UNCLASSIFIED/UNLIMITED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A199 551

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED/UNLIMITED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) DTIC/TR-87/18	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Defense Technical Information Center 6b. OFFICE SYMBOL (If applicable) DTIC-EA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Cameron Station Alexandria, VA 22304-6145	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
File Conversion and Transfer from a Lanier No Problem Word Processor to a Defense Logistics Agency (DLA) Distributed Minicomputer System (DMINS)

12. PERSONAL AUTHOR(S)
Thornett, Richard Geoffrey

13a. TYPE OF REPORT	13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
		87 Sep	67

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	DIF, DOCUMENT FORMAT, DOCUMENT INTERCHANGE FORMAT, FILE CONVERSION, FILE TRANSFER, NAVY DIF, OFFICE AUTOMATION WORD PROCESSING

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This document is about conversion and transfer of text files from a certain brand of standalone word processors to a certain minicomputer system. It describes the method used, an implementation of the method, and procedures for operators of the implementation. The word processor is a Lanier No Problem. The minicomputer is a Gould 9050. In it the Q-One word processor (part of Quadratron's Q-Office) runs under the UTX/32 operating system (a port of Berkeley Unix 4.2). IBM compatible personal computers, the Navy Document Interchange Format, and the Kermit file transfer protocol are used as intermediaries.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED/UNLIMITED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
RICHARD G. THORNETT	(202) 274-7661	DTIC-EA

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED/UNLIMITED

CONTENTS

FIGURES

1. INTRODUCTION

This report has three main purposes. The first is to document the method used by the Defense Technical Information Center (DTIC) to convert and transfer text files from its Lanier No Problem word processors to its Gould minicomputer system. The second is to document an implementation of the method used at DTIC, both for maintaining it and for possible use of it or a similar one elsewhere. The third is to document procedures to be used by operators of the implementation.

We suggest that operators read sections 1 through 4 (skipping matters too technical) to gain a general understanding of the system, and then concentrate on the parts where procedures for the steps of the conversion/transfer process are described in detail: Appendices C through E and section 5.3. Appendix B contains important background information for understanding Appendix C. Maintainers and implementers may wish to read the whole report, since a closer look at the procedures could clarify things in the main body of the the report.

Section 1.1 gives background information. Section 1.2 gives an overview of the conversion/transfer process. Section 1.3 describes notation used in this report. Section 1.4 gives the order of presentation in the rest of the document.

1.1 Background

The DTIC is a primary level field activity of the Defense Logistics Agency (DLA). A text transfer effort was required as part of DTIC's participation in a DLA conversion from various kinds of equipment and systems to distributed minicomputer systems. The new systems are for information management, office automation and end-user computing. Each DLA Distributed Minicomputer System (DMINS) provides the Q-One word processor as part of the Q-Office office automation software package. Before its DMINS became available, DTIC used Lanier No Problem word processors, which we will call 'Laniers.' These keep their files in a format unacceptable by Q-One and on diskettes illegible by microcomputers or other available devices. The transition to the DMINS required putting the Lanier text files onto a standard medium, converting them to a format acceptable by Q-One, transferring them to the DMINS, and converting them to Q-One documents.

1.2 Overview of the Lanier-to-DMINS Conversion/Transfer Process

There were two problems with transferring text files from our Laniers to our DMINS. One problem was how to get the text from here to there, that is from Lanier diskettes to the minicomputer. So the first problem may be symbolized:

LANIER -> DMINS

We decided to do this by using the one DTIC Lanier equipped for data communication.

The second problem was how to change the texts's format from Lanier format to Q-One format. This may be symbolized:

LIF -> QIF

where LIF means 'Lanier internal format' and QIF means 'Q-One internal format.' 'Internal format' here means the format in which a word processor saves a document after someone types it.

We symbolize both problems together like this:

LANIER(LIF) -> DMINS(QIF)

Here the form X(Y) is used, where X is a system and Y is a data format in X.

When the Lanier communicates a text file to another system, the file's format changes. We call the received format 'Lanier external format' (LEF). Since LEF is not acceptable by Q-One, we convert it to a format which Q-One accepts, namely Document Interchange Format (DIF), which is sometimes called 'Navy DIF' to distinguish it from other DIF formats. Q-One itself converts DIF files to QIF. So, if the Lanier were to send a file directly to the DMINS, the process would be:

LANIER(LIF) -> DMINS(LEF) -> DMINS(DIF) -> DMINS(QIF)

This method has not yet been tried, and, for practical reasons, a somewhat different method is presently being used. We did, however, try using modems and telephone lines to send files from the Lanier to a VAX minicomputer and a to a Sperry Univac mainframe computer, but communication frequently stalled. Sender and receiver remained connected, but the Lanier's keyboard locked. Since we saw no way to overcome this, we decided to send text files from the Lanier to a personal computer (PC), and then send them from the PC to the DMINS. The whole process, then, includes four steps:

```
                  1           2           3              4
LANIER(LIF) --> PC(LEF) --> PC(DIF) --> DMINS(DIF) --> DMINS(QIF)
```

LEF-to-DIF conversion is done in a PC, and DIF-to-QIF conversion is done in a DMINS. Boldface, underlining, overstrike, subscript and superscript are preserved by the process (see sections 5.1.2 and 5.2). The conversion/transfer process is described more fully in chapters 4 and 5 of this report.

1.3 Notation Used in This Report

In this report certain things are represented in a somewhat unusual ways Lanier command keyins are shown with plus signs and spaces which are not actually typed. For example, the command for deleting a paragraph from the screen is:

SCRN + DEL CHAR + P + EXEC

where SCRN means screen, DEL CHAR means delete character, P means paragraph, and EXEC means execute. This command requires four keypresses. The abbreviations are those which are printed on the Lanier key tops. On the DEL CHAR key, DEL appears above CHAR. This key is used not only to delete one character but also in commands which delete larger items, such as words or paragraphs. Again, the plus signs and the spaces beside them are not typed.

We underline key names to show that one key and not a sequence of keys, such

as S-C-R-N, is meant.

Sometimes presses of CTRL and another key are combined. An example is holding down CTRL, pressing 9, and then releasing both keys. We show this sequence as follows:

CRTL/9

It is used to key the character > (greater than) into the Lanier. The forward slash here represents holding down CTRL.

The above notations are modeled on the ones used in Lanier manuals.

The Lanier keyboard is described in section 2.1.1 below (see Figure 1).

We show the encoding of formatting control functions as sequences of characters in the American Standard Code for Information Interchange (ASCII). Each ASCII character sequence is shown in two ways, as in the following example:

ESC	[	$	H
1B	5B	24	48

where the first line shows the character itself or its standard abbreviation, and the second line shows the hexadecimal code for the character. An abbreviation is shown only if the character is not printable. Thus ESC represents the escape character. The sixteen hexadecimal digits are 0123456789ABCDEF, where A means ten and F means 15. More information on the ASCII code is given in Appendix A.

We often capitalize (a) a word or phrase at the point where it is being defined, and (b) a command or file name or other item in running text or an indented example where case makes no difference but capitals may be more readable.

1.4 Order of Presentation

Chapter 2 describes the hardware and software used in the conversion/transfer process. Chapter 3 describes the four data formats involved: LIF, LEF, DIF and QIF. Chapter 4 describes the two data transfers: Lanier-to-PC and PC-to-DMINS. Chapter 5 describes the data format conversions: LIF-to-LEF, LEF-to-DIF, and DIF-to-QIF. Chapter 6 summarizes this report.

2. HARDWARE AND SOFTWARE USED IN THE TRANSFER/CONVERSION PROCESS

This section briefly describes the hardware and software used in the Lanier-to-DMINS conversion/transfer effort.

Section 2.1 describes the Lanier word processor hardware and software used. Section 2.2 covers the PC hardware and software used. Section 2.3 covers the DLA DMINS hardware and software used.

2.1 Lanier No Problem Word Processor Hardware and Software

DTIC has several Lanier No Problem word processors, but only one is equipped for communications. We will call it "the Lanier." Sections 2.1.1 and 2.1.2 cover the Lanier No Problem hardware and software, respectively.

2.1.1 Lanier No Problem Hardware

The Lanier word processor used in the conversion effort consists of a main unit and a printer. The printer is useful but not essential to the effort. The main unit contains a communications printed circuit board with an RS-232 interface and a cable used to connect it with a modem or other device. This cable is connected to a null modem cable which is connected to the RS-232 connector in a PC. The null modem reverses the transmit data and receive data wires, as is required when a PC is directly connected to another PC or similar device.

The Lanier main unit has a keyboard, a CRT screen, and two 5-1/4 inch disk drives. The keyboard has most of the usual typewriter keys, a set of five function keys on the left, and a set of twelve word processing function keys on the right. Figure 1 depicts the Lanier keyboard layout.

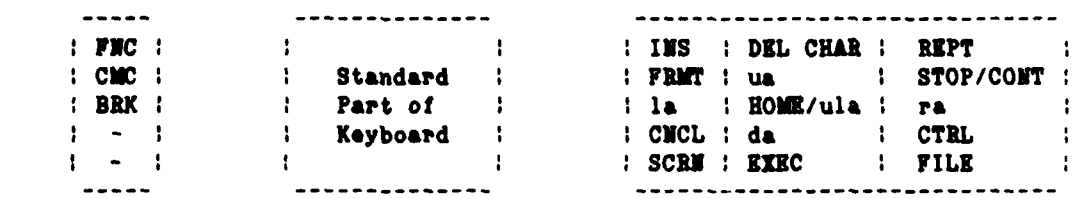
The following ASCII characters are not on the keyboard:

< > \ ^ ` { | } ~

Each of these can, however, be transmitted by holding down CTRL (or, in one case, SHIFT) and pressing another key, as follows:

Keyin	Character
CTRL/1	` (grave accent)
CTRL/2	~ (tilde)
CTRL/3	{
CTRL/4	|
CTRL/5	}
CTRL/8	<
CTRL/9	>
CTRL/=	\
SHIFT/6	^ (caret)

where SHIFT/6 is ordinarily the cents character, which is not an ASCII character. In this and some other cases the key is associated with the character by a communication transmit table mapping (see section 2.1.2 below).



Following are the meanings of the abbreviations in the depiction:

BRK	break	FRMT	format
CHAR	character	INS	insert
CMC	communication	REPT	repeat
CNCL	cancel	SCRN	screen
CONT	continue	da	down arrow
CTRL	control	la	left arrow
DEL	delete	ra	right arrow
EXEC	execute	ua	up arrow
FNC	function	ula	upper left arrow

where lower case symbols, such as ua, represent drawn arrows.

Figure 1. Lanier No Problem Word Processor Keyboard Layout

To use the Lanier one must put a program disk in its upper drive and load and run the program. This can be done by turning the machine on, or, if it is on, by pushing the red boot (load) button on the front of the machine. Each disk drive has a light used to indicate the current drive. This is the upper drive when a program starts. The current drive may be changed by the appropriate one of the following function key sequences:

FILE + da + EXEC
FILE + ua + EXEC

2.1.2 Lanier No Problem Software

Section 2.1.2.1 is on Lanier No Problem word processing software, section 2.1.2.2 is on Lanier No Problem pages and texts, and section 2.1.2.3 is on the Lanier No Problem TTY-ASCII data communications program.

2.1.2.1 Lanier No Problem Word Processing Software.

A Lanier No Problem machine is usually used for word processing. To do this a word processing program is first loaded from a disk and run. Such a program has many features and functions. Here we consider three commands relevant to communicating files. These are the commands for memorizing a page (copying what's in the Lanier's screen memory to a disk), for recalling a page from disk to screen memory, and for showing the index (file directory) of a disk. A page in screen memory is what shows on the CRT screen plus what is off-screen to the left or right or above or below. The arrow keys are used to bring off-screen symbols into view. The characters of approximately one printed page can be kept in the Lanier's screen memory at one time. Examples of the three commands are:

FILE + M + ABC + EXEC
FILE + M + EXEC
FILE + R + GHI + EXEC
FILE + R + EXEC
FILE + I + EXEC

where M means memorize, R means recall, I means index, and ABC and GHI are names given to pages. The last command displays the current disk's file index, which names each page and shows which pages are connected in texts (explained below) by putting a dividing line segment between page names not in a text or not in the same text. This divider appears as a faint dot followed by spaces to the end of the name column. The index also shows the number of available segments the disk has.

2.1.2.2 Lanier No Problem Pages and Texts

Sets of pages memorized on a disk can be combined into a text. A TEXT is a set of connected pages. (We will not consider a lone page to be a special case of a text.) A text is usually built by typing and memorizing consecutively two or more pages, specifying a name only in the first memorize command. The names of the pages in a text built this way start with the same character sequence but end with a dot and a sequence number (0, 1, 2, and so on). If you do not put a dot and a 0 after the name when memorizing the first page of a text, the index display will not show them for this page. On the other hand, if, after memorizing, say, page VACATION and memorizing two more

pages without naming them, you type:

FILE + M + VACATION.3 + EXEC

page VACATION.3 will be a separate page not connected to text VACATION.

Imitating IBM-DOS terminology, we will call the part before the dot the PAGENAME, the part after the dot the EXTENSION, and the combination of the two with a dot between them the PAGE SPECIFIER. Thus in the last example, VACATION is a pagename, 3 is its extension, and VACATION.3 is a page specifier. The extension must be a number between 0 and 99. If there is no extension, the pagename and page specifier are the same. A text is specified by giving the page specifier of its first page.

Pages with name extensions between the lowest and the highest in a text may be missing from the text, due to the way the text was formed. For example, a text might consist of pages SPEECH, SPEECH.1 and SPEECH.3. Sometimes pages with the same pagename and different extensions are not in the same text. Thus D-D.1-D.2 and D.7-D.8-D.9 might be two different texts. A case of a missing page can result from use of a delete page command, as in the following example:

FILE + DEL CHAR + P + SPEECH.2

A text can be augmented by inserting one or more separate existing pages or texts. For this, one of the two 'combine' commands (file insert page or file insert text) is used. For example, if pages X-X.1-X.2 are a text:

FILE + INS + P + X.3 X.1 + EXEC

would put X.3 in text X between X.1 and X.2.

Note that the specifier of the data unit (page or text) to be inserted is typed before the specifier of the text page after which it is to be inserted.

