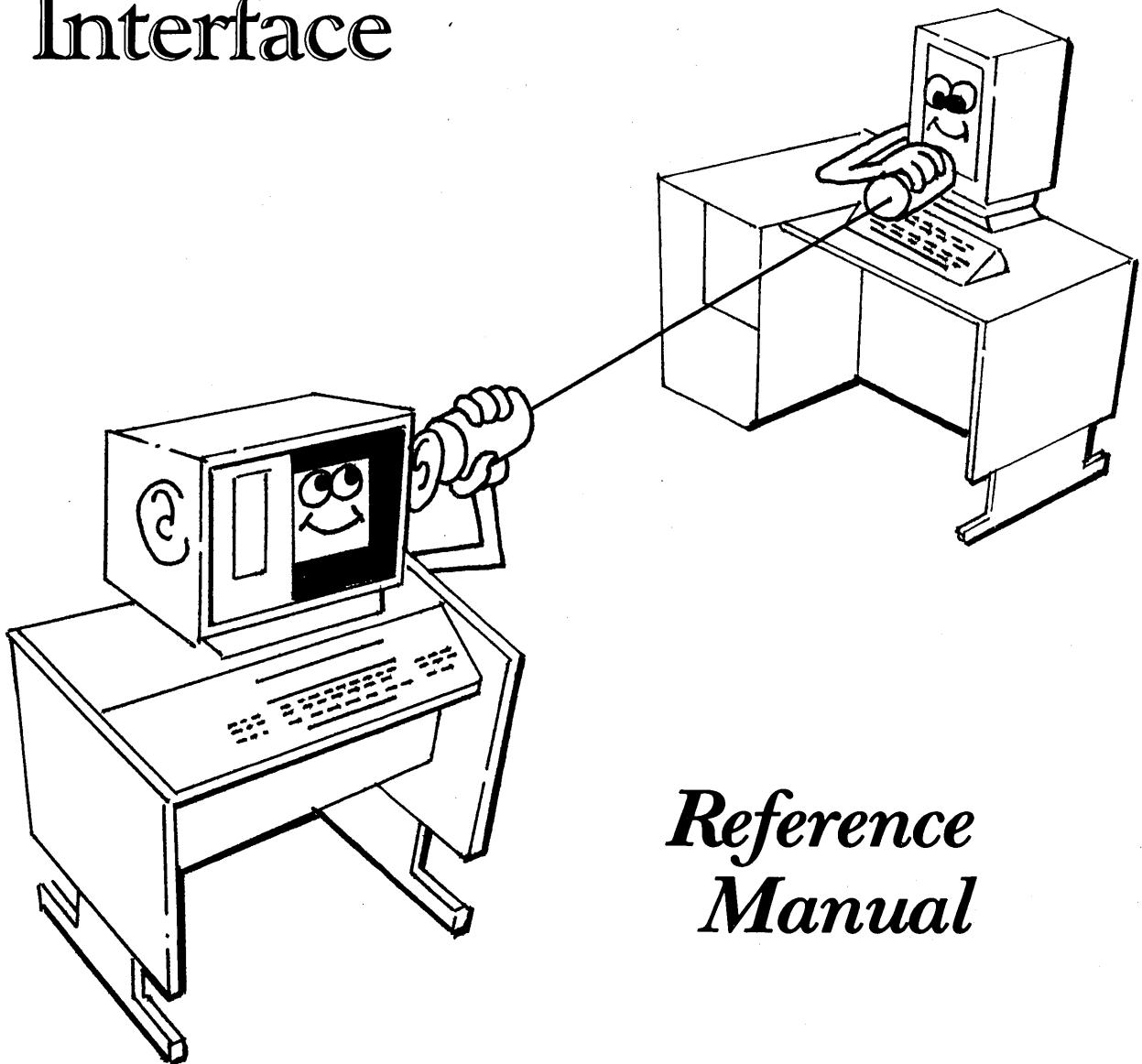


The Intelligent Communications Interface



*Reference
Manual*

Compugraphic does not guarantee the accuracy, sufficiency, or suitability of the information provided herein. Customers shall inspect and test the information to their satisfaction before using it with important data.

Copyright July 1981

THE INTELLIGENT COMMUNICATIONS INTERFACE REFERENCE MANUAL

The following text is designed to assist the operator in communicating with word/data processors via the Intelligent Communications Interface (ICI). This Manual was prepared and typeset by Compugraphic's Communications Assistance Center. The text was input, proofed, and edited on several word processors and telecommunicated to the EditWriter. The typefaces used to produce this manual are Baskerville II, Baskerville Italic II, and Baskerville Bold II.

July 1981

TABLE OF CONTENTS

WE WOULD LIKE YOU TO MEET ICI

What The ICI And EditWriter Can Do For You	7
Planning Your Work Or What Needs To Be Done	8
The Best Way...The Right Way...The Wrong Way.....	9
Translation Tables Fall Into Three Categories.....	9
Your Three Choices	10

OPERATORS GUIDE (A Button Pusher's Guide)

Preparing For A Communications Session	16
Selecting The Proper Mode of Operation	17
Filling In The Bisync Menu For ICI Mode.....	19
Filling In The Async Menu For ICI Mode	21
Setting Up Computer Terminal Mode	23
The Communications Session...What You Can Expect To See	27
Hints About Some Popular Word Processors	28
IBM's OS/6 Menu	29
Vydec's 1400 Menu	30
Wang's Communications Menu.....	31
Xerox's Communications Menu	32
Processing Copy At The EditWriter	33

TECHNICAL MANUAL

→ Asynchronous Communications	35
X-on/X-off For Pacing The Transmission	35
Async Bit Rates (Alias Baud Rates)	36
Parity And Stop Bits	37
Computer Terminal Mode	38
Asynchronous Command Key Sequences	40
Synchronous Communications (Also Known As Bisync)	41
Bisync Bit Rates (Alias Baud Rates)	42
EBCDIC WP And DP Code Set Differences	43
→ Attended And Unattended Communications	43
→ Disk Write And File Naming Conventions	44
Connecting The Hardware	44
RS-232C Minimum Requirements	46
RS-232C Port Definition	47
Telephone Modems	48
Acquiring Modems	49
Bell 212A (Data Phone 300/1200) Modem Operation	50
Bell 201C (Data Phone 2400) Modem Operation	51
Cable Connections—Modem Elimination	51
Definition And Installation Of The ICI's Null Modem	52
Timing When Cable Connected	53
~ Terminal Port Connections . . . Why They Don't Usually Work	54

WHAT DO I DO IF...?

ICI Self test	55
ICI Self-Test Using A Loopback Connector	56
ICI Self-Test Using A Modem	57
Problem . . . Cause . . . Solution	58

ALL ABOUT TRANSLATION TABLES

What You Should Know Before Getting Involved	61
Getting Started	63
What Is A Default Table?	64
What Is A Translation Table?	65

Translation Table Format	66
The Keyword	66
The Body of the Table	67
Alpha To Alpha Translations (A)	68
Alpha To Hex Translations (H)	69
Hex To Hex Translations (V)	70
Hex To Alpha Translations (R)	72
The End Statement	73
Initialization	74
Precedence Of Translations	75
Deleting Codes	76
Space Compression	77
When To Make A Translation Table	77
Representing Word Processor And EditWriter Codes	80
The Hexidecimal Numbering System	81
Recommended Techniques For Customizing Tables	81
Handling Tabs	81
What To Do About Line Endings	83
Hyphenation	86
Things To Remember About Quotes	87
Dot Leaders	87
Paragraph Indents	87
Special Printer Control Codes	88
Upper Case To Lower Case	89
Removing Page Headers	90
Generating Multijust Commands	91
Margins, Line Feeds, And Double Spacing	91
Preparing Copy For The ICI	93
Large Text Applications	93
Approach 1	94
Approach 2	94
Large Tabular Applications	95
Approach 1	96
Approach 2	96
For Big Important Jobs, Try A Small Test First	97
Coding At The Word Processor	98

Transmitting The Table And Text	101
Finding And Correcting Translation Table Errors	102
Common Table Errors	102
Finding Table Errors.....	103

APPENDIX

EditWriter Keyboard Layout	105
EditWriter Hex Chart.....	106
ASCII Code Set Chart	107
ASCII Default Table.....	108
EBCDIC/WP Code Set Chart	109
EBCDIC/WP Default Table	110
ASCII Code Identification Table	111
EBCDIC Code Identification Table	113
Sample Kerning Table.....	115
Sample Table For Decimal Tabs	116
EditWriter Keyboard Layout For Async Terminal Mode	117
 GLOSSARY OF TERMS ("Pardon me... you said what?")	119
 GLOSSARY OF WORD PROCESSOR TERMS.....	127
 INDEX	137

WE WOULD LIKE YOU TO MEET ICI

The Intelligent Communications Interface (ICI) is a general purpose device used to receive work from other products using digital telecommunications. It is installed inside the EditWriter and cable connected to an outlet on the back panel. The outlet is the industry standard called RS-232C. It is connected by a cable to other devices, such as telephone modems, word processors, computers or communicating OCR scanners.

The ICI allows the EditWriter to receive text and data electronically from word processors and computers that have the ability to telecommunicate. Information can be transferred at sustained speeds as high as 1,000 words per minute, with burst speeds of 19,200 words per minute. A direct cable connection may be used when the sending device is in the same vicinity (generally within 50 feet) as the EditWriter. A telephone modem allows machines that are located across the city or across the country to communicate with each other.

One ICI is all that is necessary to interface with all of the various kinds of communicating word processors and computers manufactured today. The ICI uses different Translation Tables to convert the keystrokes used on the word and data processors into EditWriter keystrokes before it stores them on the disk.

For your convenience, numerous Translation Tables are included for many popular brands of word processors and computer systems. Other, less popular tables are also available upon request. In addition to translating the different code sets, the tables compensate for many of the differences in the communications capabilities of the various manufacturers. The Translation Tables also adjust for many of the normal differences between typing on a word processor and keyboarding copy on a EditWriter. These tables can be modified by the user to suit individual requirements.

In the Asynchronous mode, the ICI will also allow the EditWriter to be used as a interactive computer terminal. It allows you to "log on" to even the most sophisticated computer system, select files and specify that they be transmitted to you. The EditWriter screen will display the prompts, messages, and data sent back from the computer. At your instruction, the EditWriter will switch out of Terminal Mode and into the ICI Mode. After receiving the file(s), the ICI can be switched back into the interactive Terminal Mode to request additional files or for computer log-off.

In the ICI mode, the received data will be translated and then automatically stored on the EditWriter's floppy disk. Later, the stored data can be edited, and additional typesetting commands may be easily added at the EditWriter keyboard. When the EditWriter operator is satisfied that the file is complete, the photo unit would be directed to typeset the job.

What the ICI and EditWriter Can Do For You

There are several principal advantages to using an ICI. It can eliminate the cost and the work of rekeyboarding material that was previously keyboarded on a word processor or computer system. It can eliminate the work of reprooing copy that was previously proofed on a word processor or computer system. It can enable the cost-saving features of word processing and computer systems to be used in reducing the cost of typesetting input. It can bring the benefits of typesetting to publications, that would otherwise be printed from typewritten copy, by drastically reducing the high costs normally associated with typesetting. It can be programmed to automatically add the typesetting codes to the manuscript copy, introducing brand new efficiency to the typesetting process by eliminating the need for either the word or data processor operator to key in typesetting codes by hand.

The ICI has many different features. It is not necessarily most efficient to try to employ all of them at one time. How many are best used at once may be determined by a variety of factors. Some of the most important factors will be the size (or volume) of the job, the complexity of work being done, the amount of preparation involved, and the degree of control that the typesetter can exercise over the manuscript preparation.

It will be practical in some applications to do all of the following in one pass.

- Receive the data from a modem or cable.
- Convert the ASCII or EBCDIC codes into EditWriter codes.
- String search and replace many items simultaneously.
- Format with typesetting commands such as size, line-space, font, line length, etc.
- Break into easy-to-access files.

- Generate sequential file names.
- Store files on the EditWriter disk.
- Typeset copy in the background.

All of the above may be performed automatically without an operator present when the ICI is set up in the unattended mode.

Planning Your Work or What Needs To Be Done?

To produce typeset copy from a typewritten manuscript, additional typesetting instructions need to be added to the typewritten text. These extra instructions will make use of the additional characters, aesthetics and other features available on the EditWriter. Instructions are added for purposes such as changing the sizes and styles of the various typefaces, adding ligatures, performing kerning, and reformatting into a reduced size and shape to lower printing costs. In text material, the line endings have to be changed to accomodate the difference between the number of typewritten and typeset words that will fit into a justified typeset line. Centered and tabbed words may need repositioning as the typeset words may be larger or smaller than the typewritten words depending on the size and style of type chosen.

