility (RECV-SPL.CBL)*
- *Using ITX* Kermit

➢ Each utility program is provided in COBOL 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*I-System Download Utility (SEND-PC.CBL)*, page 45.

8.1.1 I-System Simple Upload

An I-system simple upload has Anzio transmit a text file into either $EDIT or SYSIN.

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

8.1.1.1 Upload to $EDIT

1. Tell the host to set up the ITX destination file and start $EDIT for text insertion:

   AS A *filename*(*unit*),NEW,*size*,AP f
   AS EWF EWF(*unit*),OW f
   EX $EDIT f
   IN* f

2. Tell Anzio to open the PC source file with File/Open File to Send, or:

   a F OPENI *pcfilename* f

3. Tell Anzio to transmit the PC file, specifying a backslash \ as the character to follow the last sent record. Use Transfer/Transmit with Trailer, or:

   a F TRANSMIT TRAILER \ f

4. When Anzio finishes transmission, tell the host to save the file and quit $EDIT:

   SA A f
   QU f

8.1.1.2 Upload to SYSIN

1. Tell the host to set up the ITX destination file and copy SYSIN to that file:

   AS A *filename*,NEW,*size*,AP f
   MOV SYSIN A f

2. Tell Anzio to open the PC source file with File/Open File to Send, or:

   OPENI *pcfilename* f

3. Tell Anzio to transmit the PC file, specifying “END$” as the string to follow the last sent record. Use Transfer/Transmit with Trailer, or:

   TRANSMIT TRAILER END$ f

4. After Anzio sends all records and the final “END$”, the host will end its MOV and be ready for the next command.

8.1.2 I-System Download Utility (SEND-PC.CBL)

The Anzio installation disk includes a flexible file download program named SEND-PC.CBL. This utility program is designed to transfer a variety of file types to the PC. You may also add your own code for special file types, using the existing source code and comments as a guide.

To use SEND-PC.CBL, you must first upload the PC source file to the host, as described in the previous section. Next, you will need to compile the program on the host.

To compile a COBOL source file on an I-system:

AS SI *sourcename*(*unit*) f
AS BO *objectname*(*unit*),NEW,*sectors*,AP f

```
EX $COBOL  f
```

The "$COBOL" compiler name above might be "$COBOL9" or "$COBOL85". If you get a message that a printer is not assigned, just press f .

8.1.2.1 Upload and Compile Example

The following lines show how to upload and compile the SEND-PC program, placing it on host unit 3. All these commands are issued to the host system, except the Anzio commands prefaced with a F :

```
AS A SEND-PC-S(3),NEW,200,AP    f
MOV SYSIN A  f
 a F OPENI SEND-PC.CBL  f
 a F TRANSMIT TRAILER END$  f
AS SI SEND-PC-S(3)  f
AS BO SEND-PC(3),NEW,200,AP    f
EX $COBOL  f
 f
```

8.1.2.2 Using the SEND-PC.CBL Utility

The SEND-PC program sends one or more host files to the PC. SEND-PC sends Anzio commands for local PC file OPENs and CLOSEs as necessary.

The simplest method is to tell the host:

```
EX SEND-PC( unit) itxname(unit) pcfilename  f
```

This will send host file *itxname* to your PC file *pcfilename* using the default settings.

For greater control, first set switches, then assign a host input file and execute SEND-PC:

```
SET SWITCH 2, 3 OFF  f
AS A filename(unit) f
EX SEND-PC( unit) f
```

In this example, the switches indicate that each line in the destination file includes trailing spaces (2 OFF) and has a CR/LF (3 OFF). See the description of switches, below.

SEND-PC prompts for the destination PC file name. You may include a directory path as part of the PC file name. To transfer to the printer, use a file name of "LST:".

If the PC file already exists, SEND-PC asks if you want to delete it. Answer Y to continue, or N to stop without transferring the file.

Now all you have to do is sit back and watch the lines (records) go by. You can speed up transfer by turning off the incoming line display, using Transfer/Receive Quiet or the RECEIVE QUIET ON command.

- If you see file status 95 or 98, the file you are trying to transmit doesn't match one of the file specs of the SEND-PC program.

SEND-PC produces a text file on the PC, where each host record becomes a "line" ending with a carriage-return/linefeed (if switch 3 is OFF). If there are non-ASCII characters in the file, such as packed numerics, the results are unpredictable. If the file being read is a spool file, SEND-PC will convert page positioning information into the appropriate line feeds and form feeds.

The following file specs are supported by SEND-PC:

BLOCK	FIX/VAR	RECORD	
512	V	80	
512	V	510	
480	F	80	
512	F	128	
512	F	256	
512	V	132	
512	V	134	(spool file)
512	F	512	
512	V	256	
470	F	94	
512	V	87	(spool file)
512	V	90	(spool file)
495	F	95	

SEND-PC uses the following switches:

SWITCH 1 ON means "use alternate codes". Use SWITCH 1 ON if you have modems or multiplexors that interfere with passage of XON and XOFF characters (hex 11 and 13).

SWITCH 2 ON will strip trailing spaces off each line before sending it.

SWITCH 3 ON does **not** put a CR/LF on the end of each record.

SWITCH 4-6 Used only with multiple files, as described below.

SWITCH 7 ON indicates an early version of ITX (before 4.0)

SWITCH 8 ON will filter control codes out of the data. This slows the transfer down considerably.

SEND-PC can send a group of files, given a parameter file containing the host file names and optional parameters. Create a file on the host system, using $EDIT, that contains one line for each file to be transferred. Each line must contain the I-system name, with disk unit, in columns 1 through 30. Columns 31-72 may contain additional options:

SHORTEN, col. 31
Tells whether to strip trailing spaces. Y for YES, N for NO, or a space to use the switch 2 setting.

CRLF, col. 32 Tells whether to put carriage-return/linefeeds on the PC file. Y to add CR/LF, N not, or a space to use the switch 3 setting.

PCNAME, cols. 33-72
The path and name of the file to be created on the PC.

When used with a parameter file, SEND-PC checks three additional switches:

SWITCH 4 ON means to delete an existing PC file.

SWITCH 5 ON means to ignore an existing PC file. Ignored if switch 4 is ON (the file will be deleted).

If both switches 4 and 5 are OFF, and there is an existing PC file, you will be prompted to overwrite or ignore the PC file.

SWITCH 6 ON means skip any host file name that is not found, and move onto the next file name in the parameter file. OFF will cause SEND-PC to stop and ask whether to continue.

To use the parameter file with SEND-PC, tell the host system:

```
AS PRM prmfilename(unit) f
EX SEND-PC( unit) f
```

8.1.3 I-System Upload Utility (RECV-PC.CBL)

This program is designed to upload PC files with records that may be longer than 80 bytes. RECV-PC is set up for a variety of file specs, and others can be added as described in the source file.

To upload a file from the PC, first assign the host destination file, then call RECV-PC:

```
AS A filename(unit),NEW,size,AP f
EX RECV-PC( unit) f
```

RECV-PC will first ask you for the name of the PC file to be uploaded. The program will then ask you to select the type of file you want to create.

If your PC file does not have carriage-return/linefeeds between records, tell RECV-PC to create fixed-length records of the same size.

RECV-PC does not have currently support parameter files or command-line parameters.

8.1.4 I-System Upload Spool File Utility (RECV-SPL.CBL)

RECV-SPL uploads a print file from the PC to the host for printing on a faster host printer. The program is designed to upload records up to 132 characters (for 132 column width paper). RECV-SPL truncates any records over 132 characters.

Transfer the RECV-SPL.CBL source to the host using Anzio, and compile it as RECV-SPL-O, as described under *Upload and Compile Example*, page 46.

To upload a PC file and print it directly on the I-system's printer, tell the host:

```
AS LO (unit,LP) f
EX RECV-SPL-O( unit) f
```

To print out to a spool file for later printing, tell the host:

```
AS LO spoolfile,NE,size,SP,AP f
EX RECV-SPL-O( unit) f
```

RECV-SPL will ask for the name of the print file on the PC to be uploaded. At the end of the transfer, RECV-SPL displays the number of lines received by the host, and the number of truncated lines (those over 132 bytes).

As RECV-SPL transfers the lines, they will appear on the screen and scroll up. Use Transfer/Receive Quiet or the RECEIVE QUIET command to enable or disable this display. The transfer will be faster without displaying lines.

- Some PC word processors and spreadsheets allow the user to build files with special characters to control printer spacing, fonts and other options. The host printer may not recognize these printer-specific codes. For example, some programs print bold characters by *overprinting*, that is, printing the characters twice – this may show up on the host printer as two lines.

- RECV-SPL can easily be modified by your own in-house programmers for your local printing requirements.

8.1.5 Using ITX Kermit

Since release 6, ITX has included a Kermit file transfer module named $IKERMIT, although it is only officially supported for ITX Windows.

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

$IKERMIT is a server-only implementation of Kermit. To start it, tell ITX:

```
EX $IKERMIT  f
```

At this point, ITX will only accept Kermit server commands, such as KSEND, KRECEIVE, and KCOMMAND FINISH. These commands are described beginning on page 60.

To terminate $IKERMIT, tell Anzio:

```
a  F KCOMMAND FINISH  f
```

$IKERMIT assumes that you want to transfer to and from disk unit 0. There are two ways around this:

- You can start $IKERMIT on a specified disk:

  ```
  EX $IKERMIT 3  f
  ```

- Once $IKERMIT is running, you can change its working disk by using the CWD (Change Working Directory) command. Tell Anzio:

  ```
  a  F KCOMMAND CWD 3  f
  ```

$IKERMIT accepts several command-line parameters:

J	Journal – log file transfers.
D	Debug – display extra information during file transfer.
B	Binary – send and receive binary files, rather than text.
n	Disk unit number, default 0.