For another example, given two texts, Y and Z, each with extensions 0, 1, 2 and 3, in that order, typing:

FILE + INS + T + Y.0 + X.2 + EXEC

would cause text X to contain the following sequence of pages:

X.0, X.1, X.2, Y.0, Y.1, Y.2, Y.3, X.3

Text Y would cease to exist as a separate text and become what might be called a 'subtext' of text X. The page specifier of the text to be inserted need not be that of first page of a text. Thus if Y.1 were used instead of Y.0 in the previous example, the resulting text X would contain the sequence:

X.0, X.1, X.2, Y.1, Y.2, Y.3, X.3

and Y.0 would exist separately. If necessary, use combine commands to fix up a text before transferring it to a PC.

See section B.1 of Appendix B for more on the combine commands.

The pagenames of the pages in a text may all be different, and the extensions (if any) need not be in numeric order.

The order in which a text's pages are connected on their disk determines the order in which they are sent when the text is transmitted from the Lanier. Ordinarily each text should sent to a different PC file.

2.1.2.3 The Lanier No Problem TTY-ASCII Data Communications Program

The Lanier TTY-ASCII data communications program is used in our application. We will call it the 'TTY-ASCII program.' This program includes two editors which can do some of the functions done by Lanier No Problem word processing programs (see section B.1 of Appendix B). The TTY-ASCII program is loaded from a disk just like a word processing program. Then it may be removed.

After the TTY-ASCII program is loaded, a communication table must be loaded from a table file on a table disk. Then the table disk may be removed. A communication table is produced by ordinary Lanier word processing. The communication table we use is COMMUNITAB, which is shown in Figure 2.

The TTY-ASCII program uses the communication table when we type a message to a receiver, send a file, or receive characters from a sender. Each communication table has three parts: a parameter table, a transmit table and a receive table.

The PARAMETER TABLE consists of three rows specifying communications parameters such as parity, number of data bits, number of stop bits, speed, mode (half or full duplex), end of line sequence, and xon/xoff options. The values of these parameters in a given table file depend on the kind of communication to be done and the kind of system at the other end of the communication.

The TRANSMIT TABLE is a 16 by 16 matrix containing a cell for each character code which the Lanier might have to send. The first and second hexadecimal digits of a Lanier internal format (LIF) character's code specify the row and column number of the character's cell in the transmit table. Each cell in this table contains a two-digit hexadecimal number. The TTY-ASCII program consults this table each time it has a character to send. If the table cell value is FF (decimal 255), no character is sent. Otherwise, the cell value is sent. In our application, only seven data bits are transmitted, so no transmit table cell value (except FF) exceeds 7F (decimal 127). However, the second half of the transmit table is important, since the Lanier uses eight bits in encoding and saving pages in LIF. Most of the values outside the range 20 through 7F (decimal 32 through 127) have nonstandard meanings. For example, CE (206) represents the copyright symbol. We made some application-specific changes in the transmit table. These are covered in section 5.1.1 below.

The RECEIVE TABLE is similar to the transmit table but is only half as large. This table was much less important to our application than the transmit table.

The numbers within the transmit and receive tables must be separated by special spaces produced by backspacing. Such a space shows on the Lanier screen as a faint dot in the middle of the character's area.

```
02 00 00 02 02 00 0A 01 2A 01 00 00 00 00 FF 01
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E1 00 05 'LANIER TYPING ' E1 00 01

FF FF FF FF FF FF FF FF 1F FF 0A FF FF 1D FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF 5C 5E 0E 15
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
08 FF 1C 7D 7B 7E FF FF FF FF FF FF FF FF FF FF
0D FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF 0D FF 1B FF FF FF FF 20 20 1E FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

FF FF FF FF F7 1B FB FA 08 09 0A 1A F6 E1 F9 F8
FF 1C F3 1E F4 FF F5 FF FF FF FF FF FF FF FF FF
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 08 5C 5D 5E 5F
5C 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E FF
```

```
File COMMUNITAB 860922
For Lanier-to-PC
  transfers using
REV. 2.1.1 TTY Table
Mod. by Rich Thornett
EVEN PARITY
2400 BITS/SEC
FULL DUPLEX
7 DATA BITS
1 STOP BIT

Before send do:
  FRMT + H + 144 + EXEC
to preclude truncation.
```

Figure 2. Communication Table for Lanier No Problem TTY-ASCII Data Communications Program

The following command loads COMMUNITAB:

FNC + L + COMMUNITAB + EXEC

After COMMUNITAB is loaded, the Lanier communication function must be activated. Then the PC must be made ready to receive a file. When this is done, the Lanier must be told to send a page or a text or the whole disk. (The Lanier-to-PC transfer procedure is explained in Appendices B and C.)

2.2 Personal Computer Hardware and Software.

Section 2.2.1 describes the Sperry Univac PC. Section 2.2.2 covers the Zenith 248 PC.

2.2.1 Sperry Univac Personal Computer Hardware and Software.

The microcomputer we use to receive files from the Lanier is a Sperry Univac Personal Computer System, which we will call the 'Sperry PC.' It includes a keyboard, monitor, two 5-1/4 inch disk drives, and an asynchronous communications adapter used for RS-232 type communication. The Sperry PC is connected through a null modem cable to the Lanier (see section 2.1.1). The upper and lower disk drives are the A and B drives, respectively.

The operating system used in the PC is version 2.10 of the IBM Disk Operating System (PC-DOS) by Microsoft Inc. PC-DOS is practically the same as Microsoft's Disk Operating System (MS-DOS). We will call the operating system used MS-DOS. Microstuf Inc.'s Crosstalk XVI communications program is used in the PC to receive files from the Lanier and can be used to send files to a DLA DMINS. We will call the program 'Crosstalk XVI.' The Kermit file transfer protocol is used for PC-to-DMINS transfers.

An introduction to Lanier-to-PC file transfers is given in Appendix B. A step-by-step procedure for such transfers is given in Appendix C. A procedure for converting LEF files to DIF files in a PC is given in Appendix D. Kermit file transfers between a PC and a DMINS are covered in Appendices E through G.

2.2.2 Zenith 248 Personal Computer Hardware and Software.

The microcomputers we usually use to send DIF files to the DTIC DMINS are Zenith 248 Personal Computers. Each includes a keyboard, monitor, one 5-1/4 disk drive (the A drive), one 20-megabyte hard disk drive (the C drive), and an ALPS P2000G printer.

Although the procedure for connecting a Zenith 248 to the DMINS is simple, well documented, and widely known, section E.1 of Appendix E describes it.

The operating system used in the Zenith 248 is MS-DOS Version 3.20. The ZSTEMpc communications program is used to emulate a VT100 terminal when the Zenith talks with the DMINS. ZSTEMGpc is a graphics version of ZSTEMpc. We will refer to these programs generically as 'ZSTEM.' PC diskettes containing DIF files produced by LEF-to-DIF conversions are taken to one of the Zenith 248s, put in the A drive, perhaps copied to a C drive directory, and uploaded to the DMINS. The Kermit file transfer protocol is used. It is embodied in the Unix and ZSTEM Kermit processors. The Crosstalk XVI Kermit processor can also be used, but ZSTEM's is much faster, since it uses a 9600-baud

communication line, while in our system Crosstalk can use only slower lines.

Appendix E describes the ZSTEM procedures for file transfers between a Zenith 248 PC and a DLA DMINS computer. Appendix F describes the Crosstalk procedures for such file transfers. Appendix G describes the Kermit file transfer protocol.

2.3 Defense Logistics Agency (DLA) Distributed Minicomputer System (DMINS) Hardware and Software

We briefly treat the DLA DMINS hardware and software in sections 2.3.1 and 2.3.2, respectively.

2.3.1 DLA DMINS Hardware.

Each DLA DMINS is hosted by a Gould 9050 minicomputer and contains various storage, printing and communications devices. These include not only the Zenith PCs covered in section 2.2.2, but also Visual VT-102 'dumb' terminals and their printers.

The DTIC and some other DLA DMINS computers are connected to a Bell Atlantic Central Office based Local Area Network (C.O. LAN or CO-LAN), which allows customers to share information and resources by connecting terminals, workstations and computers. The C.O. LAN has gateway connections to the Direct Distance Dialing network and to Bell Atlantic or other Public Packet Switched Networks. Thus a DTIC DMINS user can remotely use, or send electronic mail to, or receive it from, another DMINS in the growing widely-distributed network of DMINS nodes.

2.3.2 DLA DMINS Software.

The operating system in the DLA DMINS is UTX/32, which is a port of Berkeley Unix 4.2. In this report we will refer to the DMINS operating system by the generic name Unix.

The DLA DMINS provides the following software relevant to our application: a Unix Kermit file transfer processor and the Quadratron Q-Office office automation package, including the Q-One word processor. In this report we generally assume that the reader has learned Q-Office software from courses and manuals, and cover only features closely related to the conversion/transfer process.

Q-Office is reached through the DMINS main menu, which looks like this:

```
        DEFENSE LOGISTICS AGENCY
DATA SYSTEMS AUTOMATION CENTER
------------------------------
1     Change Your Password
2     Q-Office (Word Proc / Mail)
3     Data Base
4     Electronic Mail Forwarding
5     Business Graphics
6     20/20 Spreadsheet
7     Lock this Terminal
8     Phonebook
9     Bourne Shell
10    C Shell
11    CAI Courses
12    INFO


Enter option or 'x' to logout
```

If you pick option 2, you get the Q-Office main menu. If you pick option 10, you get the Unix C Shell and its prompt (%), and can enter C shell commands. When finished with C Shell, type:

CTRL/D Return

to return to the DMINS main menu.

Section E.1 of Appendix E tells how to start a Zenith 248 PC and connect to the DMINS. If you then you correctly log into the DMINS and enter your password, you get the DMINS main menu.

2.4 Location of Hardware and Software Used

The location of the Lanier/Sperry PC subsystem is somewhat arbitrary but seems to work fine. It is due to proximity, generosity and cooperation.

The Lanier and the Sperry PC are in a room with other word processing equipment. This room is near the communicating Lanier owner's office, so that the owner could conveniently use it during pre-transfer days and non-transfer times. LIF-to-LEF processes have top priority. LEF-to-DIF format conversions can be done in the Sperry in this room. But, since the room has no telephone line for connecting with a DMINS, DIF file transfers to DMINS and DIF conversions to Q-One documents must be done from elsewhere, preferably using a Zenith 248 connected at 9600 baud to the DMINS. DIF-to-QIF conversions can also be done using a VT-102 'dumb' terminal.

The locations of the various DLA DMINS are beyond the scope of this report.

3. DATA FORMATS

The data involved in the Lanier-to-DMINS conversion/transfer process is stored in four formats: Lanier internal format, Lanier external format, Document Interchange Format, and Q-One internal format. These terms were introduced in the overview in section 1.2 above. Document Interchange Format is a standard format. The names of the other three formats were fabricated for this report. This chapter consists of four sections, each describing one of the four formats.

3.1 Lanier Internal Format (LIF)

Full information on how the Lanier represents text internally, that is, in its working memory and on disk, was not available and was not needed. A helpful reference was the "No Problem Transmit Table Quick Reference Chart" on page 5.17 of the Lanier document LTE-3 TTY-ASCII Tables, of which we were able to obtain only a few pages. This chart helped us build a TTY-ASCII communication table file suitable for our application.

Lanier internal format (LIF) uses an eight-bit code. The Quick Reference Chart is a 16 by 16 table. Rows and columns are numbered 0 through 15. The first half of the chart shows key presses in the cells in rows 0 and 1 and ASCII characters in those in rows 2 through 7. For example, cells 0 through 6 represent presses of the FRMT, FILE, SCRN, EXEC, CNCL, DEL CHAR and INS keys, while the corresponding cells in row 1 represent presses of the same keys while CTRL is pressed.

Special keyins are required to produce certain LIF items. Some LIF items seem not to be single characters. For example, start boldface is keyed in as CTRL/X + b. If this sequence is saved in a file, and the file is transmitted, start boldface is sent as the four-character sequence: ESC BS _ b. Thus it seems likely that the LIF item is not a single character.

3.2 Lanier External Format (LEF)

The format of text communicated from the Lanier depends on the LIF, the workings of the TTY-ASCII program, and the TTY-ASCII transmit table used. (The TTY-ASCII program and transmit tables were described in section 2.1.2 above.) For example, since the Lanier sends a carriage return character (0D) at the end of each line which does not end with a hard return, our transmit table changes a hard return into a GS (group separator) character (1D) to make lines which end in a hard return distinguishable from those which do not. (A hard return is entered by pressing RETURN.)

Another example is the BS (backspace) character (08). Lanier saved pages contain BS characters in control sequences. The Lanier ordinarily sends each underlined character as a 3-character sequence: the character followed by a BS followed by an underline character. The BS causes some receivers to drop the previous character, so that only the underline character is saved. To solve this problem, we have the Lanier send a US (unit separator) character (1F) instead of a BS. We arrange this by putting 1F in the BS (08) cell in the transmit table. LIF-to-LEF conversions are covered in section 5.1 below.

In the receiving PC, the LEF file is a standard MS-DOS text file, although some control character symbols (see Appendix A) may be unfamiliar to some.

3.3 Document Interchange Format (DIF)

The Document Interchange Format (DIF) is a representation (encoding) of control information used in formatting documents. It was developed by the Institute for Computer Sciences and Technology at the National Bureau of Standards, at the request of the Department of the Navy. It is sometimes called "Navy DIF" to distinguish it from other formats called "DIF." We will just call it "DIF." Its purpose is to facilitate document interchange among text processing systems of different vendors.

In a text processing system, a document is composed of printed characters and control information used to format the document. When a document is sent from one text processing system to another, the control information must be correctly interpreted by the receiving system. The DIF, which is defined in reference 2, is a representation of document-formatting control functions. It was based on existing standards and uses standard code extension techniques.

Section 3.3.1 describes the DIF. Section 3.3.2 covers correspondences between LEF and DIF functions.

3.3.1 Description of the Document Interchange Format

The basic code table for 7-bit ASCII may be represented as a matrix consisting of eight columns and sixteen rows (see Appendix A). Columns 0 and 1 contain the control (non-printing) characters, which are called the C0 set. Columns 2 through 7 contain human readable (printing) characters, which are called the G0 (graphic) set. Since text processing systems do not always agree regarding the C0 set functions, the DIF uses only two C0 characters: Horizontal Tab (09) and Escape (1B), abbreviated HT and ESC.