Manuscripts prepared and recorded on a word processor include many invisible keystrokes that are used to position or format the typewritten characters on the paper. These invisible codes may be used if they are translated into the corresponding typesetting format commands. This can reduce the number of typesetting commands to be added by hand.

When you typeset from a manuscript that was prepared for typewriter output, and the typesetting will only require one size and style of type, very few instructions will need to be added. When the typeset work requires many changes in size, column width etc., then somewhere, instructions will have to be added for each occurance. There is no

magic. The number of instructions, and the work involved in adding them to a typewritten manuscript, will be proportional to the extent that you use the additional features available on the EditWriter.

The Best Way...The Right Way...The Wrong Way

The actual addition of the "typesetting" instructions can be done in three different ways. No one way is best for everyone. It will be up to you to choose which method or methods you will use. Your decision will be based on the information available on any particular job. Most work will actually use a combination of the methods.

METHOD ONE—They can be added to the EditWriter file by the typesetting operator by editing the file after it has been received.

METHOD TWO—They can be automatically generated by a custom Translation Table in the ICI as the copy is received by the EditWriter.

METHOD THREE—They can be added to the manuscript by the typist using mnemonics defined in the standard Translation Table.

Translation Tables Fall Into Three Categories

DEFAULT. A Default Table is built into the ICI. With this table, all the typesetting instructions need to be added by the EditWriter operator. Coding at the word processor cannot be done.

STANDARD. Standard Translation Tables are provided for most word processors by the Compugraphic Communications Assistance Center. These tables correct predictable inconsistencies, greatly reducing the editing required by the EditWriter operator. These tables are designed to allow you

to use all three methods, and you may add to or modify them to satisfy your special needs.

CUSTOM. Custom Translation Tables are written by the customer to suit a particular job. They can (depending on the complexity of the job) greatly reduce or eliminate the need for an operator to input typesetting commands at the EditWriter or at the word processor.

Your Three Choices

People usually use a blend of the available work methods to get a job done. This is also true for the ICI. Most of the work done on the ICI is a blend of three different methods. Each method has its own distinct advantages and disadvantages. By understanding them, you can choose the best blend of methods for your work.

METHOD ONE—Use a standard Translation Table and enter the typesetting codes at the EditWriter.

This method is the most popular. It is best suited to the largest number of applications. It is generally the easiest and fastest for small to medium sized, one time jobs.

ADVANTAGES. Method one allows the word processor operator to keyboard documents in his/her everyday manner. This eliminates the need for training the word processor operator in the language of typesetting, typesetting markup, EditWriter coding, and the mnemonic representations. It utilizes the everyday skills of the EditWriter operator to insert the typesetting commands.

Less labor is required to insert the typesetting commands on an EditWriter than a word processor due to the dedicated keys on the keyboard. Method one eliminates the need to learn and use the multiple-key, mnemonic commands.

Accuracy is improved with this method as the EditWriter program and screen detects and flags typesetting format errors during entry. The EditWriter screen also displays

special half-intensity characters for typesetting commands and provides the operator with visual feedback on the formatting effect of the typesetting commands.

DISADVANTAGES. This method requires the EditWriter operator to add the additional typesetting commands each time a job is received by the EditWriter. Changes or corrections to the copy are not easily passed back to the original data base.

METHOD TWO—Using a custom Translation Table to enter the typesetting commands automatically.

This method is used on the very large jobs where the longer setup time is offset by greater labor savings. It is also used for repetitive jobs, such as lists and directories, where the same copy, with minor changes, will be republished over and over again.

ADVANTAGES. Method two eliminates almost all of the time and labor needed to insert typesetting commands at either the word processor or the EditWriter every time the document is typeset. Once the setup and testing is complete, this method offers the fastest throughput time for production work.

Uniformity of typesetting is insured by the automatic entering of the commands by the Translation Table. The word processor operator will not require typesetting training. No special computer programs need to be written for copy coming from a computer data base.

DISADVANTAGES. This method requires acquiring and practicing the skills of writing and modifying Translation Tables. The setup time and labor for the job is increased by the time needed to write and test the custom Translation Table.

The typewritten manuscript format of the document must be uniform when it is used to trigger the automatic translation into typesetting format commands. Documents needing many typesetting formats must be designed to use a special typewritten manuscript format.

The success of this method is very dependent on good planning. The plan must be based on a thorough understanding of the details of the entire process.

METHOD THREE—Entering mnemonic typesetting commands at the word processor.

This is the most frequently talked about, and least frequently used, method of the three due to the very high skill requirement of the word processor operator. This method is used for smaller jobs, published repetitively, that require complex typesetting formats.

ADVANTAGES. The document may be revised at the word processor and then typeset without the need to re-enter typesetting codes.

You can use the standard Translation Tables. It is not necessary to learn to write or modify Translation Tables.

Control of the typeset commands and the final appearance is available at the word processor or computer.

DISADVANTAGES. It requires additional training of the word processor operator. The input operator will, at least, have to learn the language and measurements of typesetting and a set of mnemonics to represent the EditWriter commands in order to input from specially marked up copy. If specially marked up or formatted copy is not available, the operator will also have to learn to translate standard typesetting markup into the EditWriter keystrokes required to produce it. When marked up copy is not available, the input operator will also have to learn to design the typesetting layout and produce their own markup.

The skills that the word processor operator will need to acquire will be determined by the amount of coding that is done at the word processor instead of at the EditWriter.

The word processor screen and proof copy will contain typesetting mnemonics that may prove confusing. If

unacceptable, additional labor may be needed to bracket them with “don’t print” commands for the typewritten proof copy.

Correction of errors in the typesetting commands in the word processor or computer documents is awkward. The errors are most likely to be detected by a different operator on a different machine at a different location at a different time.

OPERATOR'S GUIDE—Button-Pushing Guide

We call this section a Button-Pushing Guide because it is not intended to teach you everything you need to know about typesetting or data communications. It's simply designed to help you get through a transmission with a minimal amount of stress and strain. After you have received a few successful transmissions, you will probably want to learn more about data communications and the ICI (we've been told it becomes quite habit-forming). Each of you will probably be curious about different things. You will find more in-depth information in the Technical section of this manual.

We assume that you are familiar with the operation of the EditWriter. All of the material covered in this portion of the manual relates to establishing the communication link and the subsequent handling of data at the EditWriter.

You will note that all EditWriter keystrokes appear in bold face. EditWriter function commands are displayed in bold face and will be bracketed.

Preparing For A Communications Session

Before setting up for a communications session, an initialized disk must be mounted in the EditWriter disk drive. A blank disk should be used whenever possible to minimize the risk of data being lost due to "Disk Full" or "File Name in Use" conditions. A system **[RESET]** or **[CALL INDEX]**, **[EXECUTE]** function should be used prior to setting up the communications menu to ensure that the EditWriter has an accurate reading of available space on the disk.

The applicable Translation Table should be typed and tested by the word/data processor operator prior to the transmission of data. Instructions for inputting and testing tables can be found in "All About Translation Tables" further on in this manual. Translation Tables can be acquired through the Communications Assistance Center.

The EditWriter operator should be provided with a hard copy proof, composite drawing, or previously printed sample of the job to be transmitted. This will be of great help when establishing typesetting parameters (such as column widths, linespacing, etc.) at the EditWriter.

The Intelligent Communications Interface has four basic modes of operation when communicating. They are:

BISYNC ICI MODE—When in this mode, the EditWriter will automatically store the data it receives on the disk. It is used when communicating with word/data processors that use the EBCDIC code set and Bisync protocols (2770, 2780, 3780 or 3741).

ASYNC ICI MODE—When in this mode, the EditWriter will receive text and automatically store it on disk. This mode is used when the EditWriter will be communicating with a word or data processor that does not require "log-on" and uses the standard ASCII code set and the Asynchronous TTY protocol.

COMPUTER TERMINAL MODE—This mode allows the operator to transmit a "log-on" message from the EditWriter

keyboard and is used when communicating with a computer that requires a "log-on" procedure. (See "Setting Up Computer Terminal Mode" for explanations of the keystrokes used in Terminal Mode.)

SELF-TEST MODE—This mode is used to test for proper function of the communications board and related hardware.

Selecting The Proper Mode Of Operation

Use of the proper mode is imperative for a successful communications session. The mode you select is dependent on the capability of the sending device, and on the modems being used. Computers, for example, will usually communicate using the Async Mode. Some communicating word processors may transmit in Async only, some in Bisync only. Others are capable of communicating in either mode. The ICI is very flexible and will accept data transmitted either way. Ask the word processor operator which mode will be used and, if both are available, use Bisync.

The following list outlines the communicating protocols of some popular devices. We have also provided the reference number of the appropriate Translation Table to be used when communicating.

Table ID Number

Word Processor	Protocol	Text	Tabular
A B Dick	Async	23AX	24AT
AES 100 Series	Bisync	9SX	10ST
AES Plus	Async	28AX	27AT
Apple	Async	2AX	1AT
Artec	Async	17AX	18AT
Basic Four	Bisync	5SX	6AT
Burroughs/Redactron	Async	35AX	36AT
CPT	Async	31AX	32AT
CPT	Bisync	39SX	40ST
Commodore	Async	2AX	1AT
Data Point	Bisync	5SX	6AT
DEC WP	Async	47AX	48AT
DEC Computers	Async	2AX	1AT
Honeywell	Async	2AX	1AT
IBM OS/6, Display Writer, 5520	Bisync	15SX	16ST
Lanier LTE-2	Bisync	9SX	10ST
Lanier LTE-3	Async	28AX	27AT
Lanier LTE-3	Bisync	63SX	64ST
Lexitron	Async	41AX	42AT
Lexitron	Bisync	43SX	44ST
Linolex (3M)	Async	59AX	60AT
Micom	Async	34AX	33AT
Micom	Bisync	30SX	29ST
Mnemonic Pi Characters	Async/Bisync	MPC	MPC
Mnemonic Type- setting Commands	Async/Bisync	MTC	MTC
NBI	Async	21AX	22AT
Olivetti	Async	2AX	1AT
Prime	Async	2AX	1AT
QYX	Async	50AX	49AT
Radio Shack 1 & 3	Async	67AXT	67AXT
Radio Shack 2	Async	66AX	65AT
Royal	Async	2AX	1AT
Vydec	Async	37AX	38AT
Vydec	Bisync	20SX	19AT
Wang	Async	25AX	26AT
Wang	Bisync	8SX	7ST
Wordstream	Bisync	61SX	62 S T
Xerox	Async	13AX	14AT
Xerox	Bisync	11SX	12ST

In addition, General Purpose Tables are available for use with devices not mentioned above.

Almost every communicating device, whether it be a word/data processor or a phototypesetter, utilizes a unique procedure to set it up for communications. Some systems use a communications program disk, while others may require a special sequence of keystrokes. In the case of the ICI, four simple keystrokes **[QL]com** put the system in the communications mode. Once in communications, the ICI will prompt the operator for the information it needs to establish the communications link. This series of prompts is called a "menu".

Note: If you will be communicating in the Bisync Mode, read the next section, “Filling In The Bisync Menu For ICI Mode”

If you will be communicating in Async, go directly to the section, “Filling In The Async Menu For ICI Mode”.

If you will be communicating with a computer that requires "log-on", refer to "Setting Up Computer Terminal Mode".

Filling In The Bisync Menu for ICI Mode

The EditWriter's communications menu can be brought to the screen by striking **[QL]com** (for COMmunicate). This procedure will always be used to enter the communications menu, regardless of the mode of operation you will be selecting.

The first prompt in the Synchronous menu is:

SELECT PROTOCOL: Type s

Entering a lower case **s** selects the Synchronous mode (EBCDIC Code Set). The EditWriter will complete the description and continue the prompt sequence.

SELECT BIT RATE: Type **0** (zero) if communicating via modem.
Type **1**, **2**, or **3** if cable connected.

When communicating over telephone lines, the bit rate will be provided externally by the modem. When you are direct cable connected, select a bit rate from the following list that is within the speed range of the sending device.

- 1 = Low speed (1,200 bps)
 - 2 = Medium speed (2,400 bps)
 - 3 = High speed (4,800 bps)

ENTER FILE NAME: Type 1-5 alpha-numeric characters

Enter a file name consisting of 1-5 alpha-numeric characters. If less than 5 characters, strike **[RET]**.