We are still exploring $IKERMIT's operation, and we may be putting together more information. Feel free to contact us.

8.2 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 UNIX* 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 50.

- The COBOL file utility programs (such as SEND-PC.CBL) described in the previous sections are not intended for UNIX, and will probably not work.

8.2.1 UNIX Simple Upload

You can upload a PC files with up to 255-byte records directly to a UNIX host.

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

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

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

```
a F OPENI pcfilename f
a F TRANSMIT TRAILER b D f
```

- 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.

8.2.2 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 f
   ```
3. To download a file, tell UNIX:

   ```
   download filename f
   ```

`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 COBOL source files in the current directory, tell UNIX:

```
download *.cbl f
```

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

8.2.3 UNIX Download Utility (Send-PC.C)

The SEND-PC program will transfer virtually any UNIX text file down to the PC (do not try to use it on an RM/COBOL indexed file). You will need to upload the C source file Send-PC.C and compile the program on the host.

First, tell UNIX:

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

Then, tell Anzio:

```
a F OPENI SEND-PC.C f
a F TRANSMIT TRAILER b D f
```

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 f
```

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* f

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

8.2.4 UNIX Upload Utility (Recv-PC.C)

The program RECV-PC.C is included for use with PC files having records longer than 255-byte records. You will need to upload and compile Recv-PC.C as described above.

To transfer a file, tell UNIX:

recv-pc *unixname pcname* f

8.2.5 Using UNIX Kermit

The Kermit file transfer protocol is available for UNIX for a small distribution charge. 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 for Windows 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 60.

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, as many are, you will need to set up Kermit that way too. To do so on startup, tell UNIX:

kermit -p e f

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

set parity even f

Finally, it is possible to put the "set parity even" command in a file named ".kermrc" in you 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 DOS) or BINARY (which does no translation). To see its current setting, tell UNIX Kermit:

sh f

To set it , tell UNIX Kermit:

set file type *x* f

where *x* is `text` or `binary` .

SERVER MODE

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

server f

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

To terminate the server mode, tell Anzio:

a F `KCOMMAND FINISH` f

To terminate Kermit itself, tell Kermit:

`quit` f

8.3 Special Types Of File Transfer

The previous sections describe how to transfer simple files as a whole. You may also want to transfer a small portion of a host file, or to transfer more complex files.

➢ If you have special data transfer requirements, please call us.

8.3.1 Universal File Transfer Utility Program

Rasmussen Software has developed a Universal File Transfer (UFT) utility program for use with Anzio and either NCR or UNIX hosts. UFT transfers text and binary files in both directions while providing error-checking for each record.

- **(I-Series)** UFT for I-systems runs on an IMOS III, IMOS V, IRX, ITX, or RM/COS host. UFT can transfer any host file to the PC: indexed, relative, sequential, packed numeric, 8-bit data, and object programs. UFT can also transfer PC files to the host: text, binary, and RM/COBOL-compatible binary sequential files.
- **(UNIX)** UFT for UNIX transfers text, binary data, and 8-bit data files. UFT for UNIX also provides multiple file selection (with wild-carding) and can compress/uncompress files.

With UFT on different hosts, you can use the PC as a stepping stone to move files from one system to another:

- I-system to and from:
  - I-system
  - RM/COS
  - RM/COBOL on PC
  - UNIX
- RM/COS to and from RM/COBOL on PC

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 21.
- A defined key macro – see *Defined Keys and Macros*, page 23.
- From the host computer – see *Sending Anzio Commands From The Host*, page 96.

- Anzio commands applying only to the DOS, Windows, serial, or network versions are marked as such.
- Anzio Lite does not include some commands.

9.1 Commands By Type

This section lists Anzio's commands by type:

- *Operator Preference Items*
- *Communication Parameters*
- *File Transfer*
- *Local Processing*

Commands are described in alphabetical order beginning on page60.

9.1.1 Operator Preference Items

BEEP	Turns beep on or off
BEEP IDLE	Beeps when idle
BMP	Sets a background bitmap on the screen
COLOR	Sets operating colors/attributes
CURSOR BLINK	Allows non-blinking cursor
FONT	Sets the screen font size
GAUGE	Shows gauge line at bottom of screen
JUMP	Sets jump scrolling
PITCH	Sets beep pitch
PRINTFONT	Sets printer font size
PRINTLOW	Sets low-level print

SCREENMODE	Sets various hardware 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
ANSWERBACK	Sets the Anzio terminal's answerback string
AUTO-LF	Sets auto-linefeed
AUX	Turns on/off aux input
AUX-SETUP	Configures aux input
BACKSPACE	Configures the Backspace key
BAUD	Sets the baud rate
BREAK	Sends a Break character
CHARSET	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
IRQ	Sets the system network interrupt vector
ITX	Anzio is talking to an ITX system
LINE DELAY	Sets the delay for line turnaround
LOCK	Sets the keyboard locking protocol
MONITOR	Communication diagnostics
PARITY	Sets the parity type
POLL	Sets the communication polling frequency
PORT	Sets the communication port number

RECONNECT	Reconnect on line drop?
RESET	Resets the Anzio terminal
RMCOS	Anzio is talking to an RM/COS system
RTS-MODE	Sets RS-232 Request-To-Send mode, for use with some communications equipment
SCROLL-LOCK	Does the [key suspend host output?
STOP BITS	Sets the number of stop bits
SYNC	Removes screen “snow” for CGA
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
TTY	Sets “dumb” terminal mode
UNIX	Anzio is talking to a UNIX system
UPPERCASE	Changes keystrokes to upper case
VRX	Anzio is talking to a VRX system

9.1.3 File Transfer

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
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
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
PLAY NCR	Makes the PC appear to be an NCR host, for use with PC-to-PC communication
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
TIMEOUT	Sets the file transfer timeout
TRANSMIT	Provides simple file upload to host
XN	Sends next TRANSMIT record
ZRECEIVE	Receives a file using Zmodem
ZSEND	Sends a file using Zmodem

9.1.4 Local Processing

BOX	Draws a box on the screen
CALC	Invokes the built-in calculator
CALL	Calls a macro key as a subroutine
CD	Changes the logged disk/directory
CHOOSEPRINTER	Opens the Printer Setup dialog box
CLIP	Copies screen to Windows clipboard
COPY	Copies one PC file to another, optionally sending result to host
DATE	Sends PC date to host
DEFAULTS	Saves settings
DEFINE	Defines a macro
DELETE	Deletes a file from PC, optionally ignoring "file not found"
DIAL	Dials a modem
DIR	Lists a PC disk directory, optionally sending to host
DROPOUT	Exits from Anzio without resetting comm port
EJECT	Ejects a printer page
END	Quits Anzio
ENV/S	Sends an environment variable to host
ENVIRONMENT	Sets Anzio internal environment variables
FILL	Fills a screen area with a character
FLUSH	Releases a print job to the printer
FLUSHTIMER	Sets timed release of print jobs
HANGUP	Hangs up a modem
HELP	Gets help on Anzio

HEXPATCH	Sets certain hidden parameters
HOSTNAME/S	Sends host name to host
HOTKEY	Sets hotkey for DOS terminate-and-stay-resident (TSR) use
INVOKE	Stops current macro and starts another macro
KEYS	Shows special and macro keys
LAUNCH	Starts another program in Windows
LOG	Changes logged disk/directory
MENUBAR	Creates a menu
MERGE	Merges a macro file with macros in memory
MESSAGE	Displays a message box
MKDIR	Makes a directory, optionally sending result to host
MODE-132	Sets display mode for 132 by 25
PAN	Moves right or left in virtual screen
PASSWORD	Sets the password for the host, optionally sending password to host
PASTE	Pastes Windows clipboard text
PLAYSOUND	Plays a WAV file
PRINT	Prints all or part of screen
PRINTER	Sets the printer to use
PRINTER-SETUP	Configures PC's printer
PRINTFILE	Prints a text file
PRINTLOW	Configures Anzio's Windows print driver
RAW-SETUP	Configures Windows raw printing
READ	Reads a key macro file
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
SCROLL	Saves erased data
SEND	Transmit (part) of the screen to the host
SETCOLOR	Reset current color
SLEEP	Wait until certain time
STAY	Go to TSR mode, optionally reserving memory for graphics

STOP	Quits Anzio
TIME	Sends PC time to host
TYPE	Shows a PC file on screen
USERNAME	Sets the username for the host, optionally sending the username to host
VERSION	Displays Anzio version number, optionally sending result to host
WAIT	Waits a specified time
WAITFOR (WF)	Waits for specific incoming text until timeout
WIDTH	Sets 80- or 132-column virtual screen
WINDOW	Opens a text window on the screen
WINDOWCLOSE	Closes a text window
WINPRINT	Asks Windows to print a file using the associated Windows application
WINSTART	Starts a Windows program
WRITE	Writes a string to the current output file

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.

Commands are applicable to all flavors of Anzio except where noted as (Windows) or (DOS).

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.

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.

8E1

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

8E2

Shortcut for DATA BITS 8, PARITY EVEN, 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 8th bit, and sends the result to the host. By default, assigned to n and all variations (j n , b n , a n).