The DIF contains 43 functions:

1	C0 character (HT)
13	previously standardized functions
29	new functions

The DIF uses a Control Sequence Introducer (CSI) to expand the control functions as defined by the C0 set. The general format of a DIF control sequence is as follows:

CSI P...P I...I F

where

CSI is the two-character sequence ESC [

P...P are 0 or more parameters composed of decimal digits and sometimes separated from each other by the character ;

I...I are intermediate characters used to expand the set of functions defined by F

F is a function-defining character taken from the set SP through ~ (20 through 7E).

For example, the control sequence

ESC [7 2 # A (1B 5B 31 32 24 41)

is a right margin set function, in which 72 is a parameter, # is an intermediate character and A specifies the function.

Only four DIF functions do not start with the CSI:

HT	horizontal tab
ESC E	hard new line
ESC K	subscript on (superscript off)
ESC L	subscript off (superscript on)

3.3.2 Correspondences between LEF and DIF Functions

All LEF functions, except stop printing, are convertible to DIF. The LEF functions are underscore, emphasis, overstrike, subscript, superscript, hard space, hard return, and justification. Justification means inserting just enough spacing in each line to put the last character at the right margin. This is done by loading a special program from a special Lanier disk and applying the following commands to each page where justification is desired:

SCRN + J + L + EXEC
SCRN + J + P + EXEC
SCRN + J + R + EXEC

where J stands for justify, L stands for line, P for paragraph and R for remainder. The result is that one of a set of a special character sequences is inserted in the left margin of each line to be justified. Justification takes place only when a page is printed. Similarly, justification is removed by the following commands:

SCRN + J + E + L + EXEC
SCRN + J + E + P + EXEC
SCRN + J + E + R + EXEC

where E stands for erase.

Conversion to Q-One of justified lines in a Lanier text can be done by erasing justification from each Lanier page before sending the text to the PC, and restoring justification after the text is converted to a Q-One document. This is done by setting to the letter 'y' the justification parameter in the format style section of Q-One's extended format menu. Justification is done when the document is printed. Removal of justification indicators from a Lanier text might be handled by a special program to remove justification from a LEF text before converting it to DIF. We recommend that justification be removed by the Lanier commands listed above.

Some DIF functions are not found in LEF. Some of these could be added to DIF files before they are converted to QIF. Examples are:

Document Start
Pitch (10 or 12 characters per inch)

Margins (top, left, right, bottom)
Hard Page Start
Page Length and Page Width
Page Numbering Initial Value

An initial DIF function sequence is put in each DIF file by the LEF-to-DIF converter (see section 5.2 below). The user must select a subset of initial DIF functions, including page width and length, margins, and horizontal and vertical spacing. These may be selected individually or by picking the identifier of a preset subset. Settings specified by an initial DIF sequence can be changed by Q-One commands after DIF-to-QIF conversion. (See section D.1 of Appendix D.)

3.4 Q-One Internal Format (QIF).

Q-One internal format (QIF) is the format in which the DMINS word processor Q-One stores a document after someone has used Q-One to create a new document or modify an old one. A knowledge of this format is not required for Lanier-to-DMINS conversion/transfers. Conversion of documents from DIF-to-QIF is covered in section 5.3 below.

4. DATA TRANSFERS

This chapter describes the data transfer steps in our application. These were introduced in section 1.2 above. Section 4.1 covers Lanier-to-PC data transfers. Section 4.2 covers PC to DMINS data transfers.

4.1 Lanier to Personal Computer Data Transfers

At the outset, let's not forget some very important steps preliminary to conversion/transfer process itself, namely going through the Lanier disks, analyzing their contents, selecting documents to be transferred, and making sure that they are separate and that their pages are connected correctly (see sections 2.1.2.2 and B.1 in Appendix B). In some cases it may be easier to do a little document content cleanup now than after conversion to a Q-One document.

A Lanier-to-PC data transfer is done by two communications program: TTY-ASCII in the Lanier and Crosstalk XVI in the PC (see Chapter 2 above). The operator activates a receiving function in the PC and then activates a sending function in the Lanier. The Lanier reads text from a disk in its upper drive, sends it through the connecting null modem cable to the PC, which writes it to a disk file in its B drive. (A procedure for such a transfer is described in Appendices B and C.)

Sender and receiver are set for 2400-baud transmission speed. The LEF file produced in the PC is a standard MS-DOS text file (containing LEF sequences). When one is displayed, some control character symbols (see Appendix A) may be unfamiliar to some.

4.2 Personal Computer to DMINS Data Transfers

For PC-to-DMINS data transfers the Kermit file transfer protocol is used. This is embodied in the PC in the Kermit functions of the ZSTEM and Crosstalk communications programs, and in the DMINS in the Unix Kermit processor. You can view the online manual for the Unix Kermit processor by typing:

man kermit Retrn

when you have the Unix prompt, but this is unnecessary.

The procedure for a PC-to-DMINS file transfer may be summarized as follows:

1. Use ZSTEM or Crosstalk to connect to the DMINS and log in.
2. Get the DMINS main menu.
3. Select the C Shell option (item 10) and get the Unix prompt (%).
4. Enter a Unix Kermit receive command.
5. Get the PC command prompt.
6. Enter one or more Kermit send commands, each naming a file or set of files to be sent.
7. When all files have been sent return to the DMINS main menu.

The PC Kermit processor shows status information while the transfer proceeds.

In the Kermit protocol, control information and data is sent in packets composed of 7-bit ASCII characters. Each packet includes a sequence number

and a checksum of one or more characters. Acknowledge (ACK) and negative acknowledge (NAK) packets are used. If a sent packet is not acknowledged, it is sent again. A single Kermit command can send many files, and transmission could go on for hours without operator intervention.

Procedures for doing a Kermit file transfer between a PC and a DMINS computer are described in Appendices E and F. The Kermit protocol itself is described in Appendix G.

5. DATA FORMAT CONVERSIONS

This chapter describes the three kinds of data format conversion done in our Lanier-to-DMINS conversion/transfer effort. Section 5.1 covers LIF-to-LEF conversion. Section 5.2 covers LEF-to-DIF conversion. Section 5.3 covers DIF-to-QIF conversion.

5.1 Lanier Internal to Lanier External Format Conversion

Section 5.1.1 covers transmit table modifications. Section 5.1.2 covers each function whose LEF representation is a sequence of characters. Section 5.1.3 summarizes the transmit table's effect.

5.1.1 Lanier Transmit Table Modifications

The following table shows transmit table modifications peculiar to our application (see section 2.1.2.3):

Keyin	Cell	LEF	ASCII		Use
	08	1F	US	unit separator	stands for backspace
	0D	1D	GS	group separator	hard return
CTRL/=	9C	5C	\	backward slash	typing to receiver
SHIFT/6	9D	5E	^	caret	typing to receiver
CTRL/0	9E	0E	SO	shift out	overstruck 0
CTRL/u	9F	15	NAK	negative acknowledge	typing to receiver
CTRL/p	C0	08	BS	backspace	typing to receiver
CTRL/5	C3	7D	}	right curly brace	
CTRL/3	C4	7B	{	left curly brace	
CTRL/2	C5	7E	~	tilde	
CTRL/x	E3	1B	ESC	escape	control function sequence starter
CTRL/SPACE	EA	1E	RS	record separator	hard space

The Keyin column tells how an instance of the cell identifier gets into a LIF text or into a message typed to the receiver, depending on whether the current function is editing or communicating. The LEF column shows the value in the cell.

We mapped the LIF backspace to the unit separator (08 to 1F) to prevent loss of underlined characters. In LIF each underlined character is represented by the character followed by a backspace followed by an underline character. If the backspace is sent, some receivers drop the first character.

We mapped the LIF paragraph separator (hard return), which is entered by pressing RETURN, to GS (0D to 1D) to make it distinguishable from the soft return (0D) sent at the end of each line. The receiving computer program may or may not put a LF character after each CR received.

We mapped the LIF divide symbol to the backward slash character (9C to 5C) to make it possible to send a backward slash, which is important in Unix. (The divide symbol is not expected to appear in the files to be converted.)

We mapped the LIF overstruck 0 to the SO character (9E to 0E) to make it distinguishable.

We mapped CTRL/T and CTRL/U to NAK (9F to 15) to make it possible to send a NAK, which some systems interpret as erase line.

We mapped the paragraph symbol to backspace (C0 to 08) to make it possible to send a backspace while 08 is mapped to 1F.

We mapped an unknown character to tilde (C5 to 7E) to make it possible to send tildes.

We mapped CTRL/W to ESC (E3 to 1B). This character starts certain control sequences described below. It sometimes results from typing CTRL/X, for example in typing start of boldface.

We mapped hard space to record separator (EA to 1E) to make it distinguishable from an ordinary space.

The significance of some mappings in transmit tables received from Lanier representatives were not understood.

5.1.2 Functions Whose LEF Representations Are Sequences of Characters.

The LEF representation for the following functions is a sequence of characters: boldface, underlining, overstrike, subscript, superscript, and stop printing, which is not a DIF function.

BOLDFACE (emphasis) is represented in LEF by a start bold and a stop bold sequence. These are typed as CTRL/X + b and CRTL/X + e, and are sent as:

ESC	US	_	b	and	ESC	US	_	e
1B	1F	5F	62		1B	1F	5F	65

Each character between these two sequences is emphasized.

UNDERLINING is represented in LEF by sending a backspace and an underline character after each character to be underlined. For example, "It is so!" is sent as:

I	t	SP	i	US	_	s	US	_	SP	s	o	!
49	74	20	69	1F	5F	73	1F	5F	20	73	6F	2E

In Lanier editing, to start or stop automatic underlining type SCRN + -.

OVERSTRIKE is represented in LEF by an ESC followed by the overstriking character. For example, ^ overstruck by ~ is typed as ^ + CTRL/x + W + ~ and sent as:

^ ESC ~ (5E 1B 7E)

SUBSCRIPT is represented in LEF by putting the sequence "ESC US _ z" before each character to be subscripted. It is typed as CRTL/x + z. For example:

2	3	ESC	US	_	z	1	ESC	US	_	z	6
32	33	1B	1F	5F	7A	31	1B	1F	5F	7A	36

is sent for 23_{16} (23 base 16).

SUPERSCRIPT is represented in LEF by putting the sequence 'ESC US _ y' before each character to be superscripted. It is typed as CRTL/x + y. For example:

a	ESC	US	_	y	1	ESC	US	_	y	0	SP	+	SP	b
61	1B	1F	5F	79	31	1B	1F	5F	79	32	20	2B	20	62
	10													

is sent for a + b.

STOP PRINTING is represented in LEF by the sequence 'ESC US _ s'. This is typed as CTRL/X + s. The function is not in the DIF. The LEFDIF converter puts 'LANIER STOP PRINTING CODE WAS HERE' in its place in the DIF file.

5.1.3 Summary of Transmit Table Effects.

The transmit table has the following effects:

1. All the ASCII graphic characters except DEL (20 through 7E) can be sent. (But the Lanier sends a DEL after each page.)

2. The sendable control characters and their meanings are:

Keyin	Character		Meaning
CTRL/p	08	BS	backspace
	0D	CR	soft return
CTRL/0	0E	SO	overstruck 0
CTRL/u	15	NAK	cancel line
CTRL/x	1B	ESC	control function sequence starter
RETURN	1D	GS	hard return
CTRL/SPACE	1E	RS	hard space
	1F	US	backspace in a control sequence

3. Only 105 of the 128 ASCII characters are sendable. The 23 not sendable are: 00-07, 09, 0B, 0C, 0F, 10-14, 16-1A, and 7F. But the Lanier automatically sends a 7F (DEL) after each page.

4. The control functions boldface, underline, overstrike, subscript, superscript, and stop printing are sequences composed of ESC, US, underscore and graphic characters.

The MS-DOS operating system in the receiving PC puts a SUB (1A) character at the end of each disk file it closes.

5.2 Lanier External to Document Interchange Format Conversion

To convert LEF functions into DIF functions a special computer program called LEFDIF was developed by the author of this report. The source code file for this program not yet frozen and deliberately not provided in this report, to allow for improvements in its user friendliness and internal documentation.

This program requires at the outset that the user select values for the following parameters: page length, page width, left margin, right margin, top margin, bottom margin, vertical spacing (lines per inch) and horizontal spacing (characters per inch). These are explained in Appendix D, which describes the LEF-to-DIF conversion procedure.

LEFDIF then passes twice through the input LEF file and outputs a DIF file. Pass 1 checks the left margin of each line and pauses if it is smaller than specified. This permits the user to stop LEFDIF and clean up the LEF file if desired (rarely). For example, sometimes during the LIF-to-LEF step characters typed at the Lanier get into the LEF file at the start of a page, making the left margin 1.

Pass 2 inserts an initial DIF function sequence at the start of the output file. This sequence includes the values specified by the user in Pass 1. The program is written in the Microsoft C programming language and runs under MS-DOS in an IBM or Sperry or Zenith 248 PC. (The procedure is described in Appendix D).

Following are DIF functions corresponding to the LEF functions:

Emphasis (boldface) on and off are:

ESC	[	1	m	ESC	[	2	2	m
1B	5B	31	6D	1B	5B	32	32	6D

Underscore (underline) on and off are:

ESC	[	4	m	ESC	[	2	4	m
1B	5B	34	6D	1B	5B	32	34	6D

Overstrike on and off are:

ESC	[	9	m	ESC	[	2	9	m
1B	5B	39	6D	1B	5B	32	39	6D

Subscript on and off, and superscript on and off are:

ESC	K	ESC	L	ESC	L	ESC	K
1B	4B	1B	4C	1B	4C	1B	4B

The single-character LEF functions are coded as follows:

hard space	ESC	[	#	H	(1B	5B	24 48)
hard new line	ESC	E			(1B	45)	

Overstruck 0 is not a special case in DIF but is encoded using the overstrike on and off control sequences.

Justify on and off are not in LEF, but in DIF they are:

ESC	[	2	SP	F	ESC	[	0	SP	F
1B	5B	32	20	46	1B	5B	30	20	46

A standard initial DIF function sequence is shown in Figure 3. It was produced by the Q-One funnction which converts a Q-One document to DIF. The function shown between page format and left margin is unknown.