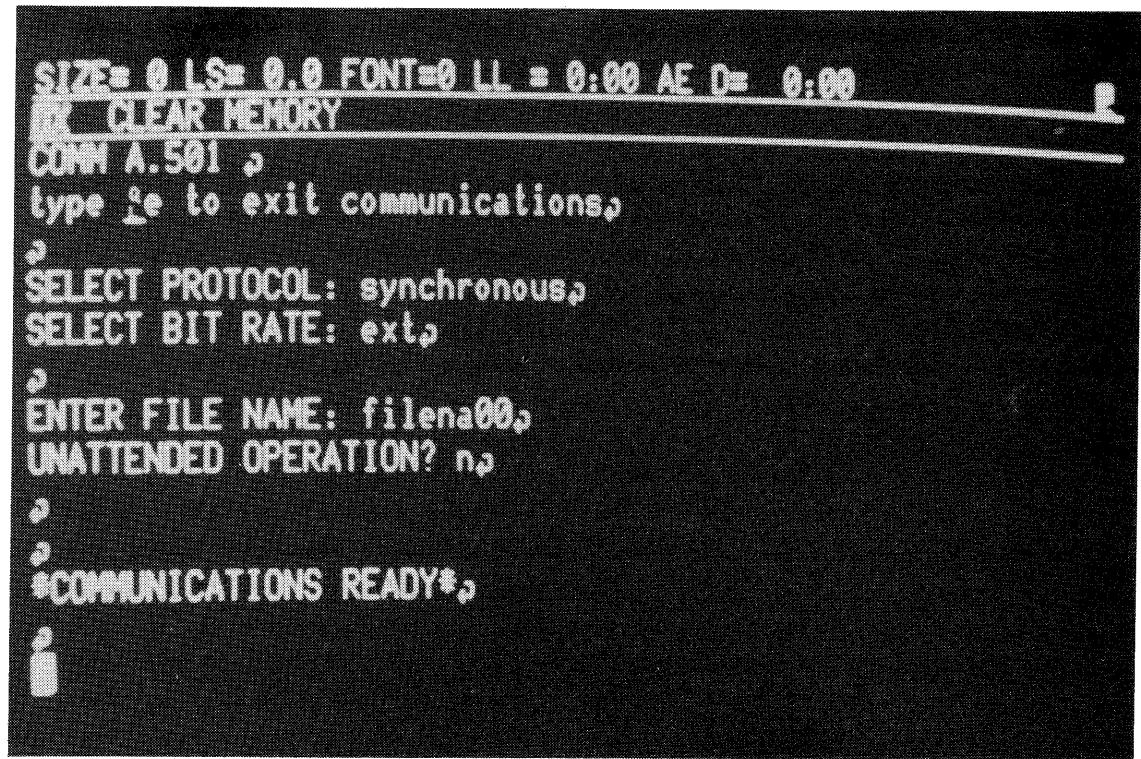
The ICI mode will break the incoming copy into blocks of approximately 2,000 characters. It then stores them on the disk as separate files. The names for these files consist of your filename plus three characters added by the ICI. The first file has "a00" added to it, the second file is written with the extention "a01". The third file has the extention "a02" and so on.

UNATTENDED OPERATION: Type n

The system will receive a single transmission and return, automatically, to normal EditWriter operation.

If answered with a lower case y, the EditWriter will remain in the communications mode at the end of the transmission and, when desired, the operator must exit the program manually by striking [QL]e (for Exit). If the ICI is in UNATTENDED mode, a break in communications will cause the next data received to be written with the extention “b00”.

When the menu has been completed, the EditWriter screen will look like this:



The EditWriter is now ready to receive a Bisync transmission.

Note: Go to "The Communications Session...What You Can Expect To See"

Filling In The Async Menu for ICI Mode

The EditWriters communications menu can be brought to the screen by striking **[QL]com** (for COMmunicate). As you step through the Asynchronous menu, you will notice that it differs slightly from the Synchronous menu. This is due to the fact that parity, stop bits, and terminal mode relate only to Asynchronous communication and have, therefore, been omitted from the Synchronous menu.

The first prompt in the Asynchronous menu is:

SELECT PROTOCOL: Type **a**

Striking a lower case **a** selects the Asynchronous mode (ASCII Code Set). The EditWriter will complete the description and continue the prompt sequence.

SELECT BIT RATE: **e** Select either: **5** (300 bps) or **7** (1,200 bps)

There are 16 transmitting speeds available on the ICI. 300 or 1,200 bps (bits per second) are the most commonly used speeds. For information on the other available speeds, see "Bit Rates" in the Technical section of this manual.

The next prompt is:

N 1
SELECT PARITY & STOP BITS: Type **N** and **1**

This prompt requires two answers. Even parity and 1 stop bit are commonly used in Asynchronous communications. For the other alternatives, refer to "Parity and Stop Bits" in the Technical section of this manual.

ENTER FILE NAME:
nothing will give 200
Type 1-5 alphanumeric characters. If less than 5, strike [RET].

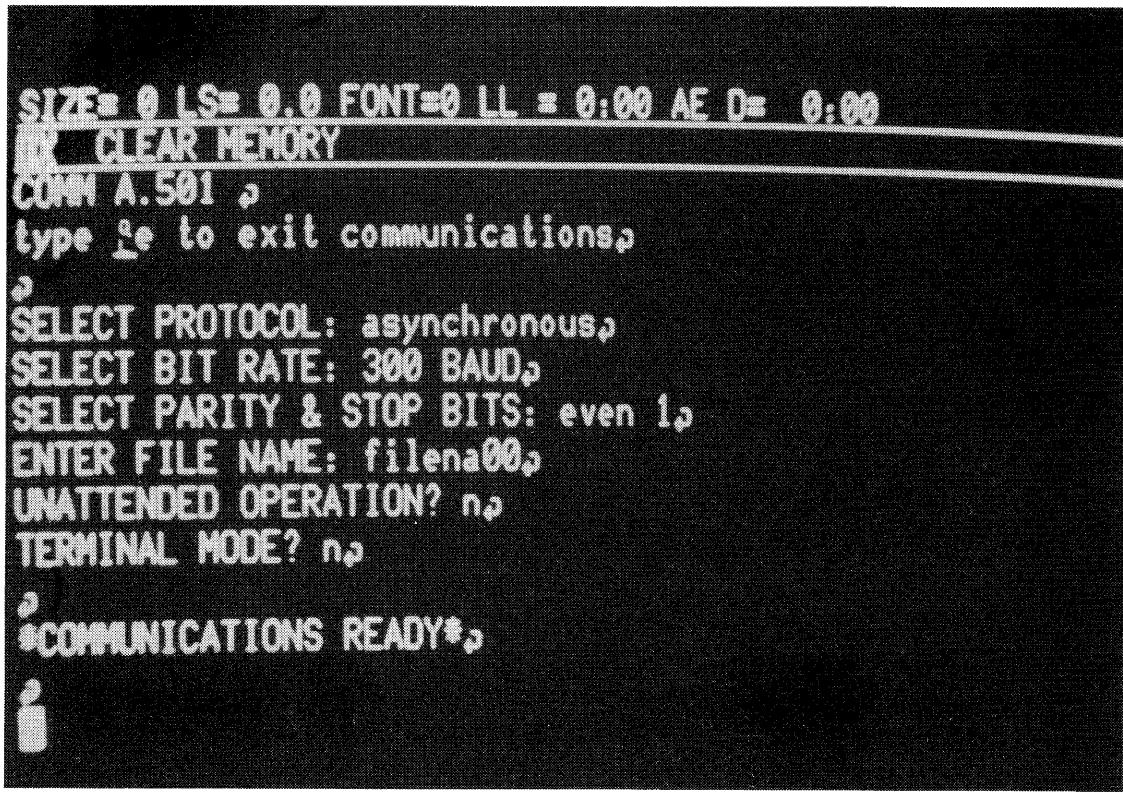
UNATTENDED OPERATION: Type **n**

The EditWriter will receive a single transmission and return, automatically, to normal EditWriter operation. If answered with a lower case **y**, the EditWriter remains in the communications mode at the end of the transmission and, when desired, the operator must exit the program manually by striking [QL]e (for Exit).

TERMINAL MODE: Type **X**

The Asynchronous ICI Mode is initiated by a negative response to this question.

When the Async menu has been completed, the EditWriter screen will look like this:



625 number Recd

The EditWriter is now ready to receive an Async transmission.

Note: Go directly to "The Communications Session... What You Can Expect To See"

Setting Up Computer Terminal Mode

Terminal mode will be used when communicating Async with a multi-user computer that requires "log-on" before data can be transmitted. A "log-on" is a series of key-strokes that identifies you as an authorized computer user, and is designed as a security measure to keep unauthorized personnel out of the computer.

Every brand of computer differs, in some way, from every other brand. Sometimes these differences are minimal...sometimes they can be quite extensive. When "logging on" for instance, some computers will cheerfully advise you that you've made an error in typing your log-on message and allow you to correct it. Others may become quite testy when they encounter an error, and disconnect immediately. Keep in mind that, like the people who program them, computers tend to have distinct personalities and can be a little awkward to deal with until you get to know them. Logging on is not difficult, but it does require precise attention to detail. Your log-on procedure will be provided by the people at the computer-end of the transmission. Copy the procedure carefully and precisely, and make sure that you have a good idea about what you're supposed to see.

The EditWriter's communications menu can be brought to the screen by striking [QL]com.

The first prompt in the Asynchronous Terminal Mode menu is:

SELECT PROTOCOL: Type a

Striking a lower case **a** selects the Asynchronous protocol (ASCII). The EditWriter will complete the description and continue the prompt sequence.

SELECT BIT RATE: **6** Select either: 5 (300 bps) or 7 (1200 bps)

There are 16 transmitting speeds available on the ICI. 300 or 1,200 bps (bits per second) are the most commonly used speeds. For information on the other available speeds, see "Bit Rates (Alias Baud Rates)" in the Technical section of this manual.

The next prompt is:

SELECT PARITY & STOP BITS: Type **N** and **1**

This prompt requires two answers. Even parity and 1 stop bit are commonly used in Asynchronous communication.

tions. For the other alternatives, refer to "Parity and Stop Bits" in the Technical section of this manual.

ENTER FILE NAME:

Type 1-5 alphanumeric characters. If less than 5, strike [RET].

UNATTENDED OPERATION: Type **n**

The EditWriter will receive a single transmission and return, automatically, to normal EditWriter operation. If the answer is **y** (yes) the EditWriter remains in the communications mode at the end of the transmission and, when desired, the operator must exit the program manually by striking **[QL]e** (for Exit).

TERMINAL MODE:

Type **y**

At this point, the screen will go blank. If you are communicating via modem, the telephone connection will be established at this time.

Type the log-on procedure precisely as it was given to you. If you find that the characters you type, other than your secret password, do not appear on the EditWriter screen, don't panic... simply strike **[QL]c** (local echo) and continue the log-on procedure.

The first file you extract from the data base will be the appropriate translation table. End the document request line with **[QR]** rather than a **[RET]**. Within a few seconds you should see the words "keyword detected" on the screen. This indicates that the table is being loaded into ICI memory. Once the table has been loaded, EditWriter default parameters will appear on the screen. When copy stops entering screen memory, strike **[OL]t**. The EditWriter will write the default parameters to disk and allow you to type in the request line for the text document. Again, end the request line with **[QR]** rather than a **[RET]**.

While the text copy is coming in, the EditWriter will pause occasionally to write the information to the disk. You

QL E.
EXIT

Radio []

QL

will notice that the red light on the disk drive is illuminated when the drive is active. When the text stream comes to a halt, and the drive light is not on, strike **[QL]t**. The remainder of the data will be stored on the disk. Type the log-off message and, if necessary, exit the communications program by typing **[QL]e**.

The following is a list of command codes used in Terminal Mode. It should be noted that, unlike many computer terminals, control and/or command keys should not be held down while the function key is depressed.

[QL]c	Used in Async Terminal Mode to initiate local echo.
[QL]n	Turns local echo off.
[QL]s	Brings the transmission status lines to the screen. Depress [QL] to resume transmission.
[QL]t	Use of this sequence will write the data held in screen memory to disk. When in the Async ICI Mode, this sequence enables the operator to access the Computer Terminal Mode.
[QR]	When in Computer Terminal Mode, this sequence transmits a return code and enables the operator to access the Async ICI Mode.
[SHIFT][DH]	Clears the EditWriter screen when in Computer Terminal Mode.
[CNCL CHAR]	Generates ASCII rubout character.
[INS CHAR]	Transmits break signal when in Computer Terminal Mode.
[QC]	ASCII control key.
[QC]q	Transmits X-On and starts transmission.

[QC]s	Transmits X-Off and stops the transmission of data.
[QC]h	Backspace. Cancels previous character entered.

The Communications Session...What You Can Expect To See

Once the EditWriter is in the Communications mode, the actual transmission can begin. If you are communicating via modem, the telephone connection would be established at this time. In applications where direct cable connections have been used, the transfer of data begins as soon both systems are set up to communicate.

The Translation Table is transmitted from the word/data processor as the first document in the session. The table prepares the EditWriter to receive copy from a specific device, and serves to minimize the editing for typographic format that would otherwise have to be done before the copy could be output by the photo unit.