ANSWERBACK *string*

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

AUTO-LF [ON] | OFF

AUTO-LF ON causes Anzio to send a line-feed character after every carriage-return (f).

AUX [ON] | OFF

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-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.

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).

BAUD *nnnn*

Sets the connection's baud rate, in bits per second. All standard rates are supported: 50, 75, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 28800, 38400, 57600. The current baud rate is displayed by the p (HELP) key.

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

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.

BMP *filename* [*window-rect* [*view-rect* [*bright* [*contrast*]]]] **(Windows)**

Tells AnzioWin to open *filename*, which must be a BMP or a GIF file, and display it as the screen background.

If you print the screen with a background bitmap, the background is also printed.

Optional parameters indicate which part of the window is affected, which part of the bitmap to use, and, for 256-color mode only, brightness and contrast settings.

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. A value of -1 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 described under *window-rect*, above.

The *right* and *bottom* numbers may be 0, to represent the bitmap's right and bottom edges.

The bitmap may 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 may also be set with the BMP-VIEW command.

bright Brightness, between 0 and 1000. Only used with 256-color mode. May also be set with the BMP-ADJUST command.

contrast Contrast, between 0 and 1000. Only used with 256-color mode. May also be set with the BMP-ADJUST command.

BMP-ADJUST *bright* [*contrast*] **(Windows)**

Sets the background bitmap's brightness and contrast, as described above.

BMP-CLOSE **(Windows)**

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* **(Windows)**

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

BMP-WINDOW **(Windows)**

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

BOX *c1 r1 c2 r2* [type]

This command draws a box on the screen from character position (*x1*, *y1*) to (*x2*, *y2*). The optional *type* is the line style, either SINGLE (default) or DOUBLE.

The corner offsets may be either absolute (from the upper-left screen character) or cursor-relative. For example, BOX .-1 .-5 10 20 indicates a box from one character to the left of the cursor (.-1) and five up (.-5) to screen position 10,20.

BREAK

Sends an ASCII BREAK character to the host computer, held on for 300 milliseconds.

CALC

Starts a simple four-function calculator on the status line. The calculator can work in decimal or hex mode.

The calculator has several commands:

+ - * /	Addition, subtraction (minus), multiplication, division.
f	Functions as a '+=' key.
c	Clears the last entry. If pressed twice, clears the total. '
^	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 **(All)**
CAPTURE *filename* | CLOSE **(All)**
CAPTURE WPRN **(Windows)**

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.

Synonym for the LOG command.

CHARSET *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.

language	CANADIAN, FINNISH, FRENCH, GERMAN, ITALIAN, NORWEGIAN, SPANISH, SWEDISH, SWISS, UK

Name can be 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 (ISO 8859-1) character set
OEM	The DOS or "linedraw" character set.
ISO-8859-*x*	*x* is 2 through 9
DEC	Slightly different from standard ISO
CP-*xxx*	*xxx* is the number of a Windows or DOS *character codepage*, such as 150. Requires Windows support for the particular codepage.
T160E	Innovative Interface's T160E multilingual font
CCCII	Chinese, Japanese, and Korean combined character set
UTF8	Byte-encoded Unicode characters
CUSTOM	Allows use of a user-editable translation file, CUSTOM.UNI

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

CHOOSEPRINTER [*name*] **(Windows)**

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 r1 c2 r2*] **(Windows)**

Copies the screen text (or just the given region) to the Windows clipboard. 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.

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 2^4, or 16, possibilities.

PC video drivers do not provide all these options. Many color PCs replace these options with different colors.

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 ^ to leave the COLOR dialog without changes, or f to save the changes.

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

COMMTYPE SERIAL | WINSOCK **(Windows)**

Tells Anzio whether to communication with the host system using a serial connection or a Windows Socket (WINSOCK) connection. Changing the COMMTYPE causes an existing connection to be dropped.

`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

`CURSOR [BLINK] [ON] | OFF`

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

`DATA [BITS] 5 | 6 | 7 | 8`

Sets the number of data bits in the communication protocol, usually 7. Note that this bit count does not include the parity bit, if any.

`DEFAULTS`

Saves the current parameter settings to a file. You will be prompted to enter the name of the 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 26.

`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

`DIAL` *`phone_number`* [*`wait retries`*]

Tells the modem to call *phone_number*, optionally waiting *wait* tenths of a second between *retries* times if no answer. See *Calling the Host with the DIAL Command*, page 33.

`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 f , that subdirectory's files are displayed.

Note that you may 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 31.

`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.

`DROPOUT`

Exits from Anzio (like END or STOP) but does not restore the communication port to its original state. This is for use in unusual situations only.

`EJECT`

Ejects a page from the printer – see the PRINTER command.

`END | E`

Stops Anzio, restores the communication port to its original state, and returns to the operating system. Same as STOP. Can be abbreviated as E.

`ENV/S variable`

Sends the value of the DOS environment *variable* to the host, terminated by a RETURN. See also ENVIRONMENT.

`ENVIRONMENT string`

Sets Anzio's internal environment *string*. This string is sent to the host system at the beginning of a telnet session, if the host system requests it. Typically used with UNIX systems.

string is a list of environment variables and their values, separated by semicolons. The format for each variable is:

name=*value*

or

name=${env}

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 pseudo-environment variables:

ANZ_HOSTNAME	Host name
ANZ_IP	PC's IP address
ANZ_LAST_RECD	Name of last file received
ANZ_PASSWORD	Value set by PASSWORD
ANZ_USERNAME	Value set by USERNAME

`FILL` *`c1 r1 c2 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 name matching the name or wildcard entry in *filespec* and sends it to the host. See also FINDNEXT/S.

`FINDNEXT/S`

Finds the next file name matching the wildcard entry in the previous FIND/S command, then sends the name to the host.

`FLUSH`

Sends any remaining data in the print buffer to the printer, by closing and reopening the printer file. If spooling is enabled on the printer (network or Windows local), this will release the print job for printing. See also FLUSHTIMER.

`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` **`(Windows)`**

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.

name	Font name, enclosed in quotes if it contains spaces, such as "Courier New".
size	Font size, as either *height* or *height*x*width*.
LARGER	The next larger font size, for example, from 10 to 12.
SMALLER	The next smaller font size.

Note that your PC may not have the indicated font and size.

`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`

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. Used primarily with a serial connection.

`HELP`
`p`
`HELP` *`command`* `| INDEX | ASCII`

The HELP command or p key displays many available commands, any open file names, and the current option settings. Press f to exit the HELP screen and return to your "working" screen.

In the HELP screen, you may enter a command without pressing q. 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.

Help files are kept in the installation directory. If this drive or directory is not accessible, no help will be available.

command	Displays information on a particular Anzio command. *command* must be spelled out completely, not abbreviated. See *Help When You Need It*, page 18.
INDEX	Displays the help topic index. Select a topic and press f to see the available information.
ASCII	Displays a chart of ASCII characters.

`HEXPATCH` *`address bytes`*

Directly sets certain Anzio internal parameters. See the installation file README.TXT for examples.

Use this command with caution, as you could kill Anzio.

`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 hostname to the host. The host name can come from a command line parameter, a telnet address, or from the network parameters.

`HOTKEY` *`xyy`* **`(DOS)`**

Sets Anzio's wakeup key combination, used after the STAY command. The *xyy* parameter is 3 hex digits, where *x* is the shift mode and *yy* is a keyboard scan code. Shift modes for *x* are:

8 = alt
4 = ctrl

2 = left shift
1 = right shift

For keyboard scan codes, enter HELP HOTKEY.

Anzio's standard HOTKEY is a l , coded as 83B. An alternative is Ctrl-LeftShift-A, coded as 61E.

`IGNULL [ON] | OFF`

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.

`IRQ` *`n`* **(DOS)**

Tells Anzio the interrupt level (IRQ) for the current communication port.

The PORT command sets the IRQ to the standard value for that port. If you are using a non-standard setup, you will need to follow a PORT command with an IRQ command.

`ITX | IRX | IMOS | RMCOS | VRX | UNIX`

Configures Anzio for the given host operating system, primarily the keyboard locking mode. Use HELP or p to display the current host operating system setting. Use DEFAULTS to save this setting for future use.

`JUMP OFF | MEDIUM | FAST` **(Windows)**

Tells Anzio how to scroll the screen for rapid incoming data:

OFF	Shows every line (default).
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

KEEP [r | *c1 r1 c2 r2*]
KEEP/N [r | *c1 r1 c2 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 r	Starts a stretchable KEEP rectangle in the center of the screen. Usethe arrow keys to move one corner of the highlighted rectangle to include the data you want to keep. Press r to nail that end down. Move the other corner to cover the desired KEEP area. Press f to display the selection coordinates on the command line.
c1 r1 c2 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 and their text (see DEFINE). Also shows the system-defined function keys, and any non-obvious editing keys for use with m and with the command line editor.

Use PRINT to print the display.

KRECEIVE [*filespec* [AS *filespec*]]

Receives one or more files from a Kermit host system, specified as:

KRECEIVE *filename* f
KRECEIVE *filename* AS *filename* f
KRECEIVE *wildcard* f
KRECEIVE *wildcard* AS *wildcard* f

See also KCOMMAND.

KSEND *filespec* [AS *unixfilespec*]

Sends one or more files to a Kermit host system, specified as:

KSEND *filename* f
KSEND *filename* AS *filename* f

```
KSEND wildcard  f
KSEND wildcard AS wildcard  f
```

See also KCOMMAND.

`LAUNCH program [parameters]` **(Windows)**

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.