Procedures for converting LEF files to DIF, and for displaying DIF files, are described in Appendix D.

```
DOC START                PAGE LENGTH = 66              TOP MARGIN = 7
ESC   [   1   #   K ESC   [   6   6   #   L ESC   [   7   #   B

BOTTOM MARGIN = 60       PAGE WIDTH = 85          PG. NR. IN. VAL.
ESC   [   6   0   #   C ESC   [   8   5   #   M ESC   [   1   #

= 1 JUSTIFY OFF             PITCH = 10             LINE HEIGHT =
  N ESC   [   0  SP   F ESC   [   0  SP   K ESC   [   0  SP   L

PAGE FORMAT = PORT.                                LEFT MARGIN = 11
ESC   [   4  SP   J ESC   [   4  SP   F ESC   [   1   1   #   @

RIGHT MARGIN = 75        LINE SPACING = 1   TEXT TAB SET
ESC   [   7   5   #   A ESC   [   0   #   \ ESC   [   9   ;   1


  7   ;   2   5   ;   3   3   ;   4   1   ;   4   9   ;   5   7


  ;   6   5   ;   7   3   ;   8   1   ;   8   9   ;   9   7  SP

    DECIMAL TAB SET
  N ESC   [   0   ;   1   7   ;   2   5   ;   3   3   ;   4   1


  ;   4   9   ;   5   7   ;   6   5   ;   7   3   ;   8   1   ;


                              HARD PAGE START
  8   9   ;   9   7   #   G ESC   [   1   #   J
```

Figure 3. A Standard Initial DIF Function Sequence.

5.3 Document Interchange to Q-One Internal Format Conversion

Conversion of DIF files to Q-One internal format (QIF) files is the last step in our conversion/transfer process. It is done by the Q-One word processor and does not require a PC. A Visual VT-102 'dumb' terminal can be used from here on.

Let's assume that you have connected to the DMINS, logged in, and gotten the DMINS main menu, which currently looks like this:

```
        DEFENSE LOGISTICS AGENCY
DATA SYSTEMS AUTOMATION CENTER
-------------------------------
1     Change Your Password
2     Q-Office (Word Proc / Mail)
3     Data Base
4     Electronic Mail Forwarding
5     Business Graphics
6     20/20 Spreadsheet
7     Lock this Terminal
8     Phonebook
9     Bourne Shell
10    C Shell
11    CAI Courses
12    INFO


      Enter option or 'x' to logout
```

The following procedure is used for DIF-to-QIF conversions.

1. Select Q-Office (option 2) and get 'Q-OFFICE MAIN MENU.'
2. Select Word Processing and get 'WORD PROCESSING MENU.'
3. Select Supervisory Functions and get 'SUPERVISORY FUNCTIONS' menu.
4. Select Convert A File and get 'FILE CONVERSION' menu.
5. Select DIF to Q1 and get and respond to two prompts.
   At the 'DIF Filename' prompt, type the name and press Return.
   At the 'New Q1 Filename' prompt, type the name and press Return.
   The messages 'In Progress' and 'Operation Complete' appear in succession, and the 'FILE CONVERSION' menu reappears. The converted file is now a Q-One document file, like one produced by the Create New Document word processing option.
6. Press Cancl twice to return through the Supervisory Functions menu to the Word Processing menu.
7. Select Edit Old Document to view or modify the new Q-One file.

The DIF-to-Q1 conversion in step 5 does not put in soft page breaks. You can do this as part of step 7 while viewing the document. Press Command, highlight the the Hyphenation + Pagination command, and select it by pressing Return. Highlight your hyphenation option by pressing the spacebar repeatedly, and select it by pressing Return. Then select automatic or interactive pagination in the same way. Try selecting none and automatic.

6. SUMMARY OF THIS REPORT

This report describes DTIC's procedure for converting documents produced by its Lanier No Problem word processing machines to Q-One word processor documents in its Defense Logistics Agency (DLA) Distributed Minicomputer System (DMINS).

Two FILE TRANSFERS are performed. In Transfer 1, a document on disk in the Lanier is communicated through a null modem cable to a Sperry PC and stored in a file on a disk in it. In Transfer 2, a DIF file is communicated from a PC over telephone lines to the DMINS computer. The Crosstalk XVI and ZSTEM communications programs are used in these two transfers, and the Kermit file transfer protocol in used in Transfer 2.

Four text DATA FORMATS are involved in the conversion: Lanier Internal Format (LIF), Lanier External Format (LEF), Document Interchange Format (DIF), and Q-One Internal Format (QIF).

Three FORMAT CONVERSIONS are performed. Conversion 1 (LIF-to-LEF) is done during Transfer 1. Conversion 2 (LEF-to-DIF) is done before Transfer 2 in the Sperry PC or in a Zenith 248 or other PC. The same conversion program would be used in any case. Conversion 3 (DIF -to-QIF) produces a Q-One document file. This conversion is a Q-One function done in the DMINS and can be done using a suitable 'dumb' terminal or PC.

The two data transfers, the four data formats, and the three format conversions are described in the body of this report.

Procedures for the file transfers and format conversions are described in the appendices below.

APPENDIX A

THE AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

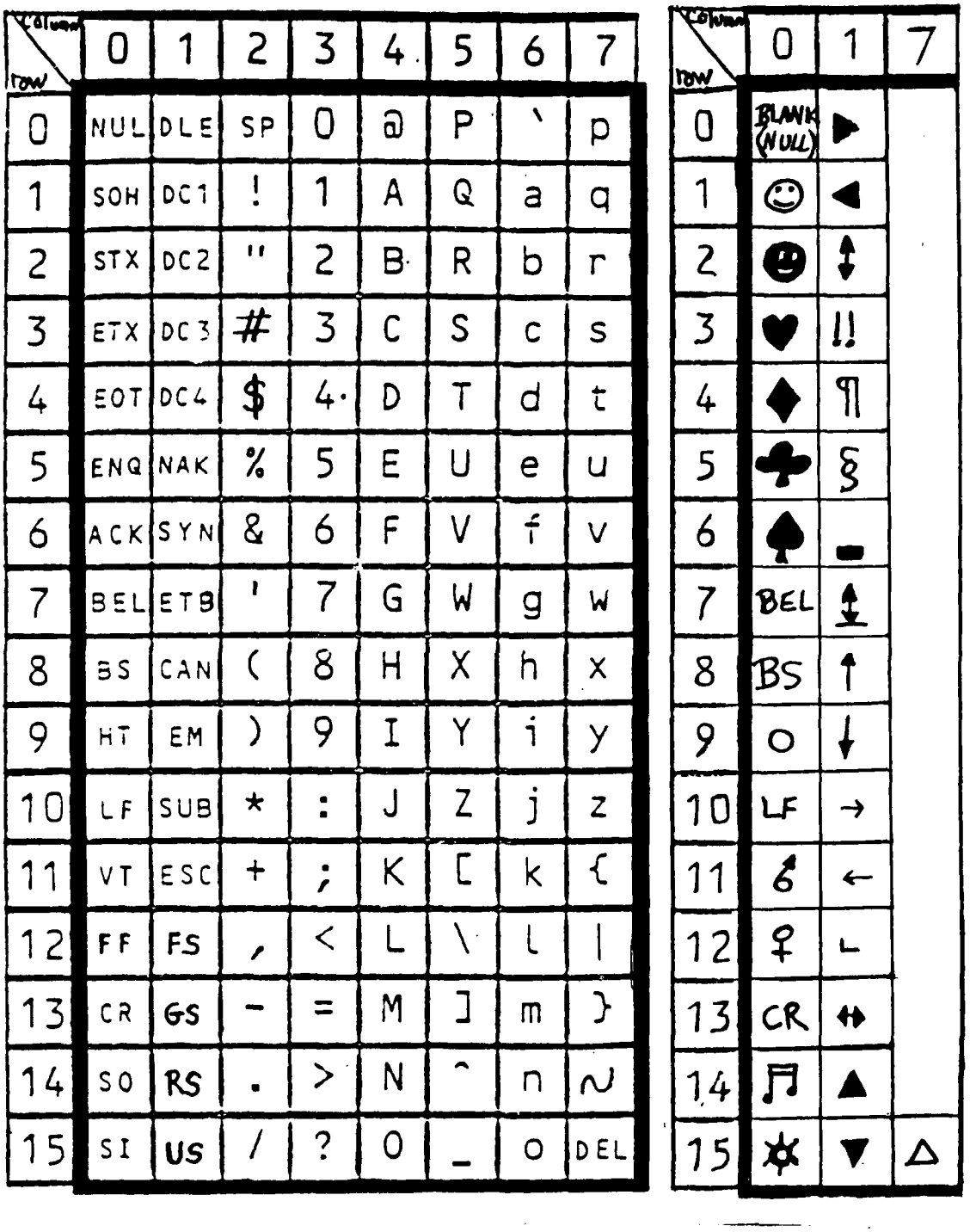
The acronym ASCII can refer to a coded character set or to a widely used asynchronous communications protocol based on that code. The ASCII character set consists of 7-bit coded characters (8 bits including parity check bit) used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters, graphic characters such as $ and +, and alphanumeric characters.

The basic code table for 7-bit ASCII is defined by standard X.34 of the American National Standards Institute (ANSI). This is represented as a matrix consisting of 8 columns and 16 rows (see Figure 4). Columns 0 and 1 contain the control (non-printing) characters and represent the C0 (control) set. Columns 2 through 7 contain human-readable (printing) characters and represent the G0 (graphic) set. A particular character is specified by indicating the column number followed by a forward slash followed by the row number. Thus 'K' is specified by 4/11. In this report we specify a character by giving column and row as hexadecimal digits. Thus we specify 'K' by 4B.

The meanings of the abbreviations used in the ASCII code table are:

ACK	Acknowledge	FF	Form Feed
BEL	Bell	FS	File Separator
BS	Backspace	GS	Group Separator
CAN	Cancel	HT	Horizontal Tab
CR	Carriage Return	LF	Line Feed
DC1	Device Control 1	NAK	Negative Acknowledge
DC2	Device Control 2	NUL	Null
DC3	Device Control 3	RS	Record Separator
DC4	Device Control 4	SI	Shift In
DEL	Delete	SO	Shift Out
DLE	Data Line Escape	SOH	Start of Header
EM	End of Medium	SP	Space
ENQ	Enquire	STX	Start of Text
EOT	End of Transmission	SUB	Substitute
ESC	Escape	SYN	Synchronous Idle
ETB	End of Transmission Block	US	Unit Separator
ETX	End of Text	VT	Vertical Tab

DC1 and DC3 are often used for flow control and called XON and XOFF (transmitter on and transmitter off). DC2 and DC4 may be used similarly for a second device.



Row \ Column	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SUB	*	:	J	Z	j	z
11	VT	ESC	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	|
13	CR	GS	-	=	M	]	m	}
14	SO	RS	.	>	N	^	n	~
15	SI	US	/	?	O	_	o	DEL

Row \ Column	0	1	7
0	BLANK (NULL)	▶	
1	☺	◀	
2	☻	↕	
3	♥	!!	
4	♦	¶	
5	♣	§	
6	♠	▬	
7	BEL	↨	
8	BS	↑	
9	○	↓	
10	LF	→	
11	♂	←	
12	♀	∟	
13	CR	↔	
14	♫	▲	
15	☼	▼	△

Figure 4. Seven-Bit ASCII Code Table with Screen Symbols for Control Characters

APPENDIX B

INTRODUCTION TO FILE TRANSFER FROM A LANIER NO PROBLEM WORD PROCESSOR TO A PERSONAL COMPUTER

This appendix describes a procedure for sending files from a Lanier No Problem word processor to a Sperry or IBM Personal Computer (PC) or a similar PC. It assumes that a particular system is to be used, and that the operator of the system is familiar with Lanier No Problem word processing. Although the machines are quite close together, two people can operate the system, if they are synchronized, and perhaps get the work done more quickly.

This appendix gives background information helpful in understanding the procedure. Appendix C gives a concise step-by-step description of the procedure for the operator to follow when performing it.

In the assumed system, DTIC's communicating Lanier word processor is connected to a Sperry PC by means of a null modem cable. The Lanier and the PC each have a keyboard, a CRT screen, and two disk drives. In the Lanier, a communications program called ASCII-TTY and a communication table called COMMUNITAB are used. In the PC, a version of Microsoft, Inc.'s MS-DOS operating system and Microstuf, Inc.'s Crosstalk XVI communications program are used. The information on MS-DOS and Crosstalk needed by the operator, which is relatively limited, is given below.

Section B.1 covers use of the TTY-ASCII program and table COMMUNITAB. Section B.2 covers relevant features of the Microsoft Disk Operating System (MS-DOS). Section B.3 covers use of the Crosstalk XVI. Section B.4 covers use of a transfer name table.

B.1 Use of the Lanier No Problem TTY-ASCII Data Communications Program and Table COMMUNITAB

At the outset, let's not forget some very important steps preliminary to conversion/transfer process itself, namely going through the Lanier disks, analyzing their contents, selecting documents to be transferred, and making sure that they are separate and that their pages are connected correctly (see section 2.1.2.2).

DTIC has exactly one Lanier word procesor equipped for communications. To use it to communicate, two disks are required. One contains the TTY-ASCII program, and the other contains a communication table specifying characteristics of the communication, such as its speed and certain character translations (see section 2.1.2 above).

Three special keys are used in communicating. These are the top three keys in the function key group on the left of the keyboard (see Figure 1 above). From the top these are the FUNCTION (FNC), COMMUNICATION (CMC), and BREAK (BRK) keys. We will refer to a key by the name printed on or beside it and underline the name.

The TTY-ASCII program recognizes five FUNCTION commands:

FNC + L + table_name + EXEC	Load a communication table
FNC + D + EXEC	Display communication table
FNC + C + EXEC	Go into communication mode
FNC + O + EXEC	Go into Online Editor mode
FNC + E + EXEC	End communications link and go into Offline Editor mode

The first command loads a communication table. The second command displays the communication table currently loaded. The third command puts the Lanier in communications mode. The fourth command puts the Lanier in Online Editor mode. This lets you use certain editing commands without breaking the communications link. The fifth command puts the Lanier in Offline Editor mode, breaking the communication link in some cases (not ours). To return to communications mode execute the C function command.