While the Translation Table is loading, the words "keyword detected" appear on the EditWriter screen. When the table has been successfully loaded, you will see the text appear on the screen. The EditWriter will automatically store the data on disk in 2,000 character (approximately), or 175 line files under the file name you established at the beginning of session. You will note that the EditWriter appends your file name with a letter and two digits. The first 2,000 characters transmitted will be appended "a00", the second 2,000 "a01", and so on.

In the last file transmitted, the EditWriter adds a line which advises the operator of the number of characters sent, and the number of errors that occurred (if any) during transmission. In both the Async and Bisync Modes, the system will tell the operator whether or not Translation Table Errors "TTE:" were encountered. Even one table error can have a

negative impact on the effectiveness of the table, and should be corrected immediately. (See "All About Translation Tables" for specific information about locating and correcting table errors.)

In Asynchronous communications, the occurrence of parity errors "PE:" and/or framing errors "FE:" should be noted. When parity errors occur, the operator should search the received data for question marks, and replace them with the proper character, prior to typesetting the copy.

Note: See the Technical section of this manual for more in-depth information concerning the material covered herein.

Hints About Some Popular Word Processors

On the following pages you will find menu set-ups for some of the more popular word processors. This section is not designed to instruct you in the use of the various word processors. It is merely designed to assist you in establishing a link between the sending device and the EditWriter.

IBM OS/6—Bisync Menus

Set up the IBM OS/6 to communicate as follows:

- 1) In the Begin Frame, OPT ADV to "Communicate" and PAR FRAME twice
- 2) Set up the Communicate Frame as outlined below:

SETUP: 1

ID: _____

Auto No Hold Line Yes Summary Yes

CPU Protocol No Code Set EBCDIC/WP Block size 512

Secondary Yes Half Speed No Switch Backup No

Send Text, File, F/T, or All — — — — — — — —

Receive No

- 3) WORD ADV to Job lines. OPT ADV to first job to be transmitted (i.e. table)
- 4) PAR FRAME twice
- 5) Ensure that the menu appears as follows:

Page Image No Send Format _____ Remote Dev _____

Transparency No Record Length _____ Data No Req Test No

- 6) MEM RET and select the second job to be transmitted. Repeat steps 3 through 6 until all desired jobs are scheduled.
- 7) START. "On line" should appear on your screen.

Note: When Cable connected, the IBM must be set up before the EditWriter.

Vydec 1400—Bisync Menu

Set up the Communications Menu as follows:

- 1) Turn "Trace" off.
- 2) Load the communications program by simultaneously holding down CODE, SHIFT, and the DIAMOND keys.
- 3) When program is loaded, remove the program disk and insert the job disk(s).
- 4) If the communications menu is not on the screen hold down the CODE, SHIFT, and the DIAMOND keys.

Answer the menu questions as follows:

Remote type (CPU or Editor)? c RETURN

ENQ: Strike RETURN

Log on: Strike RETURN

Log off: Strike RETURN

Emulator type (2780 or 3780)? 3 RETURN

Block factor? 2/56 RETURN

Send, receive, or both? s RETURN

Disc 1—File A: Type in the track numbers you wish to send. RETURN

Disc 1—File B: Type in the track numbers you wish to send. RETURN

Disc 2—File A: Type in the track numbers you wish to send. RETURN

Disc 2—File B: Type in the track numbers you wish to send. RETURN

Ready? y RETURN

Wang—Bisync Menu (2780 w/transparency)

Set the system up to communicate as follows:

- 1) Ensure that the Translation Table and all other documents are transmitted with transparency. Check the summary page of each document and ensure that two slashes (//) appear as the first two characters on the comments line.
- 2) In main menu, move cursor to "Telecommunications" and strike EXECUTE.
- 3) In the Telecommunications menu move the cursor to "Schedule a Telecommunications Session" and strike EXECUTE.
- 4) If you are communicating through a TC box, the connection name will be 2780. If you are communicating through the Wang workstation type w2780. Strike EXECUTE.
- 5) Select the jobs you wish to transmit, ensuring that the table is the first document sent.
- 6) If you are communicating through a TC box, you are now ready to transmit. If you are communicating through the work station, complete the next two steps.
- 7) Select Telecommunications and strike EXECUTE.
- 8) Select "Switch Workstation to Telecommunications mode" and strike EXECUTE.

Xerox 850—Bisync Menu

Set up the Xerox 850 to communicate as follows:

- 1) Program the 850 to communicate in Comm Mode 3
- 2) Insert job disk(s)
- 3) Move the cursor and highlight COMMUNICATE. MARK and ACCEPT the table and each job you wish to transmit. Ensure that the table is the first job sent.
- 4) Hold down CODE and depress PROGRAM. The Bisync communications menu should be filled out as follows:

COMM MODE	3	■
CODE SET		<u>WP</u> DP DP7 XWP
TRANSPARENCY		<input type="checkbox"/>
BLOCK SIZE	128 256 <u>512</u>	<input type="checkbox"/>
CPU		<input type="checkbox"/>
SEND CARD IMAGE		<input type="checkbox"/>
RECEIVE CARD IMAGE		<input type="checkbox"/>
SEND FORMAT		<input type="checkbox"/>
SEND PAGE END		<input type="checkbox"/>
UNATTENDED		<input type="checkbox"/>
MODEM TYPE	HALF <u>FULL</u>	<input type="checkbox"/>
SYNCHRONOUS CLOCK		<input type="checkbox"/>

- 5) Strike ACCEPT.

Note: MODEM TYPE should be HALF when cable connected, or when using a Half-duplex modem such as a Dataphone 2400 (201C).

Processing Copy At The EditWriter

When the communications session is complete, the files can be edited, rejustified, and output at the EditWriter.

- 1) Strike **[CALL FILE]**, type in the name of the first file in the sequence (for example, 12345a00), and strike **[EXECUTE]**.
- 2) Strike **[REJUST]** and **[EXECUTE]**.
- 3) Edit the copy as required.
- 4) Upon completion of the editing strike, **[LIMIT]**, **[SCROLL UP]**, **[REJUST]**, and **[EXECUTE]**.
- 5) Strike **[REPL FILE]** and strike the **[TYPESET]** key to ensure that the file will be entered "WITH COPY".
Strike **[EXECUTE]**.
- 6) Strike **[CALL FILE]**, 12345a01, **[EXECUTE]**.

Repeat steps 3 through 6 until all of the transmitted copy has been edited and output.

TECHNICAL INFORMATION

Asynchronous Communications

TTY communications (ASCII code set) is the most popular method of communicating with word processors and computers. It is commonly used when calling for text from a time sharing computer system. The various systems may use different combinations of parity and stop bits, and/or run at different speeds (bit rates). The ICI allows you the flexibility to accomodate these variations except for the code set, which must be ASCII. The ICI uses for standard 7-level ASCII—that is, one start bit, seven data bits, one parity bit, and one or two stop bits.

X-on/X-off For Pacing The Transmission

The ICI uses a start-stop protocol to pace the transmitting system when receiving data at speeds greater than 300 baud. Although X-on/X-off is not required at speeds of 300 baud or below, the transmitting system must not be incompatible with this pacing protocol at any speed. The ICI sends an X-on character (DC1-11 hex) when it receives a carriage return (0D hex) and/or when it is indicating that the sending system should continue transmitting. It sends an X-off character (DC3-13 hex) when the EditWriter screen

buffer is full, indicating that it wants the sending system to stop transmitting temporarily. This allows the EditWriter time to write the text it has received to the disk. X-on and X-off must be supported by the transmitting system to allow communications at speeds greater than 300 baud. Failure to support this protocol may result in loss of data. In addition, the X-on/X-off protocol requires that the sending system and any modems used be full duplex.

Async Bit Rates (Alias Baud Rate)

Speeds of transmission are defined in terms of bit rates and baud rates, which mean essentially the same thing. In Asynchronous communications, each character is made up of 9, 10, or 11 bits. A rate of 300 baud means that roughly 30 characters per second, or 300 words per minute, can be received.

The maximum speed that can be used when cable connected is 19,200 baud. Speeds above 2,400 baud are not recommended however, as they will not increase the throughput but may increase the error rate. The maximum speed when telecommunicating is dependent upon the capability of the modems. Typically, Asynchronous modems transmit in a range from 0 to 300 baud, and at 1,200 baud. The ICI allows the operator to select one of sixteen Async speeds. The following list outlines the available bit rates and the EditWriter key used to access them.

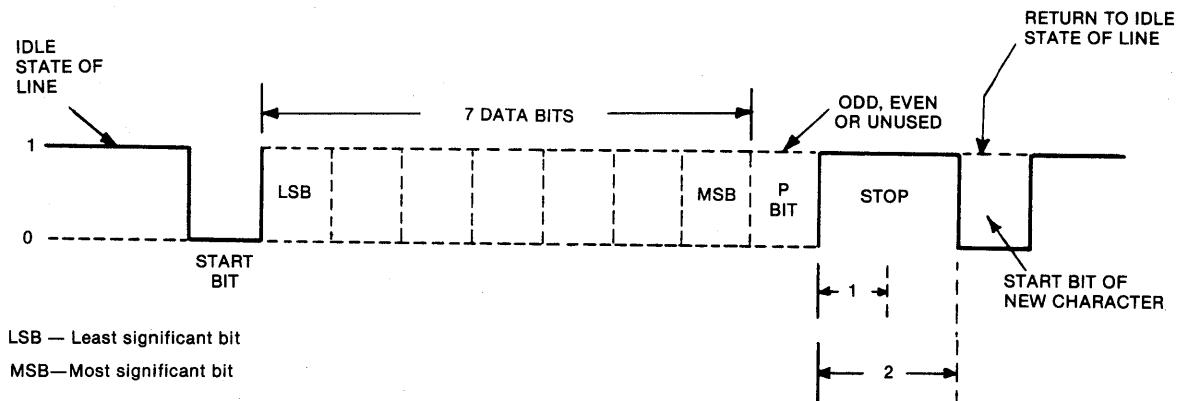
EW Key	=	Speed	EW Key	=	Speed
0	=	50	8	=	1,800
1	=	75	9	=	2,000
2	=	110	a	=	2,400
3	=	134.5	b	=	3,600
4	=	150	c	=	4,800
5	=	300	d	=	7,200
6	=	600	e	=	9,600 ✓
7	=	1,200	f	=	19,200

Parity and Stop Bits

Asynchronous communications provides a method of detecting transmission errors called "parity". Most systems use even parity when transmitting, but the ICI also allows the operator to select "odd", or "no" parity. This is done by pressing a lower case **o** or **n** in response to the prompt "SELECT PARITY & STOP BITS". The ICI must see seven ASCII data bits and a parity bit. The parity bit may be even or odd. If no parity (**n**) is selected, the parity bit must be a logical zero (Space, not mark) for proper operation.

Stop bits indicate the end of the character and can be set to one or two by pressing **1** or **2** in response to the related question in the Async menu. Conventionally, transmission speeds of 110 baud and slower use 2 stop bits, while 134.5 baud and higher usually require only one stop bit.

Asynchronous Data Character Format



Computer Terminal Mode

When a computer supporting the TTY protocol requires that you "log on" and request a file before it can transmit, the Computer Terminal Mode will be used. Terminal Mode is only available when using Asynchronous communications. Entering the Computer Terminal Mode is accomplished by answering the prompt "TERMINAL MODE?" with a lower case **y**. Terminal Mode allows the operator to transmit from the EditWriter keyboard as the keys are pressed, but does not allow you to transmit from the EditWriter disk or screen.

Most computer systems echo back each keystroke that you send. The characters that are echoed back are shown on your screen. To avoid the appearance of double characters on the screen, the ICI does not automatically provide local echo of the keys you press. If you find that the characters you type do not appear on the screen, the computer is not echoing back your keystrokes. You can have the ICI echo them back to the screen for you by pressing **[QL]c**. If you find that you have two characters appearing on the screen for every one you type, turn off local echo by pressing **[QL]n**.

Computer Terminal Mode allows the operator to converse directly with the computer and is only used to log-on. When the document you wish to receive has been identified, the operator strikes **[QR]** on the EditWriter keyboard. Pressing the **[QR]** key transmits a carriage return (0D hex) and should be used instead of a the normal **[RET]** key when instructing a computer to transmit a file that is to be typeset. The **[QR]** command switches the EditWriter from Terminal Mode to ICI Mode, and permits the computer to communicate with the ICI rather than the operator.

When the EditWriter is switched to the ICI Mode, the system will accept and utilize a Translation Table, and automatically stores the transmitted data on disk in 2,000 character files.

When an End of Transmission code (EOT-04 hex) is received by the ICI, the communications session will be terminated. Occasionally the ICI will receive a complete transmission, but the sending system will not transmit an End of Transmission code. To the EditWriter operator it may appear that the sending system stopped transmitting in mid-copy. Not true...the remainder of the copy is being held in ICI memory. The ICI merely requires the operator's attention to terminate the session. Striking **[QL]t** will store the remaining data on the disk, and return the system to Computer Terminal Mode. Exit the communications program by striking **[QL]e** or re-enter the ICI mode by striking **[QR]**.