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 ANZ_IP, ANZ_USERNAME, ANZ_PASSWORD, ANZ_HOSTNAME, and ANZ_LAST_RECD, as explained under ENVIRONMENT.

`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 I-systems, you may need to experiment with LINE DELAY and DELAY settings.

See also DELAY.

`LOCK [ON] | OFF`
`o`

LOCK ON is used for the normal I-system protocol, which 'locks' the keyboard when the host is not expecting entry. To temporarily unlock the keyboard, press `o`.

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.

`MENUBAR c1 r1 c2 r2 x3 x4 off len [cols]`

Superimposes a sliding menu bar system over the data presently on the screen, in one or more rectangles: (*x1*, *y1*) to (*x2*, *y2*) and (*x3*, *y1*) to (*x4*, *y2*). If a single-column menu is needed, just make *x3* equal *x1* and *x4* equal *x2*. 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 32, and the MENUDEMO program in Appendix D.

`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

`MODE-132` *`xx`* **`(DOS)`**

Sets the BIOS mode for 132 by 25 character display on your PC – see *Setting the Screen Size*, page 107. This allows Anzio to set itself into 132-column mode in response to a control sequence from the host, according to what terminal type is being emulated.

`MONITOR [ON] | OFF`

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. Use LOCK OFF with MONITOR ON.

MONITOR OFF returns to normal Anzio terminal emulation.

See also INTERPRET.

`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.

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

00	Completed successfully
01	File not found
02	Input file already open

`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 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 43
- KEEP
- PICK
- WRITE
- File transfer – see *File Transfer*, page 44
- Passthrough printing – see *Passthrough Printing*, page 37

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

`PAN` *n* | `LEFT` | `RIGHT` **(DOS)**

Used with the WIDTH command and REVIEW (see *Using Review Mode to See Previous Data*, page 25). Moves the 80-column screen window across the 132-column virtual window.

PAN *n* sets the starting column to *n*, showing columns *n* to *n*+79 (max 132).

PAN LEFT shows columns 1 to 80.
PAN RIGHT shows columns 53 to 132.

`PARITY EVEN | ODD | OFF`

Sets the parity for communication.

`PASSTHRU-CP` *name*

Sets the name of the code page used for printing with Print Wizard. Ignored if Print Wizard is off. *name* is one of the following:

NONE	Results in ISO or OEM depending on the terminal type.
ISO	Standard ANSI (ISO 8859-1) character set
ISO-8859-*x*	*x* is 2 through 9
CP-*xxx*	*xxx* is the number of a Windows or DOS *character codepage*, such as 150. Requires Windows support for the particular codepage.
T160E	Innovative's T160E.
CCCII	Chinese, Japanese, and Korean combined character set
UTF8	Byte-encoded Unicode characters

See also CHARSET.

PASSWORD *text*
PASSWORD/S

Sets or sends the host password. *text* is stored internally in an encoded format. 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 **(Windows)**

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 r1 c2 r2 type* [...]

Picks data off the screen and puts in the output file (see OPENO) for use by PC programs like LOTUS 123. 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 a format that Lotus can read.

Offsets can be cursor-relative – see BOX.

PITCH *n* | -1

(DOS) Sets the PC beep pitch (frequency). Try pitches in the range 200 to 2000. If you have several PCs in the same room, you can set each PC to beep in a different pitch.

(Windows) PITCH –1 causes the beep to come from the PC speaker, rather than an installed sound card.

PLAYSOUND *filename* **(Windows)**

Plays a WAV sound file.

PORT *n*

Specifies the hardware communication port number – see*Configuring the Serial Port*, page 8.

(DOS) Changing the PORT causes Anzio to set the appropriate interrupt request level (IRQ). If you are using a non-standard IRQ, you will need to use the IRQ command after changing the PORT.

PRINT [*c1 r1 c2 r2*]
PRINT/N [*c1 r1 c2 r2*]

Prints the lines visible on the screen, ignoring video attributes. Use the PRINTER command to identify the printer.

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.

To print the screen during a REVIEW, enter P.

To print displays generated by DIR, HELP, INTERPRET, KEYS, or TYPE, enter PRINT f .

`PRINTER` *name* **(All)**
`PRINTER WPRN` **(Windows)**

Identifies the printer used for the PRINT and PRINTER-SETUP commands. *name* is either a file or the PC printer's logical device name, usually PRN or LPT1. See also CHOOSEPRINTER.

If *name* is a file, that file is opened for output. If the file exists, it is overwritten.

PRINTER WPRN tells Anzio to use its Windows printer driver.

`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 f with a linefeed character (Ctrl-L). You may also enter printer-specific control codes to set compressed pitch, etc. Non-printable characters are displayed as ASCII hex. Press l to end keyboard input and exit this command.

text may also include control characters. To embed a control character in *text*, preface it with b P. For example, to send ^N Ctrl-N to your printer:

`PRINTER-SETUP` b P ^N b P b N f

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  q PRINT |
       q PRINTER-SETUP  b L | f
```

Note that a vertical bar | ends each command.

To embed any hex character in *text*, enter b 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:

b P 100D

`PRINTFILE` *filename*

Copies *filename* to the current PRINTER.

(Windows) 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.

`PRINTFONT [` *name*`] [`*size*`]` **(Windows)**

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 *height*x*width*.

`PRINTLEVEL WIZARD | HIGH | LOW | RAW` **(Windows)**

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 RAW indicates raw-level print.

PRINTLOW [ON] | OFF | RAW **(Windows)**

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 RAW indicates raw-level print.

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 **(Windows)**

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.

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

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 **(Network)**

Specifies reconnection behavior when the host system drops its end of the connection, for example, after you log off from a UNIX host.

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.

RENAME *oldname newname*

Renames a PC file from *oldname* to *newname*.

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 25.

RTS-MODE 0 | 1 | 2 **(DOS)**

Determines the behavior of the "request-to-send" line in the RS232 communication protocol.

Used only with special communications environments, including some types of half-duplex modems.

0 RTS stays high always (standard mode).
1 RTS stays high until a f is sent.
2 RTS goes on then off with each character.

RUN [*program* [*parameters*]] **(All)**
RUN/N [*program* [*parameters*]] **(All)**
RUN/S [*program* [*parameters*]] **(Windows32)**

Runs another Windows or DOS *program* on top of Anzio. That is, Anzio stays in memory while *program* runs on the PC. 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.

➢ Do not run any program that controls the serial port, or any terminate-and-stay-resident (TSR) program.

Before restoring the screen, Anzio prompts you to press any key. This is so you are able to see *program*'s output if necessary.

(Windows) 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.

(DOS) Anzio stays in memory and runs *program* only if there is enough memory.

RUN/N does not require a keystroke to restart Anzio after *program* ends.

RUN/S also sends *program*'s exit code to the host.

If you do not specify *program*, the RUN commands call the DOS command interpreter (C:\COMMAND.COM). You may 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 may be a 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  f
```

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 pseudo environment variables ANZ_IP, ANZ_USERNAME, ANZ_PASSWORD, ANZ_HOSTNAME, and ANZ_LAST_RECD, as explained under ENVIRONMENT.

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, or the default Anzio.KYS (for DOS) or AnzioWIN.KYS (for Windows).

SCREENMODE *option*

Sets the screen driver. SCREENMODE/S, below, reports screen mode parameters to the host.

(Windows) *option* is one of:

132	132 columns wide
80	80 columns wide
*C*x*R*	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.

(DOS) *option* is one of:

COLOR	Standard color 80x25
MONO	Standard monochrome 80x25
MONO/C or BW80	Monochrome screen on color driver
43X80 or 80X43	80 columns by 43 (50) lines for EGA (VGA)
132	132-column mode (must follow a MODE 132 command)
80	COLOR or MONO as appropriate
BIOS-*hh*	BIOS mode *hh* (in hex). The BIOS option can set your screen into a non-standard mode, if it is supported by the normal BIOS interrupt convention.

SCREENMODE/S

Reports several screen parameters to the host, formatted as one line:

TMMLLLCCCWWW *screenmode*

T (type)	C for color, M for mono, B for BW, 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.

`SCROLL [ON] | OFF`

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`

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` *`c1 r1 c2 r2`*

Sends a line of data from the screen to the host, from column *c1* to column *c2* on line *r1*. *r2* must equal *r1*; *r2* is a placeholder for consistency with other commands.

`SETCOLOR` *`n`* `| NORMAL`

Sets the working color to color index *n*, where *n* is either a decimal number or a hex value represented as *xx*H'. Does not affect the attribute/color table (see COLOR).

Use SETCOLOR NORMAL if you accidentally set a strange color.

`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 a A to interrupt a SLEEP and awaken Anzio.

`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.

`STAY` **(DOS)**
`STAY/G` **(DOS)**

Causes Anzio to go memory-resident (TSR).

STAY does not reserve any memory for graphics-mode programs. STAY/G reserves some memory for graphics-mode programs. See *Stay In Memory – STAY*, page 106.

`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.

`SYNC [ON] | OFF | FAST` **(DOS)**

Sets the horizontal retrace synchronization. Some video adapters (notably CGA) produce "snow" if screen updating is not synchronized with horizontal retrace.

Start with SYNC OFF. If you see snow during Anzio operation, then use either SYNC ON or SYNC FAST. SYNC FAST blanks the screen temporarily during scrolling, resulting in faster scrolling but more flicker.

`TAB` *`i j k ...`*

Sets tab stops at columns *i*, *j*, *k*..., up to 10 tab stops. Tab column numbers are entered in ascending order, separated by any non-numeric character. If less than 10 tab stops are set, the remaining tab stop are cleared.