You load the TTY-ASCII program from a disk just like a word processing program. Insert the disk in the upper drive. INSERT here means put it in the drive and close the door. If the Lanier is off, turn it on. If it is already on, push the red load (boot) button. The drive light will go on and off twice. Next remove the TTY-ASCII disk and insert in the upper drive the communication table disk marked LCTABLE at the top of its right-hand label. The drive light goes on and off once (as the disk's index is read). If this does not happen, remove it and repeat the insertion until it does. From LCTABLE load COMMUNITAB by typing:

```
FNC + L + COMMUNITAB + EXEC
```

Then set the page width by typing:

```
FNC + E + EXEC
FRMT + H + 144 + EXEC
FNC + C + EXEC
```

where H stands for horizontal. The format command SETS THE PAGE WIDTH TO 144. IT IS IMPORTANT THAT THE LANIER PAGE WIDTH SETTING BE LARGE ENOUGH, SINCE IF IT IS TOO SMALL, LINES MAY BE TRUNCATED WHEN TRANSMITTED. The last command puts the Lanier in communications mode.

Now the Lanier is ready to send data. Each send command specifies a DATA UNIT, which here means a page, a text or a whole disk. (In our Lanier-to-PC step whole disk commands are not normally used.) A PAGE is produced by copying to disk the content of the Lanier's screen memory. A TEXT is a sequence of connected pages. A TEXT can be built by typing and memorizing two or more pages, giving no page name in the second and subsequent memorize commands (see section 2.1.2.2 above). To show the names of the items on a disk type:

```
FNC + E + EXEC
FILE + I  EXEC
FRMT + H + 144 + EXEC
FNC + C + EXEC
```

where I stands for index. In the shown index the names of pages in a text

appear on successive lines, while each lone page and text is separated from the next data unit by a blank line with a faint dot at the left.

To start sending a data unit use one of the following commands:

```
CMC + T + P + page_name + EXEC
CMC + T + T + text_name + EXEC
CMC + T + D + EXEC
```

where T stands for transmit, P for page, T for text, D for disk, and page_name and text_name for names you pick (see section 2.1.2.2). The third command is rarely used, since it puts all the Lanier disk's texts into one PC file, even though they may have different characteristics.

To stop transmitting a data unit in case of emergency, type the following:

```
CTRL/CNCL
FNC + C + EXEC
```

where CTRL/CNCL means hold down CTRL and press CNCL.

A document may exist on a Lanier disk in two or more separate parts, each part being a page or a text. It is good to connect the parts in the correct order into a text before sending the document to the PC. Two Lanier COMBINE COMMANDS are available for doing this. We will give examples of their use, showing Lanier disk index information in a horizontal format more compact than that of a Lanier-produced index display.

Suppose the Lanier A drive disk contains just five pages: AAA, BBB, CCC, DDD and EEE, and the file index is equivalent to this:

```
AAA  BBB  CCC  DDD  EEE
```

which shows that no pages are connected. To form a text containing AAA followed by BBB, and another containing DDD followed by EEE, type:

```
FILE + INS + P + BBB AAA + EXEC
FILE + INS + P + EEE DDD + EXEC
```

where P stands for page. Note that the 'following page' name is typed before the 'preceding page' name, which is the reverse of their order in the file index after the combine commands are executed. The file index becomes:

```
AAA_BBB  CCC  DDD_EEE
```

where we put an underline character between pages connected in a text and two spaces between data units not so connected. To connect the two texts type:

```
FILE + INS + T + DDD BBB + EXEC
```

where T stands for text. The file index becomes:

```
AAA_BBB_DDD_EEE  CCC
```

that is, we now have a four-page text and lone page. Suppose we next try to

insert page CCC after BBB. What happens to the BBB_DDD connection? Typing:

FILE + INS + P + CCC BBB + EXEC

produces one text such that the file index becomes:

AAA_BBB_CCC_DDD_EEE

If we now type:

FILE + DEL CHAR + P + CCC + EXEC

the file index becomes:

AAA-BBB-DDD-EEE

Page CCC is gone, but the other text pages are still connected in the same order as before.

When a Lanier document to be transmitted is fragmented, use the combine commands to connect the parts into a text before sending it.

B.2 Use of Microsoft's Disk Operating System (MS-DOS)

The Microsoft Disk Operating System (MS-DOS) is used in the PC. MS-DOS is a group of programs which (a) enable you to organize and use the information you put on disks and (b) control the way your computer uses other programs. We will sometimes refer to MS-DOS as DOS.

We use the MS-DOS system disk marked PCXTALK at the top of its right-hand external label.

To load and start MS-DOS, put disk PCXTALK in the A (upper) drive. Then, if the PC is turned off, turn it on. If it is on, hold down Ctrl and Alt, and press Del. This is called SYSTEM RESET and can be used to restart MS-DOS.

When MS-DOS is loaded and started, you are prompted for the date and time. It is important to make them correct, since MS-DOS saves the creation date and time of each disk file, and this information helps you identify your data. Enter the date and time like this:

11-22-87 Retrn
15:28 Retrn

that is, November 22nd, 1987 at 3:28 p.m.

After you enter date and time, the following PROMPT appears:

A>

where 'A' indicates that the A drive is the default drive and '>' prompts you to input a command. If you give DOS a filename without specifying a drive, DOS searches for the file on the disk in the DEFAULT DRIVE. Another name for the default drive is the CURRENT DRIVE. To change the current drive enter a

DRIVE SPECIFIER, which is a drive letter followed immediately by a colon:

B: Retrn or A: Retrn

In PC commands we show an optional space between parts and before Retrn.

We assume that throughout the transmission session the A drive is the current drive and contains disk PCXTALK.

A FILE is a collection of related information. Before you can record any file on a disk, you must get DOS to format that disk. The DOS FORMAT command initializes a disk to a recording format acceptable to DOS, analyzes the disk for defective tracks, and initializes a main directory on the disk. This is the disk's ROOT DIRECTORY. A DISK DIRECTORY is like a table of contents. It contains the names of files and other information about the files. It can also contain the names of other directories, which are called SUBDIRECTORIES.

DOS can remember a DEFAULT DIRECTORY for each disk drive in your system. This is called the drive's CURRENT DIRECTORY. The current drive's current directory is the DOS current directory. If you give DOS a filename without specifying a directory, DOS searches for the file's entry in current directory. When DOS is started, it will use the root directory as the current directory of each drive. You can change the current directory, or find out what it is, by issuing the CHDIR (Change Directory) command.

An MS-DOS FILENAME consists of one to eight characters optionally followed by a dot optionally followed by an EXTENSION of up to three characters. The name SPEECH.TXT is an example. The following characters can be used in filenames and extensions:

A-Z 0-9 ! @ # $ % ^ & ' ( ) - { } _ ~

Any other character, such as a space or a comma, terminates the filename or extension. The GLOBAL FILENAME CHARACTERS ? and * are exceptions to this. The ? character in a filename or extension means that any (one) character can be in that position. The * character in a filename means that any character can be in that position and in the rest of the filename. The * character in an extension means that any character can be in that position and in the rest of the extension.

A FILE SPECIFIER consists of a drive specifier followed by a filename with optional dot and extension. If the file is on the disk in the current drive, the drive specifier may be omitted. If the file does not belong to the drive's current directory, the directory must be specified immediately before the filename.

Each time a Lanier data unit is to be sent, the PC operator must name the PC disk file which will receive the unit. If the name is already in the receiving disk's directory and you try to send data, you will be asked to pick one of these options: replace the old file's data, add the received data to it, or name a different receiving file.

Some DOS commands useful in our application are FORMAT, DIRECTORY, CHDIR, COPY, ERASE, RENAME, TYPE and MORE.

FORMAT prepares a disk for use by MS-DOS, destroying any data on the disk. You can specify a volume label to be written on the disk.

DIRECTORY (DIR) shows a disk's volume label and the name, size in bytes, and creation date and time of each file represented the disk's directory. A BYTE is eight bits and can hold 1 character.

CHDIR (CD) changes the specified drive's current directory to the one you name, if you name one; otherwise, it displays the current directory path.

COPY copies one or more files to the same or another directory. If a copy is to be in the same directory as its original, it must be given a different name.

ERASE deletes a file's entry from a disk directory and makes the file's space available.

RENAME changes the name of a file in a disk directory.

TYPE and MORE show the contents of a text file on a disk. For examples, 'DIR B:' shows the B drive disk directory, and 'TYPE README' and 'MORE < README' show the contents of text file README. MORE pauses after showing each screen and TYPE does not. To tell MORE to show the next screen, press the spacebar. To stop DIR, MORE or TYPE before they are finished, type:

Ctrl/Break

A PC disk must be formatted before MS-DOS can write on it. The FORMAT COMMAND DESTROYS ANY DATA ON THE DISK IT FORMATS, SO MAKE SURE THE DISK TO BE FORMATTED CONTAINS NO VALUABLE DATA, AND THAT ITS DRIVE IS THE TARGET DRIVE. If a disk is formatted, it will have a main directory. If you try to show the directory of an unformatted disk, you will be asked: 'Abort, Retry or Ignore?' (answer A). Thus the DIR command helps determine if a disk is formatted. To format a disk, put it in drive B and enter:

FORMAT/V B: Retrn

Here '/V' specifies the volume label option and 'B:' tells what drive holds the disk to be formatted (the target). When told to 'insert new diskette,' do so, if it is not inserted, and press the spacebar. When FORMAT is nearly done, you are asked to enter a VOLUME LABEL (up to 11 characters), such as ABC_XYZ_123, which is written on the disk. Enter one using characters acceptable in file names, but not dot. Then show the B drive directory by typing:

DIR B: Retrn

B.3 Use of Microstuf's Crosstalk XVI Communications Program

To use Crosstalk XVI, put the disk marked PCXTALK in the PC's A drive. If MS-DOS is not running, load and start it as explained in section B.1. If the MS-DOS prompt is not 'A>' type:

A: Retrn

To start Crosstalk type:

XTALK LANIER Retrn or XL Retrn

While Crosstalk is starting, a sign-on message and a communications message are displayed briefly.

Crosstalk shows just two screens: a status screen and a terminal screen. If you don't see the screen you want, switch screens by pressing Home.

The STATUS SCREEN shows the names of some commands which set Crosstalk options, and the current setting of each. Since only the first two letters of a command need be entered, they are shown in upper case.

The COMMAND LINE sometimes appears on the bottom line of the screen. You can get it on the status or terminal screen by pressing Esc. You also get it when you switch screens. It is in reverse video, with the prompt 'Command?' at the left. We will often just say 'Command?' when we mean the command prompt on the command line. To execute a command, get Command?, type the command, and press Retrn. If you type a command but decide not to execute it, backspace to Command? and press Retrn. This removes the prompt.

The TERMINAL SCREEN is useful when communicating with a another computer. The terminal screen's bottom line is its status line when it is not the command line. It shows capture and communications status. Capture is explained below. In our Lanier-to-PC system communications status is LOCAL, meaning direct connection (no modem or telephone).

If you see the terminal screen without the command line, what you type goes to the other computer. When the screen does not show the command line, Crosstalk pays no attention to what you type. Switching screens does not disrupt communications. Pressing End sends a break signal to the other computer. Crosstalk calls Esc, Home and End the ATTENTION, SWITCH and BREAK keys, respectively. The break key is rarely, if ever, used in Lanier-to-PC transfers.

If you press Esc, you get Command? without changing screens. If you press Home, you get Command? and the other screen.
Any Crosstalk command may be entered any time the command line appears. Use upper or lower case letters. Type at least the first two letters of the command name followed by other relevant items, separating items by spaces, and press Retrn. If the command needs more information, you will be asked for it. A common mistake is typing a command without getting the command line.

The bottom 10 lines of the status screen are used as a display window. The terminal screen's window, which the CLear command can clear, has a much larger display window. Characters received from the Lanier are shown on the terminal screen. Some commands which use display windows are HElp, DIr and TYpe. For example, for help with the DI command type:

HELP DIRECTORY Retrn or HE DI Retrn

To see the names of all Crosstalk commands, type:

HE Retrn

Before telling the Lanier to send data, we must tell Crosstalk to receive data and where to put it. For example:

CAPTURE B:UCSPEECH Retrn

says 'receive data and put it in file UCSPEECH to be listed in the current B drive directory.' To stop capturing data in this file type:

CA OFF Retrn.

You can exit Crosstalk without terminating PC communications by typing:

XD Retrn

which returns you to MS-DOS, where you can do MS-DOS commands. To return to Crosstalk, type:

XL Retrn

You can use the RUn command to exit Crosstalk without terminating communications, execute an MS-DOS command, and return to Crosstalk, like this:

RU RENAME XXX YYY Retrn

Use the QUit command to terminate PC communications and Crosstalk:

QU Retrn

B.4 Use of a Transfer Plan

We strongly recommend that if a person is to do many Lanier-to-PC transfers, he or she prepare a TRANSFER PLAN to be used with each session. This may take the form of a TRANSFER NAME TABLE (TNT), which has a row for each data unit to be sent. Each row gives the Lanier disk name, unit name, and type of unit to be sent; the receiving PC disk name and file name. For example:

LANIER WORD PROCESSOR			PERSONAL COMPUTER	
DISK	UNIT	TYPE	DISK	FILE
LD-B-001	TABLE-6	PAGE	DTICB-LD003	TABLE6
LD-W-012	REG3421/5	TEXT	LD-W-REG-03	RG3421-5.LEF

Here LD means 'Lanier to DMIMS.' Write a unique identifier on the external label of each Lanier or PC disk with a felt pen. No disk name is included in any Lanier sending or PC receiving command. Disks are rarely transmitted as a unit, since they ususally contain pages and texts which have different characteristics but all go into one PC file.

Another way to form a transfer plan is to print each relevan' Lanier disk index on a separate sheet of paper, staple the sheets together, and write in the blank spaces appropriate information, such as PC file names or margin settings, as it becomes available.

We also recommend displaying the first page of each text to be sent, and noting at the top of the screen the numbers of the columns marked L, R and H, and the value of V. These represent left margin, right margin, horizontal,

that is maximum line length (page width), and page length.

The margin values are important for preserving wraparound in the Q-One document produced by step 4 at the end of the whole process (see section 1.2 above). It is best if all pages in a text have exactly the same L and R values. If the R values vary, use the largest in step 2 (the LEFDIF step). It may be worthwhile to modify some Lanier pages before sending them.