The ICI allows you to type and transmit all ASCII display and control characters when in Computer Terminal Mode. Refer to the ASCII Terminal Keyboard Chart in the Appendix for specific key locations. The **[QC]** is used as the control key. It is paired with other keys to generate the ASCII control codes.

Note: Do not hold the control key **[QC]** down while striking the companion character. All control functions should be entered as separate and distinct keystrokes.

Asynchronous Command Key Sequences

Keystrokes	Function
[QL]com	Enter communications
[QL]e	Exit communications
[QL]t	Start Asynchronous Terminal Mode
[QL]t	End Asynchronous ICI Mode
[QR]	Start Asynchronous ICI Mode
[QL]c	Local Echo ON
[QL]n	Local Echo OFF
[QC]	ASCII keyboard Control key
[QC]s	X-off
[QC]q	X-on
[INS CHAR]	Generates Break signal
[CNCL CHAR]	ASCII Rubout
[SHIFT][DH]	Clear Terminal Mode screen buffer

Note: See the ASCII Terminal Mode Keyboard Chart, and the ASCII Code Set Chart in the Appendix.

Synchronous Communications (Also Known as Bisync)

The ICI will receive text transmitted using the 2770, 2780, 3741, and 3780 Bisync protocols. No special setup is required at the ICI to distinguish between the different protocols. They are all available when you enter into Synchronous communications. All transmissions however, must use the EBCDIC code set. In most cases, the ICI does not care about record size, or the number of records in a block, provided the record does not exceed 512 characters, which is also the maximum block size. Many computers and word processors however, are not designed to send Bisync at 1,200 baud if the block size is 512. In this case, the block size should be set to 256 characters.

The ICI uses the "WACK" convention to pace the sending system when it has to pause to write to disk. Reverse interrupts (RVI) are not supported.

Most word processing systems have the capability to transmit using the 2780 and 3780 Bisync protocols. Some systems are able to transmit using only the Bisync protocol. One such example is the IBM Office System 6, which uses 2770. The following vendors offer, or indicate that they intend to offer, products with Bisync communications. They are AES/Lanier, Basic Four, CPT, Data Point, Four Phase, IBM, Lexitron, Micom, Vydec, Wang, and Xerox.

In addition, most large IBM computers use Bisync to communicate. It is important to note that Terminal Mode is not available in Synchronous communications. This means that the ICI cannot transmit a log-on message to those Bisync computers that require one. If log-on is a requirement, communications must be done in the Async TTY protocol.

Bisync Bit Rates (Alias Baud Rate)

The bit rate is selected at the EditWriter by pressing one of four numbers, depending on the situation. When a modem is being used, the bit rate is established by the modem. That is, the bit rate is external to the ICI. When a cable connection is being used, the ICI provides the bit rate for the other system, so that the interface emulates a modem via a null modem cable which is included with the ICI.

Key To Press	Speed (bps)	Use With
0	external	Modem
1	1,200	Cable
2	2,400	Cable
3	4,800	Cable

Note: Some word processors and computers cannot transmit as fast as 4,800 bps.

The terms "baud rate" and "bit rate" are often used interchangeably. The words have different meanings. For the technically inclined, baud rate refers to the rate at which the data signals (not the data bits) are transmitted.

EBCDIC/WP and EBCDIC/DP Code Set Differences

The EBCDIC/WP and EBCDIC/DP code sets are largely the same. Both character sets represent the letters and numbers, and most of the other graphic symbols and control codes, with the same values. The ICI's Default Translation Table (which automatically converts the alpha-numerics and punctuation into the EditWriter code set) is designed to use the EBCDIC/WP code set when Bisync communications are used (See the Appendix). The important differences between the two code sets can be compensated for by specific entries in a Translation Table sent to the ICI. The most common table entries are as follows:

Table	Code	Protocol	Hex	Entry
Text	Req'd return	all	0D	/V0D = BCF8/
	Non-req'd return	2770, 3780	1E	/V1E = 08/
Tabs	Req'd return	all	0D	/V0D = F81B/
	Non-req'd return	2770, 3780	1E	/V1E = 29/
Both	Exclamation	all	5A	/V5A = 2F/
	Open brace	all	C0	/VC0 = 8627/
	Close brace	all	D0	/VD0 = 8603/

Not all word processors and computers have two kinds of return codes. Many other systems use other non-standard schemes for representing line endings and carriage returns.

Note: Many word processors and computers use 1E or 1F to represent both required and non-required return codes.

Attended And Unattended Communications

The ICI can be set up in Unattended Mode which means it may be set up to receive multiple transmissions without requiring operator attention. When an End of Transmission code (ASCII EOT-04 hex, EBCDIC EOT-37 hex) is received from the sending device, the ICI will store

the remainder of the information on disk, and exit communications long enough to disconnect the telephone lines. In Unattended operation, the ICI will re-enter the communications mode automatically, using the same protocol and parameters originally established.

When operating in Attended mode, the ICI automatically exits the communications program when an End of Transmission has been received from the sending system.

Disk Write And File Naming Conventions

The EditWriter stores the incoming data in separate files of 2,000 characters or 170 screen lines. The ICI appends the selected file name with "...a00", and increments the alpha character at the end of each telecommunications session. That is, the copy received in the first session will be labeled "...a00", "...a01", "...a02", etc., the second session will be labeled "...b00", the third, "...c00" and so on until the entire lower case alphabet has been used. If necessary, the ICI will switch to using upper case A through Z in the file name appendix. When a blank disk is used in Unattended Mode, the EditWriter will store a maximum of 128 files. When 128 file names have been assigned, the EditWriter will display the message:

MAXIMUM FILE CAPACITY
EXIT COMMUNICATIONS

If a very long file is transmitted in a single communications session, the ICI will append the assigned file name as follows: "a00" through "a99", "aa0" through "aa9", "ab0" through "ab9", "ac0" through "ac7".

Note: To prevent loss of data due to "Disk Full" conditions, a blank, initialized disk should be used at all times. No more than 250,000 characters should be transmitted in a single session.

Connecting the Hardware

There are two common methods used for connecting the ICI to the sending system. First, modems can be used whenever both the sending and receiving devices have access to the dial-up telephone system. Second, cables can be used to directly connect the two devices, subject to consideration of the distance involved.

The ICI utilizes an RS-232C interface. This can be identified as a 25 pin female connector labeled "COMM" on the back panel of the EditWriter. The interface is designed to work with telephone modems. The attachment to a modem is made through a 20 foot modem cable provided with the ICI. The modem cable simply passes the signals on the various pins straight through to the modem.

Some vendors provide modems as an integral part of the computer or word processing system. One example is the IBM OS/6, which is sometimes equipped with an internal modem. It is incompatible with the modems generally suggested for use with the ICI. However, these modems can be made to function with other similar modems. Check with the Communications Assistance Center for details.

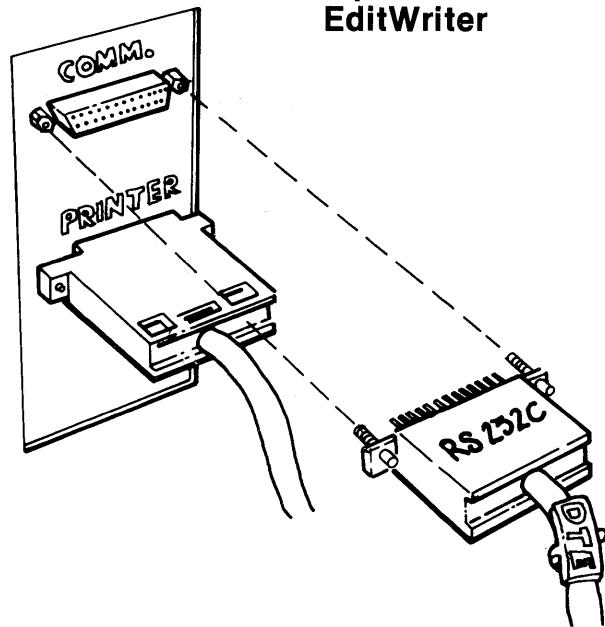
The ICI's RS-232C interface is also designed to work with a direct cable connection with another system. The total length of cable used may not exceed 50 feet according to EIA specifications for RS-232C.

The most simple cable connection has two modem cables, one attached to the ICI and one attached to the other system. The modem cables are both connected to the null modem cable supplied with the ICI. Only one of the two modem cables (which plugs into the back of the EditWriter) comes with the ICI. The other modem cable must be supplied by the vendor of the other system or by the ICI owner.

RS-232C Minimum Requirements

When the ICI is set up in the Asynchronous Mode, pins 4 (RTS) and 20 (DTR) will be set to an ON condition and will remain ON until communications is exited. For proper operation, the ICI must see an ON condition on pins 5 (CTS) and 8 (CD). The ICI will transmit data on pin 2 (TXD) and receive data on pin 3 (RXD). There must be a signal ground connected to pin 7.

Back panel of the EditWriter



Note: An ON condition is defined as +3 to +25 VDC which equals "Space" or a logical zero. An OFF condition is defined as -3 to -25 VDC and is "Mark" or a logical one. The ICI will present approximately +10 VDC for an ON condition and approximately -10 VDC for an OFF condition. Any voltage level between +3 and -3 VDC is undefined and will be in error.

Pin No.	Mnemonic	Direction	Circuit	Description
1	PG	N/A	AA	Protective Ground
2	TXD	From ICI	BA	Transmit Data
3	RXD	To ICI	BB	Receive Data
4	RTS	From ICI	CA	Request To Send
5	CTS	To ICI	CB	Clear To Send
6	DSR	To ICI	CC	Data Set Ready
7	SG	N/A	AB	Signal Ground
8	CD	To ICI	CF	Carrier Detect
11	CTS	From ICI	Non-EIA	Clear To Send
15	TXC	To ICI	DB	Transmit Clock
17	RXC	To ICI	DD	Receive Clock
20	DTR	From ICI	CD	Data Terminal Ready
24	TXC	From ICI	DA	Transmit Clock

RS-232C Port Definition

Pin 1 (PG) is not used but is connected to signal ground on the ICI board.

Pin 2 (TXD). The ICI will transmit data on this pin only when an ON condition is detected on pin 5.

Pin 3 (RXD). The ICI will receive data on this pin only when an ON condition is detected on pin 8.

Pin 4 (RTS). The ICI will present an ON condition on pin 4 (RTS) in the Async Mode when all questions in the communications menu have been answered, and an OFF condition when the communications program is exited. In Synchronous Mode, the ICI will present an ON condition before transmitting a Bisync response and an OFF condition after transmitting a response.

Pin 5 (CTS). The ICI must detect an ON condition on pin 5 (CTS) in order to transmit data. If the ICI cannot transmit when necessary, loss of received data may result.

Pin 6 (DSR). The ICI does not support pin 6 in the Asynchronous Mode. In the Synchronous Mode, the ICI must detect an ON condition in order to transmit or receive data.

Pin 7 (SG) is used as a common return for all RS-232C signals and should be used to establish a zero voltage reference. Pin 7 is connected to pin 1 on the ICI board.

Pin 8 (CD). The ICI must detect an ON condition on pin 8 in order to receive data.

Pin 11 (2nd CTS). This pin is not normally used. When the ICI is direct cabled to the sending unit and Asynchronous communications are used, this non-standard output may be used to pace the sending unit when X-on/X-off is not available. A special cable will be required and it will become the responsibility of the customer to design and support this connection. The ICI will present an ON condition to pin 11 when communications are ready. Pin 11 will be set to an OFF condition when the ICI buffer is full, and will return to an ON condition when space has been made available in the receive buffer. Pin 11 will also be set low when communications is exited. This pin is not supported in the Synchronous Mode and will, therefore, be in the OFF condition.

Pin 15 (TXC) is used in the Synchronous Mode only. If external timing is selected on the ICI, a clock must be detected on this pin in order for the ICI to transmit data. One data bit will be transmitted on pin 2 during each cycle of this clock. If a bit rate is selected on the Sync menu, the transmit clock is provided internally by the ICI. Pin 15 is not used in the Async Mode.

Pin 17 (RXC) is also used only in Synchronous Mode. A clock must be detected on this pin in order for the ICI to receive data. The data bit will be received on pin 3 during each cycle of this clock. Pin 17 is not used in the Async Mode.

Pin 20 (DTR). The ICI will present an ON condition to pin 20 when communications is ready. An OFF condition will be presented to this pin when communications is exited.