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 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`

TAB ON uses the TAB CHARACTER for the Tab key's code.

TAB OFF uses the standard ASCII TAB character for the Tab key.

`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
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 UNIX systems
ANZIO	Anzio's own definition, based on VT220

➢ Be sure that Anzio's TERM setting agrees with the host computer's expected terminal type.

TERMNAME *name* **(Network)**

Sets a TERM variable that will be sent to the host as part of the telnet 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

Sends the PC clock time to the host in the format HH:MM:SS*f* .

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. Primarily used with the optional UFT file transfer program.

TITLE *string* **(Windows)**

Sets Anzio's window title.

TRACK-WINDOW [ON] | OFF **(Windows)**

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 [ON] | OFF
TRANSMIT  CRC
TRANSMIT  SINGLE
TRANSMIT  LONG
TRANSMIT  [LONG]  TRAILER  string
```

Controls file transmission according to the host protocol. See*File Transfer*, page 44.

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 CRC is the same as TRANSMIT ON, but uses a CRC protocol for error-checking ("cyclic redundancy check"). Must be used with special host software.

TRANSMIT SINGLE is the same as TRANSMIT ON, but sends one line at a time to the host, then waits for the host to respond. Use the XN command to send each successive line.

TRANSMIT LONG is the same as TRANSMIT ON, but uses a special protocol to break long records into 80-byte or smaller pieces. The host must have software (such as the supplied RECV-PC.CBL) that uses this protocol to reassemble the records.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 "\".

(UNIX) To specify a c character as the trailer, enter *string* as either b P c or a 127.

TTY

Tells Anzio that you are working with a bulletin board system or some other unspecified host. See ITX.

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.

UNIX

Tells Anzio that the host system is running UNIX. See ITX.

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 – Defaults File Name or Telnet Host Name*, page 87).

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 a A.

`WIDTH 132 | 80` **(DOS)**

Sets the "virtual screen width". WIDTH 132 on an 80-character physical screen gives you an 80-column-wide window onto the 132-column virtual screen. Use the PAN command to move the window right or left.

WIDTH 132 can be useful for viewing spool files.

See also *Setting the Screen Size*, page 107.

`WINDOW` *`c1 r1 c2 r2`* [FILL] [BOX | DOUBLE]

Opens a "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.

See the MENUDEMO program in Appendix D.

`WINDOWCLOSE`

Following a WINDOW command, closes the window and restores the screen.

`WINPRINT` *`filename`* **(Windows)**

Asks Windows to print *filename* using the program associated with its file extension, in the same way as File Manager or Explorer.

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 ANZ_IP, ANZ_USERNAME, ANZ_PASSWORD, ANZ_HOSTNAME, and ANZ_LAST_RECD, as explained under ENVIRONMENT.

`WINSTART` *`filename`* **(Windows)**

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 File Manager or Explorer.

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 ANZ_IP, ANZ_USERNAME, ANZ_PASSWORD, ANZ_HOSTNAME, and ANZ_LAST_RECD, 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 b P, for example b P ^ .

To embed any ASCII character in *text*, enter b P followed by hex 10 and the two-character ASCII hex value of the character. The hex 10 represents a second b P, used internally for quoting ASCII characters. For example, to embed a Return character (hex 0D), enter:

b P `100D`

`XN`

Transmits the next line of a file to the host. Used with TRANSMIT SINGLE.

`ZRECEIVE [` *`filename`* `]`

Receives a file using the Zmodem protocol. If *filename* is not specified, the received file is stored under the host's original filename.

`ZSEND [-a]` *`filename`*

Sends a file to the host using the Zmodem protocol. Use ZSEND-a with text files to indicate that the host should remove the carriage return at the end of each line.

10 More On Starting Anzio

10.1 Paths And Subdirectories (DOS Only)

When you start Anzio for DOS, the operating system (DOS) must be able to find the program (Anzio.exe). There are three ways to do this:

1. You can change to Anzio's home directory, then start the Anzio program:

   ```
   C:  f
   cd \AnzioDIR   f
   Anzio  f
   ```

2. You can specify Anzio with a pathname:

   ```
   C:\Anzio   f
   \AnzioDIR\Anzio    f
   ```

3. You can add Anzio's home directory to your PATH. DOS uses the directory names in its PATH variable to search for executables (like Anzio.exe).

   ```
   SET PATH = …    f
   Anzio  f
   ```

10.2 Command Line Parameters & Default Files

You can call the Anzio program with one or more command line parameters, separated by spaces:

`Anzio` *`param1 /param2 /param3 name …`*

Command line parameters can run a defined key, specify a defaults 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.

(Windows) Command line parameters are coded into the program icon.

(DOS) Command line parameters are typed after the program name, on the same line.

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 defaults file (Anzio.def or AnzioWin.def). This file contains all user settings from the previous Anzio session. The defaults file can also specify a key definition file to be loaded. Anzio looks for a defaults 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 defaults 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 defaults file name to be loaded, or a telnet URL with a username and host name. See *Names – Defaults File Name or Telnet Host Name*, page 87.
- 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 88.

10.2.1 'Anzio NONE' For Clean Startup

Use the command line parameter NONE to start Anzio without loading the defaults file or any key file:

```
C:> Anzio NONE  f
```

Anzio will start up in a virgin state, just like after the initial installation. You will need to specify a terminal type and the communication parameters – see *Setting Default Anzio Communication Parameters*, page 10.

10.2.2 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  f
```

will start Anzio and load the standard defaults file. In the defaults 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 88.

10.2.3 Names – Defaults 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 defaults file name, or a telnet address. For example:

```
MyAnzio.def
telnet://my.host.com
user123:password777@my.host.com:80
```

A complete name parameter can contain a telnet address, including a username and password, and a host name and optional port:

```
[telnet:// ][username[:password]@]name[:port]
```

That is, all parts are optional except *name*. Anzio reads a name parameter as follows:

1. Anzio strips off an initial "telnet:" or "telnet://", and a trailing slash, if any.
2. 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.
3. Anzio first tries to use *name* as a defaults file name. Note that only 32-bit Anzio for Windows supports long file names.

   If the file exists, in the current directory, in the parent directory, or in Anzio's home directory, that file is used instead of Anzio.def or AnzioWin.def.

4. 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 (Anzio.def or AnzioWin.def), but then connects to the given host name rather than the host named in the defaults file.

- For example, if you need to connect to three host systems, you can create three Windows icons, each naming a different defaults 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 defaults file having the same name as the host. Anzio will find and read that defaults file, and so will not read the parameter as a host name. Inside the defaults file, specify the desired host name, and Anzio will connect to that host.

10.2.4 /C Choose Defaults File (Windows)

The parameter

```
/C
```

causes Anzio for Windows to prompt the user for the name of a defaults file to use.

10.2.5 /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 26, 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:

```
qDIAL qCALL 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.

10.2.6 /H Hostname (Network)

Use /H to specify a network host system to connect to, overriding the host name in the defaults 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
```

10.2.7 /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
```

10.2.8 /M Memory (DOS)

Anzio for DOS can be limited to a certain amount of memory:

`/M:`*nnn*

where *nnn* is a number of memory bytes, between 16K and 64K. This amount of memory is set aside to be shared by both REVIEW and defined keys. See *Memory Usage*, page 105.

10.2.9 /T Communication Type (Windows)

Anzio for Windows uses /T to specify which type of communication to use:

```
/Ts   serial
/Tt   TCP/IP
/Tn   Novell WLIBSOCK
/Tp   PicLan
```

Part IV Technical Reference Guide

11 Transmit Priorities

At any one time, Anzio may have output to the host pending from several sources: keyboard or function keys, file transmission, host request for status, etc. If the keyboard is unlocked, and there is something to be sent, Anzio uses the following priority:

1. **Immediate function keys**
   BREAK u , UNLOCK o , PANIC t , and FUNCTION q are all processed immediately, regardless of keyboard lock status.

2. **Retransmit**
   If the host has requested a RETRANSMIT, the previous line is resent.

3. **File transmission**
   If a TRANSMIT is active, the next line of the file is sent.

4. **Anzio calculator result**
   If the operator has exited from CALC with l , the result is sent.

5. **Host status return**
   If the host has issued a status return command, such as OPENO/S, the status is sent.

6. **Response from file reception**
   If a RECEIVE is active, the handshaking that is part of file reception is sent.

7. **Defined key**
   If a defined key is in progress, Anzio processes the next keystroke.

8. **Buffered keystrokes**
   Finally, any characters waiting in the 128-character keyboard type-ahead buffer are sent.

12 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 44, for the operator view of file transfer.

Several source programs that demonstrate the host side of file transfer are included with Anzio – see Appendix G, *Distribution Information*, page 117.

12.1 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.

12.2 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 G, *Distribution Information*, page 117.

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:

*nn*Y*ddddddddd*...

where *nn* is two ASCII characters (01-77) giving the number of data bytes ddddd. The third character indicates whether this is the last piece of the record, an ASCII "Y" or "N".

12.3 TRANSMIT CRC

TRANSMIT CRC uses a special protocol for error checking, and is intended only for use with our Universal File Transfer (UFT) utility program, from Rasmussen Software, Inc.

See *Universal File Transfer Utility Program*, page 52.

12.4 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 43.

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.

12.5 RECEIVE WITH CRC

RECEIVE CRC receives files using a special protocol for error checking, and is intended only for use with our Universal File Transfer (UFT) utility program, from Rasmussen Software, Inc.

See *Universal File Transfer Utility Program*, page 52.

12.6 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.

(RM/COS) Anzio will send a f to an RM/COS host after writing each line to the PC file, since Anzio assumes that this CAPTURE is being used for file transfer.

12.7 CAPTURE LONG

CAPTURE LONG captures long records from the host, using the piece-wise protocol described with TRANSMIT LONG, above.

12.8 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 t to exit pass-through print mode and return to normal display mode.

(RM/COS or UNIX) Anzio will use XON/XOFF handshaking with the host when the printer is slower than the connection.

13 Sending Anzio Commands From The Host

The host can send Anzio commands by using special control codes. Any q 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 G, *Distribution Information*, page 117.

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 executes the command:

`OPENO DOWNFILE` f

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 q command itself, for example, to define a key from the host. q 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 60, describes all commands having the /S option, including:

- CLOSEI/S
- CLOSEO/S
- COPY/S
- DELAY/S
- DELETE/S
- DIR/S
- ENV/S
- FIND/S
- FINDNEXT/S
- HOSTNAME/S
- MKDIR/S
- OPENI/S
- OPENO/S

- SCREENMODE/S
- VERSION/S

(UNIX) You can send Anzio commands with the UNIX shell command “echo”:

```
echo "\0034 OPENO DOWNFILE \0035"     f
```

Where “\0034” indicates octal 34 (hex 1C), and “\0035” is octal 23 (hex 1D). 7. 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 with a HEXPATCH command.

14 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 86, 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.exe executable file. You can also change or remove accelerator keys, such as a X to exit.
- Anzio provides access to some internal settings with the HEXPATCH command.

14.1 HEXPATCH Commands

Some of the newer or more obscure options in Anzio do not have menu items or Anzio commands to manipulate them. The HEXPATCH command directly changes the parameters in Anzio's working memory. Then, if you save your parameters when exiting, the change will still be in effect the next time you start Anzio.

➢ Use the HEXPATCH command with caution. A single typo can kill Anzio.

The format is:

q HEXPATCH *address data* f

where *address* is a hex number, and *data* is the hex representation of one or more bytes. For example:

q HEXPATCH D67 00 f

The following areas are affected by HEXPATCH commands:

Euro-dollar Set the character code used in your screen font for the Euro-dollar with HEXPATCH D67*xx*. Use 00 to disable Euro-dollar translation. If enabled, Anzio converts an incoming *xx* character to the Euro-dollar display character, and vice versa when sending a Euro-dollar character. Euro-dollar translation is independent of the CHARSET used. This setting will also affect passthrough printing in some printing modes.

mouse Change right or left click and double-click actions. The default is 6CA 01 (left double-click sends word to host with f). The HEXPATCH addresses are:

6CA	left double-click
6CD	left single-click
6D0	right double-click
6D3	right single-click

For each address, you can set the action performed. The HEXPATCH data is:

00	no action
01	send the word under the cursor, followed by f
02	send the word without a f
03	execute a macro (contact us for more information)
04	adjust the palette
05	send the string under the cursor enclosed by brackets [ ], without any leading spaces
06	start the URL under the cursor (call a Web browser)

shape of mouse selection

When you drag the mouse to select part of the screen, Anzio can highlight in two different modes or shapes. Do HEXPATCH D6A xx, where xx is

00 column block, that is, a rectangular area. In this mode when you copy to the clipboard Anzio will copy as a bitmap as well as text.
01 stream block, as in mode word processing programs.

host commands

You can allow or disallow processing of Anzio commands issued by the host. Use a HEXPATCH address of 7D6 for hex 11/13 (old style) or 7D7 for hex 1C/1D (current style). Set data to:

00 disallow
01 allow

display GIF files

AnzioWin can be set to always display a GIF or BMP file on the screen after it has been downloaded. Use HEXPATCH 9DB 01 to enable this option, or 00 for the default (don't display).

multiple instances

By default, each time you start Anzio for Windows, you get a new separate *instance* of the program. In certain situations, you may want to allow only one instance of Anzio to run at a time. The instance options are set at HEXPATCH address 7D8:

00 allow multiple instances
01 keep existing instance, close new connection
02 stop existing instance, and start a new Anzio instance

capture control characters

Governs whether capture to file and capture to printer will include control characters. The options are set at HEXPATCH address D4A:

00 do not capture controls
01 capture controls

prompt to save

When Anzio quits, if any parameters have changed, it will generally ask you if you want to save them. Set HEXPATCH address D66 to:

00 ask to save, if parameters have changed
01 always save if changed, without asking
02 don't ask, don't save

disable accelerators

To tell Anzio NOT to process alt-key accelerators for Windows menu items, so you can define them yourself, do HEXPATCH 695 00. Re-enable them with HEXPATCH 695 01.

ctrl-R to print In some situations, receipt of a ctrl-R (DC2) initiates "aux printing". To allow this, do HEXPATCH AE0 01. To disallow, do HEXPATCH AE0 00.

ctrl-X to print When emulating a Wyse terminal, you can govern whether receipt of a ctrl-X initiates passthrough print. To allow this, do HEXPATCH CE5 01. To disallow, do HEXPATCH CE5 00.

windows style You can change the appearance of Anzio's window by changing its "style" bits. This will take effect the next time Anzio is started. Use a HEXPATCH address of D4D, and data that consists of the hex sum of the following:

0100 miximize box
0200 minimize box
0400 thick frame
0800 system menu (also governs 'close' box)
4000 dialog-style frame (NO menu)
8000 border
0010 popup (no title bar)

variable fonts Ordinarily, Anzio lets you select only mono-spaced fonts for View/Screen Font. If you need to be able to choose variable-spaced fonts, do HEXPATCH D5C 01 (Anzio will still display them evenly spaced). To reset to mono-spaced only, do HEXPATCH D5C 00.

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.

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.

> ➢ Since Anzio uses a direct interface to the communications port, Anzio may detect and report an error when other communication software does not.

PARITY A parity error has occurred. Verify that your PARITY and BAUD settings match the host's.

Some parity errors are caused by telephone line noise over a modem. This may be a problem during file transfer, as some data may be lost. If necessary, use TRANSFER CRC to add data checking. You may also be able to install an error-correcting modem in your PC.

FRAMING 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.

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.

OVERRUN Another program interrupted Anzio before Anzio could process an incoming character. When a character arrives at the PC's serial port, it generates an *interrupt*. 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.

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.

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 Anzio for DOS

This appendix describes features specific to Anzio for DOS

- *Memory Usage*
- *Running Another DOS Program*
- *Anzio Small for DOS*
- *Setting the Screen Size*

B.1 Memory Usage

Anzio is a complex, and therefore large, program, requiring approximately 165K to 222K of PC system memory.

You may not be able to run Anzio and another memory-intensive application at the same time:

- You may elect to minimize REVIEW memory (see below).
- You may be able to use our minimal version of Anzio (*Anzio Small*), as described in the next section.

➢ Anzio is too large to be loaded into high memory or extended memory.

Anzio's program itself requires some core memory. Anzio also allocates a *memory buffer* for use with REVIEW memory, defined key strings, and file transfer data.

To set the size of the memory buffer, use the /M parameter:

```
Anzio /M:nnn f
```

where *nnn* is a number of bytes between 16384 (16K) and 65535 (64K). If *nnn* is outside this range, Anzio will round it to the closer number. To set minimum memory usage:

```
Anzio /M:1 f
```

See */M Memory (DOS)*, page 89.

B.2 Running Another DOS Program

Anzio for DOS can call another DOS program to run either sequentially or simultaneously.

- The RUN command pauses Anzio, runs a DOS program to completion, and then reawakens Anzio.
- The STAY command puts Anzio in memory and starts DOS. You can call up Anzio at any time with a keystroke (*hot key*), switching between Anzio and another program. This is also called *terminate and stay resident* (TSR).

You can minimize Anzio's memory buffer for use with RUN or STAY, as described above. If Anzio has access to more memory than it needs, it will release that memory when it does a RUN or STAY.

B.2.1 Run a DOS Program Over Anzio – RUN

The RUN command tells Anzio to pause itself, then run some other DOS program and wait for that program to finish. For example:

```
RUN WP.EXE f
```

The WP program starts up, and after you are through, exits. The previous Anzio screen reappears.

- If you run a DOS program that accesses the serial port, you may lose your host connection.
- The RUN command can be used in a defined key, or as a command from the host computer, to execute a PC program.

The RUN command without a program name just calls the DOS command interpreter. You can then enter DOS commands and run any series of programs. To return to Anzio, use the DOS exit command:

```
EXIT f
```

B.2.2 Stay In Memory – STAY

The STAY command puts Anzio into a "memory-resident" mode where it can "pop up" over another program:

```
STAY f
```

Anzio will go to sleep, and put you at the DOS command level. You can then run some other program, such as a word-processing program WP:

```
WP f
```

While you are in the word processing program, you can enter Anzio's hot key code to pop up Anzio:

a l

These keystrokes can be changed with the HOTKEY command. When Anzio pops up, it saves the current screen and restores its own screen. Use another STAY command to return to the other application.

- Anzio can pop up on most video boards, over most programs running in *text* mode.
- To pop up over a program in *graphics* mode (WP in preview mode, WORD in a graphics mode, 123 release 3, etc.), you must use STAY/G rather than STAY. STAY/G saves enough memory for graphics. STAY/G is only required for the first time; after that, STAY has enough memory.

Anzio can be unloaded from memory at most times when nothing else is loaded on top of it, with a X, or the END or STOP command.

The sample key files define a l as STAY/G, so that you can switch in and out of Anzio using the same key combination. See *Sample Defined Keys*, page 119.

a l only works when the keyboard is unlocked (no LOCK on the status line). Press o to unlock the keyboard if necessary.

B.3 Anzio Small for DOS

There is a special "small" version of Anzio for DOS, included as AnzioS.exe This has Anzio's basic communication abilities, but without support for file transfer, REVIEW, and other advanced functions.

9. Use the standard version (Anzio.exe) to get everything set up and running.
10. Exit from Anzio, saving your defaults.
11. Run AnzioS.

Note that AnzioS automatically does the equivalent of "/M:1", as explained in */M Memory (DOS)*, page 89.

B.4 Setting the Screen Size

Anzio for DOS is very flexible in terms of the number of characters on a line and lines on a screen. If your PC's video driver board provides more than the standard 80 by 25 character layout, Anzio can work with it. Or, if the PC hardware supports only 80 columns, Anzio can keep a 132-column "virtual screen" in memory and show you any 80 out of the 132.

B.4.1 DOS Screen Hardware

Many video drivers support extended character-mode resolutions, such as 80 by 43, 80 by 50, 132 by 25, and 132 by 43 or 50. The most popular of these is 132 by 25.

Each video board manufacturer uses different codes to set these extended modes. You have to tell Anzio how to set your screen to 132 by 25, as described below.

However, once a video mode is set, Anzio can read the current number of columns and lines and reset itself to match.

There are three ways to change the video screen mode:

- Set it before you run Anzio. Anzio will then initialize itself according to that setting.
- Use Anzio's RUN (or RUN/N) command to run a mode-setting program that came with your video board. After any RUN, Anzio checks the screen parameters and readjusts if necessary.
- You may be able to use Anzio's SCREENMODE command to set a hardware code for the desired mode.

B.4.2 Setting the Screen Mode

Most video boards have a *BIOS mode number* for 132 by 25 mode. This number should be found in the documentation for your video driver board. For example, if the code is a hex 55, the SCREENMODE command is:

q `SCREENMODE BIOS-55` f

To switch back to normal color mode (80 by 25), enter:

q `SCREENMODE COLOR` f

You can also set the BIOS mode number for a 132 by 25 display directly, with the MODE-132 command. For example, if the mode number is hex 55:

q `MODE-132 55` f

Once Anzio knows the mode number, it will switch to 132-column mode in response to control codes from the host. You can change the mode manually with the SCREENMODE command:

q `SCREENMODE 132` f

➢ The **host** system software is probably designed for use only with one standard display size. For example, the host software may assume that moving the cursor past line 24 position 80 will scroll the screen. If you have set the SCREENMODE to 132, the cursor will actually move to line 24 position 81.

B.4.3 132-Column Virtual Screen

Anzio for DOS is capable of displaying a 132-column "virtual screen" on an 80-column display. The PC's video driver card determines the number of characters visible on the screen at one time. This is the "physical" width. Even if the physical width is 80, you can still tell Anzio to use a 132-column "logical" width. Anzio receives and stores 132 columns of data, allowing you to "pan" (scroll horizontally) across the 132 logical columns to see an 80-column-wide slice.

The virtual screen width is set with the WIDTH command. The virtual screen WIDTH must be greater than or equal to the physical width. That is, if your video hardware is showing 132 characters, WIDTH must be 132.

The following example assumes a physical width of 80 columns with a logical WIDTH of 132.

When the logical width is greater than the physical width, your screen shows a "window" of the full virtual screen. You can move the displayed window in three ways:

- Use the PAN command, for example, PAN LEFT moves the window one column to the left. PAN RIGHT moves the window one column to the right, and PAN ***n*** moves the window to start at column number ***n***.
- Use REVIEW mode, with the scroll (arrow) keys, to pan in all four directions. Press g to move to the upper left corner of the virtual screen, and d to move to the lower right corner of the virtual screen.
- PICK from the virtual screen by dragging the cursor across the area you wish to see.

Anzio includes a demonstration program to show a virtual screen on an NCR I-system. This is a COBOL source program (SPOOLCRT.CBL) that browses through spool files. To use the demo program:

- Move the SPOOLCRT.CBL file to the host system and compile it.
- Assign A to the spool file and execute the program. It will tell you its commands.

Appendix C 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*
- *I-Series Hosts*
- *RM/COS Hosts*
- *Unknown Host Type – TTY Mode*

C.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 "set" command. To change UNIX's setting, tell UNIX:

```
TERM=VT220; export TERM    f
```

for instance. Note that some application software on UNIX may assume you are using a certain TERM type, regardless of the current setting.

If you use RM/COBOL under UNIX, and configure Anzio (and UNIX) for the NCR 7900, you will find that they don't get along too well, because of the 7900's use of propagating attributes (attributes take a space on the screen). You can either modify the terminfo to disable attributes, or use a different terminal type. 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'   f
```

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   f
```

Finally, set Anzio for UNIX, LOCK OFF, FULL DUP, BACKSPACE 8, and probably TAB OFF.

C.2 I-Series Hosts

When working with one of these I-systems, be sure your parameters are set as follows:

LOCK — LOCK should be ON at most times with these systems, in order for keyboard buffering to work. Exceptions are 1) when talking to a local modem, 2) during bootup of some systems, and 3) newer releases of SCLEDIT and VIEW under ITX, which do not beep.

TERM — The I-systems were originally designed for the 7900 terminal, so you'll probably want to set TERM to N7900. This is changing even as we write this, however, with later releases of ITX including support for other terminal types. Just be sure that TERM in Anzio agrees with your SYSGEN setting.

C.2.1 ITX Host Configuration

On an ITX system, each terminal's parameters are set in the SYSGEN entry for that terminal line.

C.3 RM/COS Hosts

Following are some notes on running under RM/COS

Cabling — We have received reports of problems with RM/COS locking up the terminal line, effective RM/COS release 2.7. We have also heard that an alternative wiring scheme for "intelligent terminals", suggested by NCR support, eliminates the problem. This new cabling scheme is embodied in NCR cables with part numbers 1308-C045 (for 9-pin) and 1308-C046 (for 15-pin). Contact us or your NCR FE for more information.

Terminal Type and Passthrough Printing
Anzio will work if RM/COS thinks it is an N7900, N7901, or ADDSVPT. Only the ADDS Viewpoint supports pass-through printing, however, so we suggest using that setting. Pass-through printing also requires that you define (in the SYSDEFIL) a printer slaved to your CRT.

DUPLEX — Always use FULL DUP.

LOCK — You should generally work in LOCK OFF mode, except when doing file transfer. This is because the operating system does not ordinarily send a BELL code with each prompt.

TAB — Some programs under RM/COS are designed to make use of the TAB key. In order to send the tab code (hex 09) to the host, instead of translating it into spaces, set TAB OFF.

Defined Keys
We suggest you set up a function key for "acknowledge" (control-G A), and one for "interrupt" (control-X). See the sample key file (RMCOS.KYS), as described in *RMCOS.KYS Key File*, page 121.

File Transfer — As mentioned, file transfer works differently on RM/COS.

C.4 Unknown Host Type – TTY Mode

Anzio has a "dumb terminal" mode, called TTY. Use this mode when you are using Anzio to talk to a bulletin board system or some other foreign system. In TTY mode, Anzio does not respond to most of the control codes listed in this manual.

Most hosts will also support a VT100 terminal.

Appendix D Additional Programs

The Anzio installation disk includes two demonstration programs

MENUDEMO.CBL A COBOL program for use on an I-series host. Demonstrates how a host can control various Anzio features, including color, pop-up menus, and running another PC program.

SPOOLCRT.CBL A COBOL program for use on an I-series host. Displays spool files on an Anzio or Wyse terminal.

Each of these programs is described in the following sections.

D.1 MenuDemo Utility (I-series Host)

The MENUDEMO program is a COBOL source program for use with I-systems (IMOS III, IMOS V, IRX, and ITX). It demonstrates some of the newer features of Anzio, under control of the host program.

Transfer MENUDEMO.CBL up to the host, using the "simple upload" instructions for your particular system. Then simply compile it and run it.

The MENUDEMO program demonstrates the following features:

- Commands sent from the host computer
- Control of color
- MENUBAR
- WINDOW
- FILL
- RUN - dropping out to the DOS level
- RUN program - host initiates a PC program

D.2 SpoolCRT Utility (I-series Host)

SPOOLCRT is a program for IMOS III, IMOS V, IRX, and ITX to read and display a manual spool file on the screen. It allows you to move forward and backward in the file by lines, screens, or pages.

SPOOLCRT will work with either an 80-column or a 132-column display, and as such is a good way of demonstrating Anzio's 132-column virtual screen capability. Of course SPOOLCRT has to know which type of screen it has. You can tell it you have a 132-column screen either by setting switch 1 on before executing it, or by giving it the command "132W" once it is running.

The program does not restrict itself to Anzio as a CRT. In fact, it contains the necessary code to switch a WYSE terminal into and out of 132-column mode.

SPOOLCRT is distributed on floppy, and must be uploaded and compiled before execution – see *Upload and Compile Example*, page 46.

As a sample session, enter the following commands to the host:

```
SET SWITCH 1 ON  f
AS A MYSPOOL(3)  f
EX SPOOLCRT(5)   f
```

Once loaded, SPOOLCRT will give you a list of its command options.

Appendix E 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.

The README.TXT 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.

If you have problems with a new release, start Anzio with command line parameter NONE:

```
ANZIO NONE  f
```

Anzio will start up in its “virgin” state, without reading either a defaults file or a key definition file.

Appendix F 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. Set LOCK mode off with Communicate/Lock Off, or enter

   LOCK OFF f
3. If the status line still shows LOCK, press a U to unlock the keyboard.
4. Try setting Anzio for different BAUD rates.
5. Try setting Anzio for different 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 If possible, make the port and IRQ match one of the standards listed in *Configuring the Serial Port*, page 8. Otherwise, you may need to use Anzio's IRQ command to set it for a non-standard situation.
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 G Distribution Information

Anzio is distributed on one or more standard PC disk(s). Each distribution disk is your master copy, and should be stored in a safe place. The distribution contains Anzio executables and support files, including these types of files:

*.CAP	AcuCobol *termcap* file containing terminal settings.
*.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 *Defined Keys and Macros*, page 26.
*.TIC	*Terminfo compiler* files for UNIX host-side terminal settings.
*.TTF	*TrueType font* files for multi-language screen display.
*.UNI	Unicode mappings for various character sets – see *Using Anzio's Language Support*, page 23.
NCR3000/	Subdirectory containing pre-compiled SEND-PC and RECV-PC files for an NCR 3000.
NCRTOWER/	Subdirectory containing pre-compiled SEND-PC and RECV-PC files for an NCR Tower.
RECV-*	Host programs to receive PC files on the host (upload).
SEND-*	Host programs to send host files to your PC (download).

- The following list describes all distribution files in alphabetical order, except the self-explanatory UNI (Unicode mapping) and KYS (key definition) files.
- See also the installation file README.TXT.

Anzio.exe	Anzio for DOS program file. Might instead be AnzioD.exe.
ANZIO.HLP	The help file used by Anzio for DOS.
ANZIO.TIC	A UNIX terminfo file defining host terminal type TERM ANZIO – see *UNIX Host Configuration*, page 14. Used with ANZIO-TIC.KYS.
Anzio32.exe	AnzioWin program file.
Anzio32r.exe	Anzio Lite program file.
ANZIO-M.TIC	A UNIX terminfo file defining host terminal type TERM ANZIO-M for a monochrome PC – see *UNIX Host Configuration*, page 14. Used with ANZIO-TIC.KYS
AnzioS.exe	Anzio Small (for DOS), designed for situations where memory is limited. It does not have many of the advanced features of Anzio.
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.
DOWNLOAD	A UNIX shell script for downloading files to Anzio.
INSTALL.BAT	Installation batch file for Anzio for DOS.
KERMIT.DOC	A file that explains the keys used in KERMIT.KYS
KERMIT.KYS	A file of function key definitions to work with Word Perfect on a UNIX system, when WPTERM is set for KERMIT.
PRINTWIZ.TXT	Additional information about the Anzio Print Wizard – see *Anzio Print Wizard*, page 38.
README.TXT	A file of last-minute notes, etc. To see them, enter: `TYPE README.TXT` f
RECV-PC.C	A UNIX source program that allows upload of longer records into UNIX.
RECV-PC.RM	A COBOL source program for file transfer of any file larger than 80-byte variable files to an RMCOS machine from a PC. Currently it is set for 510-byte files.
RECV-SPL.CBL	A COBOL source program for file transfer of a PC print file to an IMOS III, IMOS V, IRX or ITX spool file or printer.
RMCOS.KYS	A sample key file for RM/COS hosts.
RSIMARC.TTF	*TrueType font* file for library-standard multi-language screen display, with additional characters defined by Rasmussen Software, Inc. (RSI) – see *Using Anzio's Language Support*, page 23.
SAMPLE.KYS	A sample key file for NCR I-systems.
SEND-PC.C	A C program used to download UNIX text files to a PC – see *UNIX Download Utility (Send-PC.C)*, page 50.
SEND-RM	A source program for file transfer of 80-byte variable files using Anzio's CAPTURE procedure under the RM/COS operating system.
SEND-L.RM	A source program for file transfer of 510-byte variable files using Anzio's CAPTURE LONG procedure under the RM/COS operating system.
SET-EGA.EXE	A PC program that helps you configure an EGA or VGA video adapter to allow underlining, etc.
SET-EGA.DOC	Documentation for SET-EGA.EXE.
SPOOLCRT.CBL	A COBOL source program for NCR I-series, to display spool files on a CRT or PC.
VIDMODES.EXE	A PC executable that looks for available video modes.
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
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

WYSE60.KYS Sample function keys for WYSE60 emulation.

WYSE60.DOC A file that explains the keys used in WYSE60.KYS

NCR3000/RECV-PC
The RECV-PC.C program, precompiled for a UNIX V.4 host. See `README.TXT` for installation information.

NCR3000/RECV-PC.ECH
A UNIX script file that creates an NCR3000/RECV-PC program (above). See `README.TXT` for installation information.

NCR3000/SEND-PC
The SEND-PC.C program, precompiled for a UNIX V.4 host. See `README.TXT` for installation information.

NCR3000/SEND-PC.ECH
A UNIX script file that creates an NCR3000/SEND-PC program (above). See `README.TXT` for installation information.

NCRTOWER/RECV-PC
The RECV-PC.C program, precompiled for a UNIX V.4 host. See `README.TXT` for installation information.

NCRTOWER/RECV-PC.ECH
A UNIX script file that creates an NCRTOWER/RECV-PC program (above). See `README.TXT` for installation information.

NCRTOWER/SEND-PC
The SEND-PC.C program, precompiled for a NCR Tower host. See `README.TXT` for installation information.

NCRTOWER/SEND-PC.ECH
A UNIX script file that creates an NCRTOWER/SEND-PC program (above). See `README.TXT` for installation information.

Appendix H Sample Defined Keys

Defined Keys and Macros, page 26, describes how to assign macros to keyboard keys. The Anzio installation includes two sample key-definition files:

- SAMPLE.KYS for NCR I-series operating systems
- RMCOS.KYS for RM/COS

> ➢ To call a defined special key, such as a function key or a control character, just press the key(s). To call a defined regular keyboard key, press l and then the key.

H.1 Defined Keys for NCR I-Series

H.1.1 SAMPLE.KYS Key File

The sample key file SAMPLE.KYS defines some macros for I-series operating systems, especially ITX:

1	"IN*D" to begin INSERT in $EDIT.
B	**(ITX)** Special BREAK/SUSPEND key. Captures the entire screen as a WINDOW, then sends a BREAK and an S (for SUSPEND) to save the old screen. See also "b".
b	Undoes the B key, above. Sends a RET to exit the current process, then restores the screen saved by the B key.
C	Displays the COLOR screen.
T	Sends the current TIME.
H	Hangs up the modem.
L	**(ITX)** Logs on to an ITX system by sending a series of BREAKs followed by an "R". The ITX system should then be at command level.
c	**(IRX, some ITX)** Logs on as "L" above, using *ctrl-C* characters rather than BREAK characters.
R	Shortcut for RUN/N command; takes you to the DOS command prompt.
t	"Transmit" – prompts for a file name, then opens and transmits the file, with trailer END$.
u	Does a UNIX simple file upload, with prompts.
r	Displays the command line CALCulator.
s , {	Shortcut for REVIEW mode.
g	In $EDIT, start line editing – sends q CH | {w}m.
b z	PAN LEFT.
b x	PAN RIGHT.

a l Causes Anzio to go memory-resident (STAY/G command).

H.2 Defined Keys for RM/COS

H.2.1 RMCOS.KYS Key File

The sample key file RMCOS.KYS defines some macros for the RM/COSoperating systems:

C Displays the COLOR screen.

H Hangs up the modem.

R Shortcut for RUN/N command – takes you to the DOS command prompt.

r Displays the command line CALCulator.

s , { Shortcut for REVIEW mode.

a l Causes Anzio to go memory-resident (STAY/G command).

D *ctrl-G*Z for DELETE LINE

I *ctrl-G*N for INSERT LINE

j TAB *ctrl-G*B for BACKTAB.

g *ctrl-G*C for ERASE FIELD

{ *ctrl-G*A for ACKNOWLEDGE

d *ctrl-G*R for ERASE RIGHT

} *ctrl-G*Q for COMMAND

h *ctrl-G*I for INSERT CHARACTER

j l through j u
*ctrl-G*1 through *ctrl-G*0 for FUNCTION 1 through FUNCTION 10.

TAB (if TAB OFF)

c DELETE CHAR

arrow keys

Appendix I Anzio On a Network

The question often comes to us, “Can Anzio work on a network?” The short answer is “yes”. The other answers are:

- You can use Anzio on a PC that is connected to a network, bypassing the network to communicate with the host through a serial connection.
- You can access files on a network server through a serial connection to the server.
- You can talk to the host computer via the network, using AnzioWin or Anzio Lite (Windows), or AnzioNet or Anzio14 (DOS).
- You can get through a SOCKS firewall, by entering the SOCKS host name and optionally a port number with Communicate/Network/SOCKS Server.
- Anzio does not provide the underlying network software for your PC. The networking software must come from other sources:
  - **(Windows)** For AnzioWin or Anzio Lite to communicate via the network, you need one of: WINSOCK.DLL for TCP/IP, or support software for PicLan, or Novell’s WLIBSOCK.
  - **(DOS)** For AnzioNet or Anzio14, you need DOS-level network software.
- You can put the Anzio program itself on a network (but please don’t violate licensing restrictions

Index