The horizontal is important if line truncation is to be avoided. The Lanier typist could have used the right arrow key to pass through column R and put characters in the right margin. Such characters can be truncated in a Lanier-to-PC transfer if the value of H is too small. At least this has happened several times. The transfer operator can manually set the value of H, but when the Lanier reads a page for transfer, the H value may become that of the page.

The PC receives and displays Lanier formatting characters which make their lines appear longer than they would if printed.

If you are transferring a wide page, watch for truncated lines on the PC's terminal screen as the incoming characters are displayed. Note that if a line contains more than 80 characters, the 81st and following Lanier line characters are displayed on the next PC screen line, although one Lanier line is saved on disk as one PC line.

You can get a good idea of a Lanier page's margin settings by studying the received text lines (a) as displayed on the terminal screen or (b) as displayed by the DOS MORE command after the capture file is closed.

Although the Lanier specifies margins as column numbers, Q-One shows them as number of columns to the left of L or to the right of R. For example, if L is 11, R is 75 and H is 85, then each Q-One margin is 10 with 65 columns between. But if L is 13, R is 90 and H is 102, then each Q-One margin is 12 with 78 columns between.

A Lanier typist can produce completely empty lines by using the down arrow key to pass through empty screen lines and then typing text below them. Lines with hard returns (paragraph separators) at the left margin should be used instead of empty lines. The LEFDIF process can omit empty lines, or replace them with hard returns, or let the transfer operator decide which to do.

The LEF-to-DIF process is described in Appendix D.

APPENDIX C

STEP-BY-STEP PROCEDURE FOR FILE TRANSFERS FROM A LANIER WORD PROCESSOR TO A PERSONAL COMPUTER

Note: [LW] means 'at the Lanier' and [PC] means 'at the PC.'
Names of keys are underlined. We show PC commands with one optional space between parts and before Retrn. To get Crosstalk's command prompt (Command?), press Esc or switch screens by pressing Home.
A symbol like your_page_name stands for a name you pick.

1. [LW] Turn on the Lanier. Put the Lanier disk labelled No Problem TTY-ASCII Data Communications in the upper drive and close the door. Push the red load (boot) button. The TTY-ASCII program will be loaded and start executing. Remove the TTY-ASCII disk.

2. [LW] Put the communication table disk marked LCTABLE in the upper drive and close the door. Load table COMMUNITAB by typing:

FNC + L + COMMUNITAB + EXEC

Remove the table disk.

3. [LW] Go to offline editor mode and set the page width to 144 by typing:

FNC + E + EXEC
FRMT + H + 144 + EXEC
FNC + C + EXEC

The third command returns to communications mode. This step is a precaution against line truncation during transmission.

4. [PC] Turn on the PC. Put the system disk marked PCXTALK far into the A (upper) drive and close the door. Hold down Ctrl and Alt, and press Del. MS-DOS will be loaded and start executing. Enter date and time when prompted.

5. [PC] If any PC disk you plan to use in the session is not formatted, format it now. But first make sure the disk holds no valuable data. Put it in drive B and list its file directory information by typing:

DIR B: Retrn

If the disk is not formatted, a 'Disk error' message eventually appears. If a directory listing shows, make sure each file in each directory is worthless. THE FORMAT COMMAND DESTROYS ALL DATA ON THE DISK IT FORMATS, SO BEFORE USING IT, BE SURE IT IS OK TO DESTROY ALL DATA ON THAT DISK. If sure, type:

FORMAT/V B: Retrn

When told to 'insert new diskette,' do it if not done and press the spacebar. When asked for a volume label, enter up to 11 of the characters acceptable in file names, but not dot. Repeat this step for each unformatted disk.

6. [PC] Make sure that drive A is the current drive.

7. [PC] Start the Crosstalk XVI communication program by typing:

XTALK LANIER Retrn or XL Retrn

Get the status screen. Look at it. Then get the terminal screen.

8. [PC] From your transfer plan determine which file in which directory on which PC disk will capture (receive) the data unit (page or text) to be sent next. If the disk is not in drive B, put it there.

9. [PC] Get the terminal screen, get 'Command?', and type:

DI B: Retrn

Make sure the capture file's name is not on the directory list (see step 13).

10. [LW] From your transfer plan determine which page or text on which Lanier disk to send next. If the disk is not in the upper drive, put it there.

11. [LW] If unsure of the data unit's name or of its presence on the disk, have the offline editor show the disk's index (table of contents) by typing:

FNC + E + EXEC
FILE + I + EXEC
FRMT + H + 144 + EXEC
FNC + C + EXEC

The last two commands reset the page width and return to communications mode.

12. [LW] Type one of the following (do not yet press EXEC):

CMC + T + P + your_page_name
CMC + T + T + your_text_name
CMC + T + D

where T stands for transmit, P for page, T for text, D (rarely used) for disk.

13. [PC] Get 'Command?' and tell Crosstalk to take data by typing:

CA B:your_capture_file_name Retrn

where your_capture_file_name is the file specifier of the receiving file. If it exists, you'll be asked to append to it, write over it, or pick a new name.

14. [LW] Start transmitting by pressing:

EXEC

To stop transmitting in an emergency type CNTRL/CNCL and FNC + C + EXEC

15. [PC] When the transmission is done, get 'Command?' and stop capturing:

CA OFF Retrn

This saves the capture file. If you have more to send, go back to step 8.

16. [PC] To view any capture file, get 'Command?' and type:

RU MB your_capture_file_name Retrn

which causes a return to MS-DOS, a showing of the capture file by MORE, and a return to Crosstalk. To tell MORE to show the next screen, press spacebar.

17. [PC] If you wish to capture two or more data units in the same capture file, you can leave it open (omit step 15) until the last data unit is received and omit steps 9 and 13 when sending the second and following units. But between data unit transfers, accidentally or otherwise pressing Lanier keys may cause characters to be received in the capture file. These may produce unwanted effects. So be careful.

18. [PC] Stop Crosstalk by getting 'Command?' and typing:

QU Retrn

This terminates PC communicaions and brings back the MS-DOS prompt. Now you can use MS-DOS commands, such as DIR and MORE.

19. [LW] and [PC] Remove your disks from the machines. Collect all your disks and other items.

20. [LW] and [PC] Put disks TTY-ASCII, LCTABLE and PCXTALK in their box.

21. If no one is ready to use the Lanier-to-PC system, tell the scheduler and the next group scheduled that you are done. Turn off the Lanier, the PC, and all other Lanier-to-PC system equipment. See that the system disk box is put in its safe storage place. Take all your items with you when you leave.

THANKS

APPENDIX D

PROCEDURE FOR DATA CONVERSION FROM LANIER EXTERNAL FORMAT TO DOCUMENT INTERCHANGE FORMAT

Appendices B and C describe a procedure used to transfer text files from the Lanier word processor to an IBM or Sperry PC. The PC file received from the Lanier is in Lanier external format (LEF). This appendix describes a procedure for converting a PC file in LEF to a PC file in Document Interchange Format (DIF). The program LEFDIF does the conversion processing. It can be done in an IBM or Sperry PC or in a Zenith 248 PC. For information on how to start a Zenith 248 PC see section E.1.

Section D.1 tells how to use program LEFDIF. Section D.2, which is for someone modifying the LEFDIF program, tells how to compile and link LEFDIF.

D.1 How to Use Processor LEFDIF.

Section D.1.1 covers the LEFDIF command and some DOS commands done before executing it. Section D.1.2 covers parameters which LEFDIF requests from the user.

D.1.1 The LEFDIF Command and Some DOS Commands Done before Executing It.

Program LEFDIF.EXE does all LEF-to-DIF format conversion processing. LEFDIF.EXE is on the PC disk called PCXTALK used in the Sperry PC's A drive in the Lanier-to-PC file transfers described in Appendix C. The LEF-to-DIF conversion can be done in the Sperry PC after a Lanier-to-PC transfer. For example, assuming that the current drive and directory are the A drive and its root, you can convert LEF file MEMO.LEF on drive B to a DIF file on drive B by typing:

LEFDIF B:MEMO.LEF B:MEMO.DIF Retrn.

This causes LEFDIF to input (read) MEMO.LEF and output (write) MEMO.DIF. In general, the LEFDIF input and output file names may have any extension or none.

Since we have only one Lanier-to-PC system, we may want to do the LEF-to-DIF conversion on another PC.

There are several ways to do a LEF-to-DIF conversion on a Zenith 248 PC. We will describe one way, continuing the example begun above. We will use only two directories: the root directory of the disk in the Zenith's A drive and the root directory of the disk in its C drive. If processor LEFDIF.EXE is not in the C drive's root directory, we will put it there by copying it from a disk in the Zenith's A drive. Disk PCXTALK contains LEFDIF.EXE in its root directory. To do the copy type:

```
A:  Return
CD  \ Return
C:  Return
CD  \ Return
COPY  A:LEFDIF.EXE Return
```

where CD means 'change directory.' The first two lines make the A drive the current drive and make the A drive root (\) directory the current A drive directory. The next two lines make the C drive the current drive (for the rest of the example) and make the C drive root (\) directory the current C drive directory. The last line copies processor LEFDIF.EXE to that directory. The copy's name is the same as the original's, since no new name is specified.

Now we put a disk containing MEMO.LEF in the A drive and type:

LEFDIF A:MEMO.LEF A:MEMO.DIF Return

Then we copy the resulting DIF file to the C drive's root directory by typing:

COPY A:MEMO.DIF Return

This puts the DIF file in a good position for transfer to a DMINS node (see Appendix E). If the A drive root has several DIF files, you can copy them all to the C drive's root by typing:

COPY A:*.DIF Return

where the global character * matches any filename.

D.1.2 Parameters Which LEFDIF Requests from the User.

When processor LEFDIF starts, it asks the operator for the values of certain parameters which the Lanier does not communicate to the PC. Following are the abbreviations, names and definitions of these and some others:

LM	Left Margin	The number of the leftmost column of the body text.
RM	Right Margin	The number of the rightmost column of the body text.
PW	Page Width	The maximum number of characters per line.
PL	Page Length	The maximum number of lines to be printed per page.
TM	Top Margin	The number of the topmost line on which body text can be printed.
BM	Bottom Margin	The number of the bottommost line on which body text can be printed.
HS	Horizontal Spacing	The number of characters per inch.
VS	Vertical Spacing	The number of lines per inch.
TW	Text Width	The maximum number of characters in one body text line.
TL	Text Length	The maximum number of body text lines on one page.
BS	Body Start	The number of lines from the top edge of the paper to the first body text line.
BE	Body End	The number of lines from the bottom edge of the paper to the last body text line.

The first eight definitions are essentially from the DIF standard, the next two are derivable from the first eight, while the last two are from Q-One. In specifying TM and BM the line at the top edge of the paper is line 1. Lanier pages do not have headers and footers. For example, if you want one-inch top

and bottom margins and VS is 6 and PL is 66, then set TM to 7 and BM to 60. This will cause the DIF-to-QIF converter to set BS and BE to ?.

Following are examples:

Type	LM	RM	PW	PL	TM	BM	HS	VS	TW	TL	BS	BM
A	1	80	85	66	7	60	10	6	80	54	7	7
B	11	75	85	66	7	60	10	6	65	54	7	7
C	12	96	96	66	7	60	12	6	85	54	7	7
D	13	90	102	88	9	80	12	8	78	72	9	9

These values assume 8-1/2 inch by 11 inch paper. This amounts to 85 or 102 columns, depending on the value of HS (10 or 12). If a Lanier page is recalled, its LM, RM, PW and PL values can be obtained visually from the top screen line as L, R, H and V. The Lanier does not send to the Sperry PC values for LM, RM, PW, PL, TM or BM. For LM it sends at the start of each line just enough spaces to make the next character go into the LM column. It also sends a DEL (7F) character after each page.

LEFDIF requires the first eight values: LM, RM, PW, PL, TM, BM, HS and VS. LEFDIF first shows you the standard settings:

```
  1  left margin          7 top margin
 80  right margin        60  bottom margin
 85  page width          10  horizontal spacing
 66  page length          6  vertical spacing
```

and asks if you want to use them. If you do, LEFDIF proceeds to process the input file. Otherwise, it prompts you for each of the first six values and uses 10 and 6 for HS and VS.

LEFDIF checks the LEF file's left margin, and displays the number of each line whose left margin is less than the one you specified. If it finds any such lines, LEFDIF terminates to allow you to fix the lines or decide on a different left margin, and then run LEFDIF again.

If the left margin is OK, LEFDIF reads the LEF file and produces the DIF file. If a set of nonstandard settings is used frequently, time can be saved by putting them, one per line, in a RESPONSE FILE and redirecting LEFDIF's questions to that file. For example, if the margins are standard except for left and right margins of 11 and 70, the response file would contain these eight lines:

```
N Return
11 Return
70 Return
85 Return
66 Return
7 Return
60 Return
Y Return
```

where N answers 'Use standard settings?', and Y answers 'OK?'. For example, if the input file is NARROW.LEF and the response file is LDRESP70, type:

```
LEFDIF  A:NARROW.LEF  A:NARROW.DIF  <  LDRESP70
```

and LEFDIF will complete without more typing (unless it detects an error).

A response file is an ordinary text file. You can build one using the COPY command. For example, type:

```
COPY  CON:  LDRESP70
N Return
11 Return
70 Return
85 Return
66 Return
7 Return
60 Return
Y Return
F6 Return
```

where CON:, meaning 'the console device' (keyboard), specifies the input to the copier, and the last line uses a function key to end copying. This is a simple way to build a small file, but you cannot change a mistake in a line after you have entered that line.

After the LEF-to-DIF conversion, the MS-DOS commands MORE (if available) and TYPE will display DIF files at 80 characters per line, showing control characters as special display symbols (see Figure 4 in Appendix A). For example, to display MEMO.DIF type one of the following:

```
MORE  <  MEMO.DIF Retrn
TYPE  MEMO.DIF Retrn
```

The information given in this section is enough for using LEFDIF.

D.2 How to Compile and Link LEFDIF.

LEFDIF is written in the C programming language and is compiled and linked using a Microsoft C Compiler and Linker.