Pin 24 (TXC) is used in Synchronous Mode only. If a bit rate is selected in the Sync menu, the ICI will present an appropriate clock to this pin. This clock may be used to clock a sending unit in a direct cable connection, or may be used with a Synchronous modem when the external timing option has been selected in the modem.

Telephone Modems

Modems are devices that attach your EditWriter, computer, or word processor to the telephone system. They allow text to be sent between any two devices that have access to telephones.

A 20 foot modem cable is provided with the ICI. Behind the connector on each end of the cable, you will find plastic tabs labeled DTE (Data Terminal Equipment) and DCE (Data Communications Equipment). The DTE end of the cable should be connected to the COMM port on the back of the EditWriter. The DCE end will be plugged into the female connector on the back of the modem.

Acquiring Modems

Modems may be rented from the telephone company or purchased from a large number of other vendors. The Communications Assistance Center uses the specifications for various Bell Telephone Company modems as reference standards for the purpose of specifying how modems are to function.

There are many different types of modems available from a variety of different vendors. The Compugraphic CAC recommends several Bell Telephone type modems for use in different circumstances. The ICI owner may rent these modems from the local telephone company or buy compatible modems from other vendors. Selection of the proper modem depends on consideration of the following questions.

- Cost—modems may be leased from the Telephone Company, or purchased from a variety of private vendors. Two year's rental from the Telephone Company may be approximately equal to the purchase price of a comparable modem from another vendor.
- Bell compatibility—most of the modems used in the United States are Bell compatible. For this reason it is usually convenient to also use Bell compatible modems. It is important to note that the ICI is not guaranteed to operate with non-Bell compatible modems.
- Protocols—what protocols are the systems that will transmit to the ICI likely to use? Bisync protocols use Synchronous modems. The TTY protocol requires Asynchronous modems.
- Customer compatibility—do any of your customers already have modems that you must be compatible with?

- Flexibility—some modems allow you to communicate with more than one other type of modem.

The four Bell compatible modems recommended for use with the ICI are:

Protocol	Mode	Speed bps	Speed cps	Modem
TTY	Async	0-300	0-30	103J
TTY	Async	0-300	0-30	212A
TTY	Async	1,200	120	212A
Bisync	Sync	1,200	150	212A
Bisync	Sync	2,400	300	201C

Note: The 212A, which is a dual speed modem, comes in two versions. Each version supports 0-300 bps Asynchronous communications in the low speed mode. One version supports 1,200 bps Synchronous communications in the high speed mode. The other version supports 1,200 bps Asynchronous communications in the high speed mode.

Bell 212A (Data Phone 300/1200) Modem Outline

When you have power to the modem, the MC (Modem Check) light will be on.

When the EditWriter is set up to communicate, the MC (Modem Check) and TR (Terminal Ready) lights will be on.

When the connection is established, the MC (Modem Check) light will go out and the TR (Terminal Ready) and MR (Modem Ready) lights will be on. If using 1,200 Baud, the HS (High Speed) light will also be on.

When data is being received, the the TR (Terminal Ready) and MR (Modem Ready) lights will be on and the SD (Send Data) and RD (Receive Data) lights will flash. If using 1,200 Baud, the HS (High Speed) light will also be on. The MC light should be off during transmission.

Bell 201C (Data Phone 2400) Modem Outline

When you have power to the modem, the ON and MC (Modem Check) lights will be illuminated.

When the EditWriter is set up to communicate, the ON and TR (Terminal Ready) and MC lights will be illuminated.

When the telephone connection is established, the ON, TR, and MR (Modem Ready), and MC lights will be illuminated.

When data (responses) are being transmitted, the ON, TR, MR lights will be illuminated, and the RS (Request to Send) and CS (Clear to Send) will flash.

Cable Connections—Modem Elimination

A 20 foot modem cable is provided with the ICI. There is a male 25 pin connector on either end of the cable. The end of the cable labeled "DTE" should be attached to the Communications port on the back of the EditWriter. The "DCE" end will be attached to the 12 inch null modem cable.

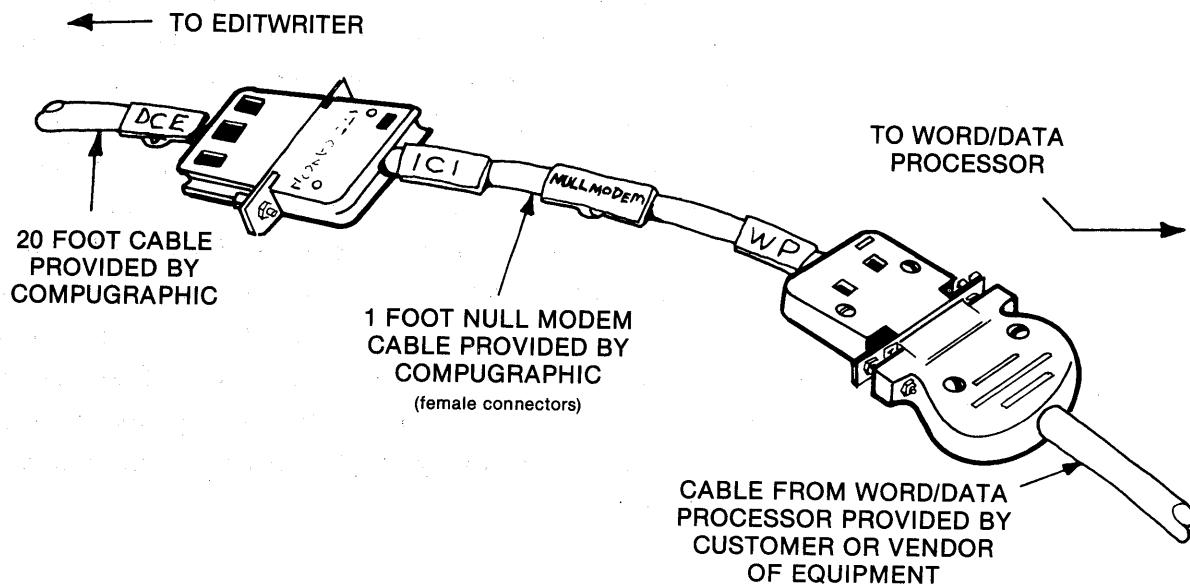
According to EIA standards, the distance that a cable connection can span is limited to 50 feet. This is the maximum distance that the RS-232C electrical signals can be carried without degradation. Cable connections over 50 feet can be accomplished with the use of devices such as modem eliminators, short haul modems, and line drivers. These devices cannot be ordered with reference to Bell standards, but are functional and available from many vendors. The distances over which they communicate range from several hundred feet to several miles. Some require the use of coaxial cable rather than RS-232C modem cable. The performance and price of these devices vary widely. Check with the vendor for installation details.

Definition And Installation Of The ICI's Null Modem

The ICI is provided with a short, one foot cable called a null modem. It is used in the place of modems to enable the ICI to be directly cabled to another system. To another transmitting system (either a word processor or computer) the null modem looks exactly like a real modem. The electrical signals are the same.

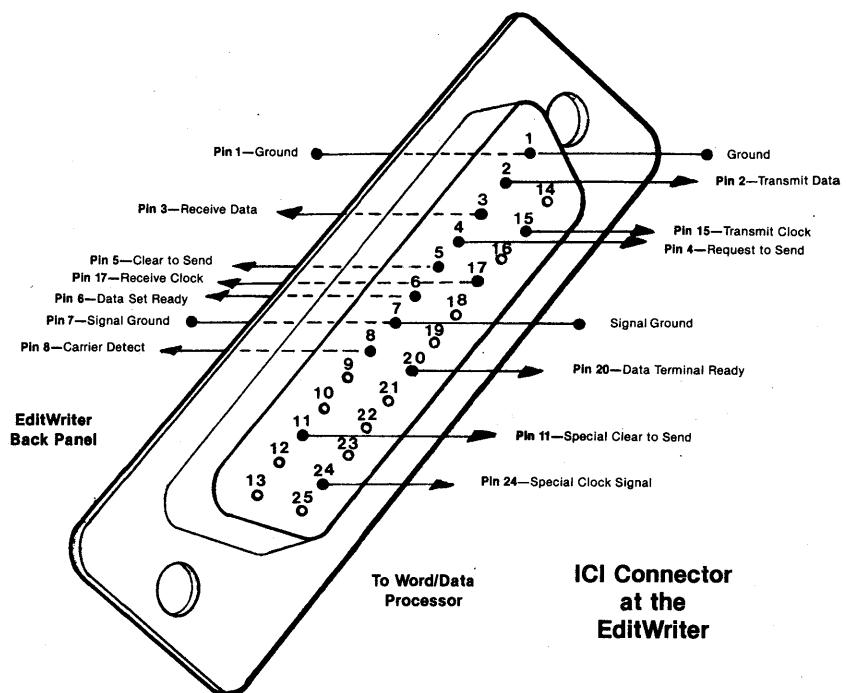
The ICI's null modem has two female 25 pin connectors. These are physically the same as the connectors found on real modems. The two ends of the null modem are labeled WP (Word Processor) and ICI (EditWriter). The null modem should be installed with the WP end attached to the modem cable of the computer or word processor. The DTE end of the twenty foot modem cable supplied with the ICI should be attached to the COMM port on the back of the EditWriter. The DCE end of the modem cable should be connected to the ICI end of the null modem.

ICI NULL MODEM CABLE



If the other system already has a null modem, it should not be used. If two null modems are attached, end to end, the connection will not work.

The null modem is defined below. Paired lines are tied together by the null modem. Specifically, on the ICI end pin 4 (RTS) is tied to 5 (CTS), and 24 (TXC) is tied to 17 (RXC). On the word processor end, 4 (RTS) is tied to 5 (CTS) and 15 (TXC) and 17 (RXC) are tied together. Non-standard CTS is not passed through the null modem.



Timing When Cable Connected

When the ICI is receiving Bisync communications, data is being sent synchronously. This means that characters and control codes are sent in synchronization to a common clock. Normally the clock is provided by the modems used to transmit and receive data. When the null modem is used, the ICI provides the timing clock to the other device. The null modem provides the transmit clock (TXC) on pin 15 and receive clock (RXC) on pin 17.

Terminal Port Connections...Why They Don't Usually Work

Many ICI users wish to connect their EditWriters to computer systems, using the same cables and computer ports as their computer terminals use. This is often impossible because the computer and its terminals do not use a type of connection that is compatible with the TTY protocol. To substitute for a computer terminal, the terminal connection should have all of the following characteristics.

- a) Full duplex—for speeds higher than 300 baud
- b) Supports X-on, X-off pacing protocol for speeds higher than 300 baud.
- c) Not incompatible with X-on, X-off pacing protocol at any speed.
- d) ASCII code set
- e) The ability to electrically emulate a modem interface

If the terminal cable uses only three wires: 2 (TXD), 3 (RXD), and 7 (SG), the connector of the computer's terminal cable that plugs into the null modem must be modified. Jumper pin 4 to pin 5. Jumper pin 20 to pins 6 and 8.

WHAT DO I DO IF...?

The following information is designed to help you at those times when nothing seems to be going right. This section includes ICI Self-Test instructions, as well as a list of common occurrences that people often interpret as system failure.

ICI Self Test

The Self-Test procedures provide a method of checking for proper the function of the ICI board and its related cabling. When running the test, the ICI continuously transmits a test file that is re-directed to the ICI. The ICI compares the transmitted data against the received data and checks for errors.

ICI Self-Test Using A Loopback Connector

To check the ICI board and the cables using a loop-back connector (CG part number 47900-501), perform the following procedure.

- a) Verify that the end of the 20 foot modem cable labeled "DTE" is connected to the "COMM" connector on the back of the EditWriter.
- b) Connect the end of the one foot null modem cable labeled "ICI" to the end of the modem cable labeled "DCE".
- c) Plug the loopback connector into the end of the null modem cable labeled "WP".
- d) Activate the ICI by pressing **[QL]com**.
- e) Press **a** for Asynchronous
- f) "Select Bit Rate" **5** (300 baud) or **7** (1,200 baud).
- g) Press a lower case **e** and **1** to "SELEST PARITY & STOP BITS".
- h) Select an unused filename and press **[RET]** if less than five characters.
- i) Press a lower case **n** for "UNATTENDED OPERATION?"
- j) Press a lower case **s** in answer to "TERMINAL MODE?"

The message "SELF TEST INVOKED" should be displayed on the EditWriter screen, followed by the message "self diagnostic test completed: A-OK". The system should repeat the message until the test is terminated. Failure to repeat the message properly may indicate improper installation or hard failure. Call your local service office.

- k) Run the test for at least 15 seconds.
- l) The test may be stopped by pressing **[QL]t**. The test data will be stored on disk.
- m) Exit communications with **[QL]e**.
- n) Call the test file to the screen to locate the error messages at the end of the file.