The COMPILATION has two phases, which may be depicted thus:

```
LEFDIF.C  ->  phase 1  ->  LEFDIF.Q  ->  phase 2  ->  LEFDIF.OBJ
```

The file input to phase 1, called the 'source file,' must have name extension C. Each source file, such as LEFDIF.C, which refers to a standard library function must contain near its beginning the line:

```
#include  <stdio.h>
```

File STDIO.H contains certain things required by such references. Phase 1 outputs an intermediate 'quad' file with name extension Q. Phase 2 produces an object file with name extension OBJ. This file is input to a linker, program LINK.EXE, which produces an executable program, in this case LEFDIF.EXE.
LINKING requires two special files, whose names depend on the memory model to

be used, which is indicated by the letter S, P, D or L. We use the S (small) model, for which the special files are CS.OBJ and MCS.LIB. CS.OBJ defines the program's execution entry and exit points. It must be specified first in the link command. MCS.LIB is a library file defining runtime and input/output library functions.

Figure 5 shows the names of the files required for compiling and linking program LEFDIF, and the the commands for doing so. Messages resulting from the commands are also shown. Note that file name extensions are not required.

```
C>dir
 Volume in drive A is DISK 01
 Directory of  A:\msc

.            <DIR>       9-12-86   6:14p
..           <DIR>       9-12-86   6:14p
STDIO    H       1895  11-22-83  11:08a
MC1      EXE    64896   4-09-84   3:52p
MC2      EXE    65664   4-09-84   4:09p
CS       OBJ     1003  11-22-83  12:32p
MCS      LIB    79360   4-06-84   8:49p
LINK     EXE    39680  12-09-83  12:17p
LEFDIF   C       8069   9-12-86   7:16p
LEFDIF   OBJ     5050  11-06-86  12:19p
LEFDIF   EXE    14762  11-06-86  12:19p

C>mc1  lefdif
Microsoft C Compiler (Phase 1) V2.03
Copyright (C) 1983 by Lattice, Inc./Lifeboat Associates

C>mc2  lefdif
Microsoft C Compiler (Phase 2) V2.03
Copyright (C) 1983 by Lattice, Inc./Lifeboat Associates
Module size P=085F D=0396

C>link  cs + lefdif, lefdif, nul, mcs

C>
```

Figure 5. Files Required for Compiling and Linking Program LEFDIF and the Commands Used to Do So.

APPENDIX E
ZSTEM PROCEDURES FOR FILE TRANSFERS BETWEEN A ZENITH 248 PERSONAL COMPUTER AND A DLA DMINS COMPUTER

This appendix tells how to do a Kermit file transfer from a CO-LAN-connected (see section 2.3.1) Zenith 248 Personal Computer (PC) to a DLA DMINS computer using Zenith's ZSTEMpc communications program, which we will call ZSTEM. We assume that you know how to start the ZSTEM program in the Zenith you use, how to connect with the DLA DMINS computer, and how to bring up the DMINS main menu.

Section E.1 tells how to start a Zenith 248 PC and use it to log into the DTIC DMINS. Section E.2 describes the ZENITH's ZSTEM communications program. Section E.3 covers Kermit file transfers between a Zenith 248 PC and a DLA DMINS. Section E.4 gives examples of such file transfers.

E.1 How to Start a Zenith 248 PC and Log into the DTIC DMINS.

Zenith 248 PCs are standard components of DLA DMINS systems. Each contains a floppy disk drive and a hard disk drive. These are the A and C drives, respectively. Files can be transferred to or from the root directory or a subdirectory of either drive.

In the DTIC DMINS a Zenith 248 PC and its ALPS P2000G or other printer are each typically connected by a power cord to an outlet in a nearby power strip on the floor or on the desk or table holding the PC. The PC and printer each has its own on/off switch which are often left on when the power strip is switched off. If these are off, switch them on. Switch on the power strip. This will cause the PC to do some initialization and then present the prompt:

```
C:\>
```

which says that the C drive (C:) is the current drive, that the root directory (\) is the current directory, and that MS-DOS is waiting (>) for a command from the user.

To communicate with the DMINS, we use a copy of the ZSTEM.COM communications program which has been configured for such communication. "Configured" here means that port number, baud rate and other parameters have been set to appropriate values. For example, the baud rate is set to 9600. We'll assume that the configured copy is in the root directory, that this is the current directory, and that ZSTEM.COM has been renamed DMINS.COM. (If this is not the case, find out the name and location of the configured copy from someone who knows.)

To start communicating type:

```
DMINS
```

and get a response like:

```
ZSTEMpc: VT100 SMART TERMINAL EMULATOR (V2.4P1)
Copyright (C) KEA Systems Ltd 1984,1985

                RESTRICTED RIGHTS LEGEND
                ------------------------
     'Use, duplication, or disclosure by the . . . .'

ORIGIN: NODE cndk1 MODULE 52 PORT 3

WELCOME TO CAMERON CO-LAN NODE 1

DESTINATION:
```

where we have shortened the legend. 'CO-LAN' refers to a Bell Atlantic Central Office based Local Area Network which allows customers to share information and resources by connecting terminals, workstations and computers.

Now type:

```
? Return
```

and see:

```
            ?:          CAMERON NODE 1
AREA:       703:        Northern Virginia
EXCH:       106:        Cameron Node 0
            116:        Cameron Node 1
LOCAL:      1000:       DTIC DMINS HOST 1

DESTINATION:
```

Now type:

```
116/1000 Return
```

where 1000 alone is often sufficient, and see something like this:

```
GOULD UTX/32 2.0U01 (CSD-URB) (dticg1) (tty92)

login:
```

After you enter your login and password, the DMINS main menu is displayed:

```
    DEFENSE LOGISTICS AGENCY
DATA SYSTEMS AUTOMATION CENTER
------------------------------
1     Change Your Password
2     Q-Office (Word Proc / Mail)
3     Data Base
4     Electronic Mail Forwarding
5     Business Graphics
6     20/20 Spreadsheet
7     Lock this Terminal
8     Phonebook
9     Bourne Shell
10    C Shell
11    CAI Courses
12    INFO


Enter option or 'x' to logout
```

E.2 Use of the ZSTEM Communications Program

Actually, there are two versions of ZSTEM: the text version ZSTEM.COM, and the graphics version ZSTEMG.COM.

ZSTEM provides two distinct modes of operation:

> **terminal mode** (or emulation mode), in which your computer functions like a terminal.
>
> **command mode** (or alternate mode), in which you can enter ZSTEM commands to modify the configuration, initiate file transfers, or access other extended features.

ZSTEM starts in emulation mode. You can switch to command mode at any time by pressing Alt, which causes the prompt:

> ZSTEM?

to be displayed on the 25th line. You can then enter ZSTEM commands. You can display a list of ZSTEM commands by typing:

> HELP Return

Many ZSTEM commands prompt you to select a subcommand or a value.

When in ZSTEM command mode, remember:

1. Any prompt that ends with a question mark (?) requires you to type a line of text and then press Return.

2. Any prompt that ends with a colon (:) requests that you type one character and press Return. The default, shown in square brackets, is chosen by just pressing Return.

3. Any prompt that ends with three periods (...) requires a single keypress only.

4. If an error occurs during command processing, an error message is displayed on the 25th line.

5. You need enter only enough of the initial part of a command name to make it unique. For example, the list of ZSTEM commands can be displayed by entering HE at the ZSTEM? prompt.

6. You can type ahead at the following command prompts:

```
ZSTEM?
Send, Get, Receive, Directory, Finish, or Command:
Select item?
```

The second line is ZSTEM Kermit's main menu. The third is from the CONFIGURE program. Typing ahead saves time by eliminating prompts. Instead of entering the command and then answering prompts for subcommands or values, pressing Return after each, you can enter the respective responses all in one line, separated only by a space. For example, at the ZSTEM? prompt you could enter:

```
BAUD REMOTE 9600
```

or

```
B R 9600
```

To embed a Return in a type ahead use the null string ''. You can enter multiple commands in one type ahead if you separate them by null strings.

To return to terminal mode, press Return at the ZSTEM? prompt. To terminate ZSTEM and return to MS-DOS command mode, enter EXIT at the ZSTEM? prompt. EXIT may or may not hang up, that is disconnect communications, depending on the value of a setting in the Zenith's configuration. If 'hangup and exit' is not set, enter HANGUP before entering EXIT. If you are finished with the DMINS, make sure disconnection takes place.

E.3 Kermit File Transfers between a Zenith 248 PC and a DLA DMINS.

To start a Kermit file transfer between a Zenith 248 PC and a DLA DMINS computer, two Kermit processors must be running: one in the PC and another in the DMINS minicomputer. A Kermit processor is a communications processor which implements the Kermit communications protocol (see Appendix G). The ZSTEM communications program includes a Kermit processor. The DMINS computer has a Unix Kermit processor, which is a remote Kermit in server (host) mode.

An example of the whole command sequence for a Zenith to DMINS transfer and one for a DMINS to Zenith transfer are shown in sections E.4 as examples A and B, respectively.

To send a PC file to a DMINS, get the DMINS main menu and select C Shell (item 10). This puts you in the Unix C Shell environment, where the prompt character is the percent sign (%). To start the Unix Kermit processor type:

kermit -r Return

using lower case letters. The Unix Kermit will wait for characters sent from the Zenith. Immediately press Alt to get the ZSTEM? prompt and type:

kermit

The Zenith Kermit starts and displays the following prompt on the 25th line:

Send, Get, Receive, Directory, Finish, or Command:

Enter the first or more letters of the subcommand you choose or press Return to get back the ZSTEM? prompt. In our case, type:

s Return

The Zstem Kermit Send subcommand causes Kermit to send one or more local (PC) files to the remote system. Send displays the following prompt on the 25th line:

Local file?

You may specify a single file or specify multiple files by using one or more wildcard characters (? and *) in the file specification. If your local file name is invalid or the file does not exist, Send reprompts you. To abort Send, type no specification and press Return. If your file specification is OK, Send prompts:

Rmt file [local]:

If you want the received file to have the same name as the sent file, just Press Return. Otherwise, enter the new name. If the remote file name is invalid, Send reprompts you for the local file name. Otherwise, ZSTEM starts sending the file immediately.

If the Send command is entered too late, the DMINS Kermit procesor will stop and have to be restarted. To do this, get the ZSTEM? prompt, press Return to get the Unix prompt, and reenter the Kermit receive command.

While the file transfer is active, ZSTEM shows statistics on the 25th line.

The message 'Too many errors: command aborted' that an error retry limit was exceeded or the remote Kermit sent an error packet. If it appears, restart the DMINS kermit processor.

When ZSTEM finishes sending a file, it logs the message:

File send completed

with packet, byte, error and retry counts.

If ZSTEM cannot finish sending a file, 'File send ABORTED' is displayed. You can abort a file transfer by typing Ctrl/X, which aborts the current file, or Ctrl/Z, which aborts the rest of the files in a group transfer, or by presssing Alt, which gets Kermit's main menu without notifying the remote

Kermit.

When all specified files have been sent, ZSTEM Kermit's main menu. After sending is done, ZSTEM Kermit's main menu is displayed. You can then transfer more files or stop the remote Kermit processor by using the Finish option, in which case the ZSTEM? prompt is displayed, or just press Return to get the ZSTEM? prompt. At this prompt, you can get the Unix prompt (%), by just pressing Return. To show the names of the files in the current Unix directory enter:

```
ls  -l  -t  |  more
```

which lists their names and other information a screen at a time in reverse creation time order (newer before older). You can examine a transferred file using one of the following Unix processors:

```
ex        line editor
vi        visual editor
more      shower of a page at a time
od -a     ascii character dump
```

For DIF files, you may well prefer to convert each to a Q-One document file and examine the document. The procedure for such conversions (DIF-to-QIF) is described in section 5.3 of this report.

To return from the Unix C Shell to the DMINS main menu, press Ctrl/LD.

The rest of this section briefly describes the Kermit subcommands other than Send. The next section gives two exxamples of Kermit file transfer command sequences.

The ZSTEM Kermit Receive subcommand is used after the remote Kermit server has initiated a send, specifying the file(s) to be transferred. Example B in section E.4 shows this.

The ZSTEM Kermit Get subcommand is used when the remote Kermit is in server mode, as in the DMINS. It requests the remote Kermit to send one or more files to ZSTEM. It is similar to the Kermit Send command, but the receiver specifies the file(s) to be transferred. We recommend using receive, since we have been unable to make Get work yet.

The ZSTEM Kermit Directory subcommand displays directory information on any drive or subdirectory on your Zenith PC system. It displays the following prompt on the 25th line:

```
File specification?
```

Enter a drive designator, a wildcard specification, and/or a directory path name; or press Return to get the Kermit main menu. This subcommand assumes a wildcard if you do not specify a file name or file name extension, so that the following are equivalent:

```
C:*.*
C:*
C:
```

But if you enter a period without an extension, only files with no extension are selected.

The ZSTEM Kermit Finish subcommand terminates the remote system Kermit when you have finished transferring files. It transmits a finish packet to the remote system, causing it to terminate Kermit and return to normal terminal communications. When this happens, the ZSTEM? prompt is displayed, and you can press Return to return to terminal emulation. If the remote Kermit does not respond to the finish packet, perhaps having timed out, ZSTEM displays the error message:

Unable to shut down server

and the ZSTEM? prompt.

The ZSTEM Kermit Command subcommand sends commands to MS-DOS or to a remote Kermit. It prompts:

Local (DOS), Remote, or Kermit

We will consider only an example of use of the Local option in a type ahead. Suppose at the ZSTEM? prompt you enter:

L C L DIR

This causes ZSTEM to exit to MS-DOS, perform the MS-DOS DIR command, and return to the Kermit main menu. In this case the MS-DOS command displays information in the current directory, giving more than does the ZSTEM Kermit Directory subcommand.

When finished with ZSTEM Kermit, press Return until you get the Unix prompt. Then type Ctrl/D to get the DMINS main menu. Select EXIT (item x) and get the DESTINATION prompt. Then press Alt to get the ZSTEM? prompt and type:

ha y Return

to hang up and return to MS-DOS command mode.

E.4 Examples of File Transfers Using the DMINS Unix and ZSTEM Kermit Processors.

EXAMPLE A. Use the DMINS Unix and ZSTEM Kermit processors to send a file called NEWFOLKS from a Zenith 248 PC to a DMINS computer. Then get the DMINS main menu.

1. Get the DMINS main menu.

2. Select C Shell (item 10) and get the Unix prompt (%).

3. Type 'kermit -r Return'.

4. Press Alt to get the ZSTEM? prompt.

5. Type 'kermit send newfolks Return'.

6. Press Return when prompted for the remote file name.

7. Press Return to get the Zstem? prompt after the transfer is done.

8. Press Return again to get terminal mode and the Unix prompt.

9. Press Ctrl/D to get the DMINS main menu.

After step 6 a Kermit statistics line is displayed and updated. When the transfer is done, a 'File send completed' message with statistics is displayed. Then the Zstem Kermit main menu is displayed.

Note that in PC file names, letters are upper case but may be typed in upper or lower case, while in Unix file names, the case of a letter makes a difference, and lower case letters are usually used.

EXAMPLE B. Use the DMINS Unix and ZSTEM Kermit processors to send a file called AGENDA from a DMINS computer to a Zenith 248 PC, supposing AGENDA to be in subdirectory IOM immediately under the current directory. Then get the Zstem? prompt.

1. Get the DMINS main menu.

2. Select C Shell (item 10) and get the Unix prompt (%).

3. Type 'kermit -s IOM/AGENDA Return'.

4. Press Alt to get the ZSTEM? prompt.

5. Type 'kermit receive Return'.

6. Press Return or enter a file specifier when prompted for the local file name.

7. Type 'C L DIR Return' to see the current Zenith directory.

8. Press Return to get the Zstem? prompt after the transfer is done.

After step 6 a Kermit statistics line is displayed and updated. When the transfer is done, a 'File receive completed' message with statistics is displayed. Then the Kermit main menu is displayed.

Note that in PC file names, letters are upper case but may be typed in upper or lower case, while in Unix file names, the case of a letter makes a difference, and lower case letters are usually used.

APPENDIX F

CROSSTALK XVI PROCEDURES FOR FILE TRANSFERS BETWEEN A SPERRY UNIVAC PERSONAL COMPUTER AND A DLA DMINS COMPUTER

F.1 Kermit File Transfers Using the DMINS Unix and Crosstalk XVI Kermit Processors.

This appendix tells how to do a Kermit file transfer from an IBM or Sperry personal computer (PC) to a DLA DMINS computer using the Microstuf's Crosstalk XVI communications program. It was used for testing by dialing up a DSAC (Columbus, OH) DMINS long before DTIC's DMINS was installed.

To start a Kermit file transfer from a PC to a DMINS computer, two Kermit processors must be running: one in the PC and another in the DMINS computer. A Kermit processor is a communications program which implements the Kermit communications protocol (see Appendix G). The Crosstalk XVI communications program includes a Kermit processor. The DMINS Unix also has a Kermit processor, which is a remote Kermit in server mode.

The Crosstalk XVI communications program is described in section B.3 of Appendix B, which deals with sending files from a Lanier word processor to a directly connected PC. Crosstalk calls this a 'local' connection. Here we consider a PC and a remote computer connected through modems and a telephone line. The command we use to start Crosstalk in the local connection setting is 'XTALK LANIER'. To start Crosstalk in a DMINS session we type:

```
XTALK  DMINS Retrn
```

When asked to enter the character to be used for EOL character, type Ctrl/J (hold down Ctrl and press J). Kermit settings will be shown and should be:

```
Packet size      94
Timeout          10
End of line      ^J
Control Quote    #
Binary Quote     &
Mode             Text-only
```

If the ones shown are not these, set them by entering:

```
KE  P  94
KE  T  10
KE  E  Ctrl/J
KE  C  #
KE  B  &
KE  M  T
KE  L
```

where the last command lists the current Kermit settings. Although Kermit can be used to transfer binary files, that is, files containing eight data bits per character, the present description is limited to transfers of ASCII text files composed of seven-bit characters. When the Crosstalk Kermit mode is

text-only, binary file transfers cannot be done, and the binary quote character is irrelevant.

The procedure for dialing the DMINS telephone number and making the connection depends on the PC communications hardware, the modem, and the telephone equipment. Once the connection is established, the procedure is relatively simple.

We assume that you know how to start the Crosstalk program in the PC you will use, how to connect with the DMINS computer, and how to bring up the DMINS main menu.

To send a PC file to a DMINS, get the Crosstalk terminal screen by pressing Home once or twice, get the DMINS main menu and select C Shell (item 10). This puts you in the Unix C Shell environment, where the prompt character is the percent sign (%). To start the Unix Kermit processor simply type:

```
kermit -r Return
```

using lower case letters. Then press Esc to get the Crosstalk COMMAND? prompt and enter a Kermit send command, with a file specifier, which may include wildcards. The transfer will begin, proceed, and end automatically. A transmission status table will appear on the PC's screen. When the transfer is done, the status table will say '100% complete.'

The Crosstalk procedure for sending a DMINS file to a PC is similar to the one just described.

Examples of Kermit file transfer command sequences are given on the following pages.

F.2 Examples of File Transfers Using the DMINS Unix and a Crosstalk XVI Kermit Processors

EXAMPLE A. Using the DMINS Unix and Crosstalk Kermit processors to send a file called NEWFOLKS from an IBM or Sperry PC to the DMINS computer.

1. Get the DMINS main menu.
2. Select C Shell (item 10) and get the Unix prompt (%).
3. Type 'kermit -r Return'.
4. Press Esc to get the Crosstalk COMMAND? prompt.
5. Type 'kermit send newfolks Return'.
6. When the transmission is done, press Retrn.

Note that in PC file names, letters are upper case but may be typed in upper or lower case, while in Unix file names, the case of a letter makes a difference, and lower case letters are usually used.

After step 5 a transmission status display like the following appears:

Transmitting A:NEWFOLKS

Block #	% complete	Consec. errors	Total errors
7	20%	none	2

PROTOCOL TRANSFER UNDERWAY -- Press Esc to cancel

When the transmission is 100% complete, Crosstalk's command line says:

File transmission complete. Press ENTER:

After pressing Retrn, you can press Home to get the terminal screen, and then interact with Unix.

The presence of a count in either of the 'errors' cells in the transmission status display does not imply that the received file differs from the sender's copy, since such errors can be overcome by retransmission. The '100 % complete' message implies that the received file results entirely from packets received with no error detected.

One could send all the files in the current PC directory to the DMINS computer by typing 'kermit send *.* Retrn' in step 5. A series of file transmissions started this way could go on for hours.

EXAMPLE B. Using the DMINS Unix and Crosstalk Kermit processors to send a file called AGENDA from the DMINS computer to an IBM or Sperry PC, supposing AGENDA is in subdirectory IOM immediately under the current Unix directory.

1. Get the DMINS main menu.

2. Select C Shell (item 10) and get the Unix prompt (%).

3. Type 'kermit -s IOM/AGENDA Return'.

4. Press Esc to get the Crosstalk COMMAND? prompt.

5. Type 'kermit receive Return'.

6. When the transmission is done, press Retrn.

Note that in PC file names, letters are upper case but may be typed in upper or lower case, while in Unix file names, the case of a letter makes a difference, and lower case letters are usually used.

After step 5, transmission status is displayed as in Example A. The completion indications and the procedure after completion are the same as in Example A.

APPENDIX G

DESCRIPTION OF THE KERMIT FILE TRANSFER PROTOCOL

This appendix briefly describes the Kermit file transfer protocol. This information in it is not required for transferring files by using the steps given above, but is presented for anyone wanting a brief overview of Kermit. It is based on the two-part article "Kermit: A File-Transfer Protocol for Universities," by Frank Da Cruz and Bill Catchings, in the June and July 1984 issues of Byte.

KERMIT is a portable file-transfer communication protocol. Here PROTOCOL means a set of rules for forming and transmitting packets, carried out by programs that embody those rules. Programs implementing Kermit are written in various languages and run on various computers.

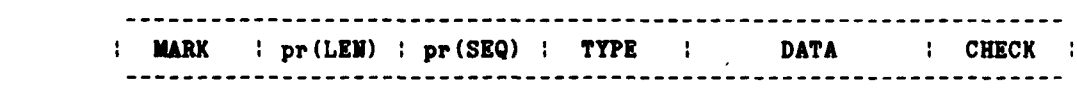
In a Kermit file transfer communication takes place over ordinary terminal connections. Communication is asynchronous and half duplex. All information is sent in packets. The packet length is variable, but the maximum is 96 characters. Packets are sent in alternate directions, and each packet requires a reply.

All transmission is in ASCII (see Appendix A above). Unprintable ASCII characters are prefixed with a special character and then converted to printable characters during transmission, as explained below. A single ASCII control character, normally SOH (1) is used to mark the beginning of a packet. Lines in text files are terminated during transmission with prefixed carriage return/linefeed sequences, which are transparent to the protocol and may appear anywhere in the data in the packet.

Only a file's name and contents are transmitted. File attributes are not.

The simplicity and generality of the protocol allow Kermit on any machine to communicate with Kermit on any other machine: microcomputer with mainframe, microcomputer with microcomputer, mainframe with mainframe (not to mention minicomputers). The back-and-forth exchange of packets keeps the two sides synchronized, while the communications hardware operates asynchronously.

Each packet starts with the mark character and contains a character specifying each of the following: packet length, packet sequence number, packet type, and checksum. Data (if any) goes between the type and the checksum. The length and sequence values are increased by 32 to ensure that they are printable. The Kermit packet format looks like this:



where the pr function converts a number in the range 0 to 94 to a printable ASCII character by adding 32.

The packet length (LEN), which does not include the MARK and LEN fields, can be up to 94 characters. Sender and receiver have to agree on the maximum length. The packet sequence number (SEQ) goes from 0 to 63, and then recycles

to 0. The sender increases it by 1 each time the receiver acknowledges a packet reception.

Commonly used packet types are:

D	Data
Y	Acknowledge
N	Negative Acknowledge
S	Send Initiate
R	Receive Initiate
B	Break Transmission (EOT)
F	File Header
Z	End of File
E	Error

CHECK may be one, two or three characters. It is based on all the characters in the packet, except MARK and CHECK. The method of computing CHECK depends on the implementation. One method is to get the sum of the ASCII codes, isolate the low-order six bits, add to this the value of the next two bits, and apply the pr function to the low-order six bits of the sum.

Sender and receiver use the same CHECK computation method. The receiver computes CHECK from the contents of the packet received. If it agrees with the one sent, the receiver sends back an acknowledge (ACK) packet. If not, the receiver sends a negative acknowledge (NAK) packet. If the sender receives a NAK packet, or does not receive an ACK packet soon enough, the unacknowledged packet is sent again. If the number of attempts to send a packet exceeds the limit, the sender terminates the transfer.

Kermit uses PREFIXING to handle unprintable characters and to do rudimentary data compression. In order to make each data character in the packet printable, Kermit prefixes, or quotes, each unprintable character by transforming it into a printable one and preceding it with by a special prefix character, normally #. The transformation is done by altering the seventh bit. (This is not the pr function, since the value of the seventh bit is 64, and in one case it is subtracted.) Thus Control-A (1) becomes #A and carriage return (13) becomes #M. Where the prefix character # is part of the text, it is prefixed with itself: ##. The unprintable characters are the control characters and DEL (0 through 31 and 127). Printable characters are not transformed. The Kermit processor receiving the file performs the reverse transformation.

DATA COMPRESSION is done by replacing repeated characters by the repeat count prefix character ~ (tilde), followed by a repeat count character, followed by the character repeated. The count is encoded printably by using the pr function. For example, ~)A represents a series of 93 letter As, and ~H#B represents a series of 40 Control-Bs. Where the repeat count prefix character is part of the text, it is prefixed with #. Thus ~W#~ stands for 55 tildes.

Consider an errorless file transfer between two Kermit processors, Kermit 1 and Kermit 2. Kermit 1 initiates the transfer by sending an S or R packet specifying what it wants to do (send or receive) and some variable values, such as maximum packet size and end of line character. Kermit 2 sends an ACK packet specifying its variable values. Kermit 1 sends an F packet naming the file to be transferred. Kermit 2 acknowledges. Kermit 1 sends the contents

of the file in D packets. Kermit 2 acknowledges each. Kermit 1 sends a Z and a B packet. Kermit 2 acknowledges each. Finally, both Kermits terminate.

We end this appendix by showing the packets used to transfer a file called KERMNAME from a PC to a mainframe. KERMNAME contains these seven lines:

```
                **********
Kermit is not an acronym.  It was named
after Kermit the Frog, star of the
television series, The Muppet Show.
Used by permission of Henderson
Associates Inc.
                **********
```

The packet exchange was as follows:

```
s:  , Sn, @-*&1~!!
r:  , Yn, @-*  ~O!
s:  +!FKERMNAMEE!
r:  *!Y?!
s:  h'D~* ~4**M*JKermit is not an acronym.  It was named*M*Jafter Kermit the*!
r:  *'Y@!
s:  h*D Frog, star of the*M*Jtelevision series, The Muppet Show.*M*JUsed by C!
r:  **YA!
s:  [*Dpermission of Henderson*M*JAssociates Inc.*M*J~* ~4**M*JZ!
r:  **$YB!
s:  'A*%ZD!
r:  *%YC!
s:  'A*&B-!
r:  *&YD!
```

where we indicate to the left of each packet whether its originator is the sender or receiver of the file, we do not show the mark character, and we put a ! character immediately after each packet, in case the checksum is a space. The first two packets specify, for example, that the maximum packet length is 78 (n), and that the end of line character is carriage return (-). In this transfer, the sending Kermit program was not Crosstalk but MS-Kermit, and the receiving computer was not running under Unix.

REFERENCES

1. Allen, John. Lanier Data Communications. Lanier Business Products, Atlanta, Georgia, 1985.

2. Knoerdel, Joan E. and Shirley Ward Watkins. Document Interchange Format, NBSIR 84-2836. U.S. Department of Commerce, National Bureau of Standards, Institute for Computer Science and Technology. Washington, D.C., 1985.

3. Lanier Business Products. Lanier No Problem TTY-ASCII Communications. Atlanta, Georgia, 1979.

4. Lanier Business Products. LTE-3 TTY-ASCII Tables.

5. Microsoft, Inc. Disk Operating System. IBM Personal Computer Language Series. Version 2.10. September, 1983.

6. Microstuf, Inc. CROSSTALK XVI. 1984.

7. Zenith Data Systems. ZDS ZSTEMpc: Version 2 with VT100 Option. 1985.