ICI Self-Test Using A Modem

Any Asynchronous or Synchronous modem with a local analog loopback capability may also be tested. The local modem may be tested by pressing the “AL” button and running the Self-Test. A remote modem may be tested by pressing the local modem’s “RDL” button or by pressing the remote modem’s “DL” button and running the self test. It’s important to note that this test applies to full duplex modems only.

A Synchronous modem cabled to the ICI may be tested as long as the speed of the self test matches the speed of the modem.

Modems with “AL” (automatic loopback) buttons may be used to self test the ICI. To test the ICI, follow this procedure.

- a) Verify that the end of the 20 foot modem cable labeled “DTE” is connected to the “COMM” connector on the back of the EditWriter.
- b) Verify that the end of the modem cable labeled “DCE” is connected to the “Business Machine” connector on the back of the modem.
- c) Depress the “AL” button on the modem’s front panel.
- d) Activate the ICI by pressing **[QL]com**.
- e) Press **a** for Asynchronous
- f) Select bit rate **5** for Bell 103 and 113 type modems, or **7** for Bell 212A type modems.
- g) Press **e** and **1** to “SELECT PARITY & STOP BITS”, or select your normal configuration.
- h) Select an unused filename and press **[RET]**.
- i) Press **n** for “UNATTENDED OPERATION?”
- j) Press **s** to the question “TERMINAL MODE?”

The message “SELF TEST INVOKED” should be displayed on the EditWriter screen, followed by the message “self diagnostic test completed: A-OK”. The system should repeat this message until the test is terminated. Failure to

repeat the message properly may indicate improper installation or hard failure. Call your local service office.

- k) Run the test for at least 15 seconds.
- l) The test may be stopped by pressing **[QL]t**. The test data will be stored on disk.
- m) Exit communications with **[QL]e**.
- n) Call the test file to the screen to locate the error messages at the end of the file.

Note: Some errors may be encountered when using the Self Test with a Synchronous modem.

Problem...Cause...Solution

<u>PROBLEM</u>	<u>PROBABLE CAUSE</u>	<u>SOLUTION</u>
ble comes up on screen	Improper keyword Format line sent	Check pattern / = // = = //
“keyword detected” never goes away	End of Table code missing Mark Parity	Ensure that /I appears at the end of the table Ensure that Mark parity is not being used, that is, the eighth bit is a logical zero (space, not mark)
ops in the middle copy—Async only	No EOT from sending system	Strike [QL]t
cessive parity errors	Improper parity selection	Ensure that the ICI is set up to use the same parity as the sending system
cessive framing errors	Improper speed selection Improper stop bit selection	Ensure that the sending and receiving devices are transmitting at the same speed Ensure that the proper stop bit selection has been made
V screen never clears	Cable connections—Null modem cable missing Null modem cable improperly installed ICI failure	Ensure that null modem is in place Ensure that ICI end of null modem is toward the EditWriter and the WP end is toward the word/data processor Perform ICI self-test and/or call your local service office
	Modem connections—No power to modem	Ensure that the modem is plugged in the AC line
	Modems are incompatible	Ensure that compatible modems are used at each end of the transmission.
	Modem failure	Perform ICI self-test and/or call the modem vendor.

<u>PROBLEM</u>	<u>PROBABLE CAUSE</u>	<u>SOLUTION</u>
EW screen clears but data is never received	File consists of less than 128 characters	Strike [QL]t
Data is received but isn't stored on disk	"Disk Full" or "File Name In Use" condition encountered Disk is write protected	Use blank disk Ensure that a write permit tab is in place
	RESET or CALL INDEX function was not used before entering communications	[RESET] or [CALL INDEX] [EXECUTE] prior to entering communications
No characters appear on the screen when I'm typing	Sending system does not echo.	Strike [QL]c
Two characters appear on the screen for every one typed	Both devices are echoing	Strike [QL]n
No half-intensity characters on screen	System is in Terminal Mode rather than ICI mode	Switch to ICI Mode by striking [QR]
All copy runs together	No Translation Table	Send Translation Table
Modem doesn't answer the phone	EditWriter is not in the communications mode.	Initiate communications
	Modem is not plugged into the EditWriter or AC line	Check connections between EditWriter and the modem and modem to power source.
Modem answers but copy is not received	Incompatible modems Incompatible protocols	Ensure that like modems are being used on both ends of the transmission. Ensure that compatible protocols are used at both ends of the transmission
Some copy is received but modem hangs up	Incompatible modems	Ensure that like modems are used at both ends of the transmission.

ALL ABOUT TRANSLATION TABLES

What You Should Know Before Getting Involved

The EditWriter Intelligent Communications Interface (ICI) provides several important capabilities to the EditWriter user. Primarily, it allows text to be transferred to the typesetter without rekeyboarding. In addition, it allows a substantial amount of the required typesetting parameters to be automatically added and word processing codes to be deleted as the text is transferred from the word processor or computer. The Translation Table is used to achieve this second function.

This is an introduction to the concept of the Translation Table. The purpose of this document is to explain:

- How the characters of the word processor or computer are converted into the EditWriter character set.
- How and when to use the Translation Tables provided by Compugraphic.
- How and when to modify these tables.
- How to create customized, special purpose translation tables.

Using the ICI to receive text from word processors and computers is fairly easy. Word processors and computers fall into three similar groups in terms of the ways they use and transmit their codes. Because of these similarities, we have prepared six General Purpose Translation Tables that can be used with almost all communicating devices. These are tables for:

Async Text & Tabular—TTY Protocol with ASCII data

Sync Text & Tabular—Bisync Protocol with EBCDIC/WP data

Sync Text & Tabular—Bisync Protocol with EBCDIC/DP data

If you were to rate the efficiency of these Translation Tables on a scale of 1 to 100, you would find that the accuracy and completeness of their code translations rate between 60 and 90. Translations not added by the table are added at the EditWriter when the transmission is complete.

In addition to the General Purpose Translation Tables, numerous tables have been prepared for use with the more popular word processors. These tables account for the transmitting characteristics of the individual machines, and raise the efficiency of translation to between 80 and 95 on the previously mentioned scale.

A Translation Table allows the sending system to define many multiple character search and replacement operations that the ICI is to perform as it receives the text. When instructed, the ICI will search for a sequence of 1 to 64 word processor characters or codes. The word processor codes can then be deleted, or replaced with up to 128 appropriate EditWriter characters or codes. Up to 2,000 characters can be searched and replaced within a single table.

In general, the Translation Table allows a large part of the format of the original document, as it was created on the word processor or computer, to be retained using EditWriter formatting codes. In addition, it allows the word processing operator to generate command codes like "size" and "font". If desired, the word processor operator can type unique character sequences like ";s10" and have them translated into the EditWriter command **[SIZE]10**.

When you begin to use your ICI, you should refer to the Book of Translation Tables to check on the availability of Translation Tables customized for the word processors and computers with which you will be working.

It's important to note that a Translation Table can only be loaded in the ICI Mode. When the message "keyword detected" appears on the EditWriter screen, the ICI has found the beginning of the table. When it encounters the end of the table code, the screen will clear and text will be received and stored on the disk.

A Translation Table remains loaded in the ICI as long as the power to the EditWriter is not turned off, the ICI is not changed from Asynchronous to Synchronous communications (or vice-versa), the system is RESET, or you send another table.

Getting Started

The first thing to do when starting up is provide the word processor operator with a copy of the appropriate text and tabular Translation Table. The tables will be rekeyboarded and stored in the word processor exactly as they are written. Each table should be stored as a separate document, as only one table can be transmitted per session. It is best if the Translation Table has the same margins as the text file that it is to translate. And, in general, it is best if all left margins are set to zero prior to transmission. Once the tables are entered, copies should be printed and proofed for obvious errors.

The next step is to test the accuracy of the table with a transmission to the ICI. Send only the Translation Table, with the ICI in the non-terminal (ICI) Mode. When the initialization parameters have been written to disk, check the message "TTE:00" at the bottom of the file. If there are no Translation Table Errors (TTE) the count will be 00. The sections "Common Table Errors" and "Finding Table Errors" further on in this section review some of the more common errors and outlines the procedures used to identify and correct them.

When the table is error-free, it is time to try a test transmission of a representative sample of the kind of work that you will be typesetting. This is more a test of the characteristics of the word processor or computer and its operators than a test of the specific job. Check to see that the table is doing a reasonable job. Are the paragraph endings and/or tab codes being preserved? Or are the paragraphs being chopped into smaller pieces by excessive quadding codes?

If the table is not quite right, see the ASCII and EBCDIC Debug Tables in the Appendix. These tables enable the operator to find out exactly what hex codes are being transmitted to the ICI. With this information you can either suggest new techniques for text entry at the word processor, or adjust your Translation Table entries accordingly.

The section entitled "Preparing Copy for the ICI" describes how to prepare copy—both text and tabular material—to maximize the efficiency and simplify the preparation of ICI Translation Tables.

If the length of the job, or frequency of its repetition, justifies investing more time to customize the Translation Table to increase its efficiency, see the section "When To Make Translation Tables".

What Is A Default Table?

The ICI utilizes either an Asynchronous or a Synchronous communications protocol. By pressing "a" or "s" in response to the prompt "SELECT PROTOCOL", you

select a communication protocol and one of the built-in character Translation Tables. These built-in tables are called Default Tables. If you select the Asynchronous Mode, the ICI will prepare to translate the ASCII code set into the internal code set of the EditWriter. If the Synchronous Mode is selected, the ICI sets up to translate EBCDIC.

If you compare the EBCDIC Code Set Chart and the EBCDIC Default Table (see the Appendix), you will note that the alphabet, numbers, and all forms of punctuation are assigned the same hex values. This means that the ICI will automatically translate these characters during transmission. Characters or functions that are not assigned like hex values should be converted by the Translation Table.

What Is A Translation Table?

The Translation Table is an ordinary text document created at the word processor or computer. It is written in a special format so that the ICI will recognize it as containing instructions on how characters and codes should be translated, rather than as a normal text document.

When you would like the ICI to translate something differently than the way the Default Table would do it, you have the freedom to replace portions of the Default Table with a Translation Table. For example, the IBM Office System 6 has two kinds of return codes: required returns and ordinary (non-required) returns. The EBCDIC Default Table translates the former into an EditWriter Return and the latter into a word space. With an addition in the Translation Table, an IBM required return could be converted into the appropriate EditWriter codes—Quad Left and Return.

Word processor sends: Required return
Standard default translation: Return
New translation: Quad left and return

Many word processors have a Search & Replace function that enables the operator to search for a string of characters, remove them from the text, and automatically replace them with another string of characters. Each statement in a Translation Table performs the same function—a string of characters from the word processor can be searched out and replaced with the appropriate EditWriter keystrokes. Up to 64 word processor characters (whether visible or invisible on the word processor's screen) can be searched within a single table entry. When the ICI finds the characters, it removes them from the text. The characters may be deleted or replaced by one or more (up to 128) characters appropriate for the EditWriter. These translations make it possible to add typesetting parameters to text prepared at a word processor or computer.

Translation Table Format

A table has four main sections: (1) the Keyword, which defines the beginning of the table and how it will be structured, (2) the body, which contains the specific translation instructions and comments, (3) the End statement, which simply defines the end of the table, to distinguish it from any normal text that might follow, and (4) an initialization, which sets default values for typesetting parameters and tabs.

The Keyword

The Keyword consists of eight characters and is composed of a pattern of two characters that are infrequently used in the text that is to be sent to the EditWriter. It is written in a special pattern that the ICI is programmed to recognize. For example, in our General Purpose Translation Table, we use / (slash) and = (equal sign) as the two characters. We could as well pick + (plus) and & (ampersand), A and Z, or any other two characters. The characters are also chosen for their readability.

When the two characters are picked, the keyword is written in a pattern like this (this is an arbitrary but functional example):

/=//==//

The Keyword must be within 128 characters of the beginning of the document. Some word processors transmit headers or format lines ahead of the text on the screen. This data counts in the 128 character total mentioned above and can cause the keyword to be missed.

When the Keyword is received by the ICI, the message "keyword detected" appears on the EditWriter screen. This means that the Translation Table is being loaded into the ICI's off-screen Translation Table buffer. It's important to note that the table should never appear on the EditWriter screen.

The Body of the Table

The translations that you wish to define are written as statements, each of which have seven parts. They are: (1) the beginning code ("delimiter")—the first character of the keyword, (2) the type of translation code ("designator"), (3) the code(s) that the word processor will send, (4) the "separator" is always the second character of the keyword and is used to separate the characters to be searched from those that will be put in their place, (5) the code(s) that the EditWriter will save, (6) the ending code, which is the same as the beginning code, and (7) comments.

It's important to note that the characters, or hex values, to the left of the separator (=) will always represent the "search routine", or the data coming from the word/data processor. Those to the right of the separator represent the "replace routine", or the data required at the EditWriter.

By way of illustration, let's assume that you want the required return transmitted from your word processor, to

become a Quad Left and Return at the EditWriter. We will step through the components listed above and construct the applicable define statement.

The beginning code would be a slash (/), or the first character of the Keyword.

The type of translation code, or designator, can be either H (alpha to hex), A (alpha to alpha), R (hex to alpha), or V (hex to hex). In this example we will use V, as we will use hexadecimal numbers to represent both the word processor and EditWriter keystrokes.

The word processor's required return is represented by the hex value 06 (zero, six).

The second character of the Keyword (=) is used as the separator, that is, the equal sign is used to separate the search routine from the replace routine.

The codes that are desired at the EditWriter are Quad Left and Return or BC and 29

The ending code would be a slash (/).

Lastly, we add comments to explain the translation and the final product looks like this:

/V06=BC29/...required return becomes Quad Left, Return

Note: The 06 hex (RCR—required carriage return) can be found in the seventh row, first column of the EBCDIC/WP code set chart in the Appendix. The quad left and return can be found in the EditWriter Code Set chart.

The ending code of one statement may be the beginning code of the next statement.

Alpha to Alpha Translations (A)

The Alpha to Alpha translation is one in which the actual characters, or keystrokes, are typed into the define

statement rather than their hex representations. The designator in this case is an A, representing an "Alpha to Alpha" translation. This type of translation is just like the "search and replace" functions of many contemporary word processors.

Case in point...suppose instead of typing "EditWriter" all the time, you wanted to just type "EW". A single entry added to the standard Translation Table could take care of this situation quickly and easily. The table entry could read:

/AEW=EditWriter/.....EW becomes EditWriter

We say "could read" for a reason. What would happen if the work NEW appeared in the text? Based on the example given above, the word NEW would be translated NEditWriter...not acceptable. The table entry *should* read:

/A EW= EditWriter/.....Space, EW becomes space, EditWriter

A word space placed prior to the EW makes the search routine unique. The ICI will only convert EW into EditWriter when EW is preceded by space.

Note: Alpha to alpha translations cannot be used to modify the default character to character translations.

Alpha to Hex Translations (H)

An Alpha to Hex translation is one in which the search routine is represented in actual keystrokes and replace routine is represented in appropriate EditWriter hex values.

Most word processors don't have command keys that correspond to the command keys on the EditWriter. How then do you generate the such things as SIZE, FONT, LINE LENGTH, and LINE SPACE parameters at the word processor? The solution can be to use multi-character command sequences like "FONT" and "SIZE" to represent the Font and Size keys on the EditWriter keyboard.

Suppose, for example, you would like to state the type style and size which should be used to typeset a document. Here are three table entries that specify font and size

represented in alpha characters, and translate them into EditWriter codes represented by hex values.

```
/HFONT 1 =D23B/. . . . . FONT 1 becomes Font 1  
/HFONT 3 =D203/. . . . . FONT 3 becomes Font 3  
/HSIZE 10 =D63B2D/. . . . . SIZE 10 becomes Size 10
```

In the text the operator would insert “FONT 1 or 3” and “SIZE 10” where the font and size changes are needed, like this:

SIZE 10 FONT 1 The nice thing about bold type
is that, when you read it, you FONT 3 know
FONT 1 it is important.

The end result of this entry would look like this:

The nice thing about bold type is that, when you read it,
you know it is important

In the Appendix you will find two charts that will provide the hex values for all of the EditWriter keystrokes. One chart looks like an EditWriter keyboard layout. Where there are two values assigned to a single key, the lower value is the unshift character (eg. “b”) and the higher value is the shifted character (eg. “B”). The other chart tells you what the key or function is, if you know the hex code.

Note: If you are unfamiliar with the hexadecimal numbering system, see the section “The Hexidecimal Numbering System.”

The string of hex codes that appear on the left or right side of the equal sign must all be on the same line—not broken by a return code—or a Translation Table error will occur.

Hex to Hex Translations (V)

The V designator is used when the word processor and EditWriter keystrokes are represented by their hex equivalents rather than the actual keystrokes. See the Appen-

dix for the ASCII, EBCDIC, and EditWriter code set charts.

For example, suppose you are using a word processor which sends a 0D hex code for a carriage return. You know that two carriage returns appear between paragraphs in the word processing document. At the EditWriter, all paragraphs should be quadded left, the line ended, and a line skipped. The appropriate table entry would be as follows:

```
/VODOD=BC29CE/...return, return becomes Quad Left,  
Return, Plus Line Space
```

Suppose that you have a document that is frequently reproduced on your word processor called "Monthly Report". Suppose those are the first two words that appear in the document. Now you can use the V designator to assign size and font parameters to the job automatically. You might use a table entry like this:

```
/V4D6F6E7468=D63B2DD23BD43B272DD0272D2D2D7830182028/....Month  
becomes      Sz10   Ft1  LS120    LL2000    M o n t h
```

Note that the ASCII code chart in the Appendix was used to define the codes on the left side of the statement. This implies that Asynchronous (TTY) communications would be used. The result of this table entry would be an expansion of the word "Month" in the word processor into a string of typesetting parameters and the word "Month" at the EditWriter.

Also note the way a comment has been added to make the long string of hex codes more understandable. The process of adding comments is strongly recommended.

If the character to character translations of the Default Tables are to be modified, then hex to hex statements are to

be used. For example, the Default Tables do not translate tabs. Thus the statement:

/Apumpkin=tasty(tab)pie/...pumpkin becomes tasty, Tab, pie

would translate the word "pumpkin" to:

tastypie

If this entry is added:

/V05=1B/.....Tab to Tab

then the results are different. "pumpkin" now becomes:

tasty(tab)pie

Hex to Alpha Translations (R)

If you wished to find out exactly what hex codes your word processor was transmitting, the R designator allows you to build a special table that will capture and display each code in a readable form. For a system with Bisync communications, the table would have entries that look like this:

/R00=-00-/R01=-01-/R02=-02-/R03=-03-/
/R04=-04-/R05=-05-/R06=-06-/R07=-07-/
/R08=-08-/etc.

These entries tell the ICI to put the code string -02-into the text when it receives a 02 hex code. It tells the ICI to put a -06- and carriage return in the text when it gets a 06 hex code from the word processor. This process is used only as a test to see how V type table entries would be constructed.

For Bisync communications, this diagnostic table would have at least 64 entries for all values between 00 and 3F hex. For TTY communications, the table would have at least 32 entries for 00 to 1F hex. See the Appendix for copies of the ASCII and EBCDIC code identification tables.

Note: A different but allowable format was used to create these Translation Table statements. The ending delimiter of one entry becomes the beginning delimiter of the next.

In summary, we will compare the four different designators. In cases where the V and R designators were utilized, you will notice a difference in the left side (word processor) of the define statement depending on the protocol used.

Suppose we were to make a Translation Table entry to convert the word "Apple" to "Pie" automatically at the ICI. Here's how the table entries would look using the various designators outlined in the previous text.

ASCII—TTY

```
/AApple=Pie/  
/HApple=6C0C02/  
/V4170706C65=6C0C02/  
/R4170706C65=Pie/
```

EBCDIC—Bisync

```
/AAApple=Pie/  
/HApple=6C0C02/  
/VC197979385=6C0C02/  
/RC197979385=Pie/
```

Note that the left side of the translations always represent the word "Apple" completely with alpha codes (normal characters) or with hexadecimal code pairs. The two forms are never mixed within the same define statement. The same applies to the right side of the translations. Mixing alpha and hexadecimal codes on the same side of the equal sign will cause a Translation Table Error.

The End Statement

The end of the table is simply:

/I

This statement is always composed of the first character of the keyword and a capital I.

If the "keyword detected" message fails to disappear from the EditWriter screen, or if a portion of the beginning of the text file mysteriously disappears, it is likely that the /I

was missing from the end of the table. Without this marker to define the end of the table, the ICI will assume that your text file is part of the Translation Table, and mistakenly load it into its off-screen buffer.

Initialization

Following the end statement in every table prepared by the Communications Assistance Center, is a default initialization of typesetting parameters and tabs. The purpose of this sequence is to allow copy received at the EditWriter to be galleyed (if so desired) immediately after transmission.

The default parameters can be easily changed by rewriting the define statements in the Translation Table, or striking over the applicable entries at the EditWriter after transmission. The default parameters of the text tables differ slightly from those of the tab table.

The default parameters outlined below will be found in all standard text tables for use with the ICI.

```
;INIT :ts102004300j ;ts204004100j ;ts306003900j ;ts408003700j  
;ts510003500j ;ts612003300j ;ts714003100j ;ts816002900j  
;ts918002700j ;ct1 ;E ;tr
```

The entry ;INIT generates an initialization sequence in the first file of the transmission. The system will be set up to produce ten point type on twelve point line space. The EditWriter will use font one, the line measure is set to 45 picas (seven and one-half inches), and line-end decisions will be made, automatically, by the system. The entries prefaced with ;ts set up nine tab positions in two pica increments from the left margin. ;ct1 programs the system to recognize tab codes, ;E ensures that copy will be received in the Edit mode, and ;tr generates a tab return to end the initialize sequence.

The default parameters for tabular work are set as follows:

```
;INIT ;ts100000900j ;ts209000406j ;ts313060406j  
;ts418000406j ;ts522060406j ;ts627000406j ;ts731060406j  
;ts836000406j ;ts940060406j ;ct1 ;E ;tr
```

Again, the entry ;INIT generates and initialization sequence in the first file of the transmission. For tabular work however, the system will be set up to produce eight point type on ten point line space. The EditWriter will use font one, the line measure is set to 45 picas (seven and one-half inches), and line-end decisions will be made, automatically, by the system. The entries prefaced with ;ts set nine tab positions. The first tab position is programmed to start at the left margin with a length of nine picas. The remaining tabs are assigned a length of four and one-half picas. ;ct1 programs the system to recognize tab codes, ;E ensures that copy will be received in the Edit mode, and ;tr generates a tab return to end the initialize sequence.

Precedence Of Translations

When a string of text characters may be translated by several different table entries, the longest entry that starts with the current character will apply, to the exclusion of others. Characters translated by one entry may not also be translated by other, perhaps overlapping, entries. Consider these examples.

```
/=//==//  
/A23=BC/.....Entry 1  
/A234=BCD/.....Entry 2  
/A345=CDE/.....Entry 3  
/A3456=CDEF/...Entry 4  
  
/I
```

To illustrate, let's assume that the number 23456 has been sent from the word/data processor.

The EditWriter would receive the string BCD56 because entry 2 /A234 = BCD/ is the longest define statement that applies.

Entry 1 (/A23 = BC/) would not be used because it is shorter than entry 2. Entries 3 and 4 would not be used because the characters 3 and 4 are removed from the data stream when entry number 2 is processed.

Table entries may be typed in any sequence. The ICI sorts them and determines their order of precedence.

Deleting Codes

1 to 64 word/data processing characters or codes can be deleted from the text stream by writing a define statement that equates them to a FF hex (delete).

For example, if you wish to create a table entry that deletes a series of codes, you would write a statement like this:

```
/Hcat=FF/.....cat is deleted
```

With this entry in the Translation Table, if the word processor transmitted "cathode", the EditWriter would receive only "hode".

If you were to transmit the define statement:

```
/V0D=FF/.....Carriage return is deleted
```

and followed by this text:

```
Line number one  
Line number two  
Line number three
```

the EditWriter would receive:

```
Line number oneLine number twoLine number three
```

A more practical example would be the deletion of an underscore within a word processing document. Most type-setters would prefer to change the type style to bold or italic for emphasis. Typically, an underscored character is transmitted as character, backspace, underscore or underscore, backspace, character. As a result these table entries are suggested:

```
/H_=08/.....Underscored space becomes space (EBCDIC)  
/V166D=FF/.....Backspace, underscore is deleted (EBCDIC)  
/V6D=08/.....Underscore becomes space  
/V085F=FF/.....Backspace, underscore is deleted (ASCII)  
/V5F08=FF/.....Underscore, backspace is deleted (ASCII)  
/V5F=08/.....Underscore becomes space (ASCII)
```

Space Compression

Unless a string of space characters is defined to be translated in a specific way (for example, into a tab), the ICI will reduce them to a single word space. That is, 2, 3, or more adjacent space characters will be automatically reduced to a single space by the ICI's Default Table (See the Appendix).

Space compression is the last logical function that is performed before the incoming data enters EditWriter screen memory, so it is also performed on characters that have been converted into spaces.

Space compression may be turned off with the table entry:

/H =48/.....Space to shifted space

When To Make A Translation Table

Creating new Translation Tables is not always appropriate. Sometimes it is far easier to use one of the standard Translation Tables and add the necessary typesetting codes at the EditWriter. This is particularly true of short jobs that do not require a lot of coding in the first place. It is also true of jobs that are very complex or irregular in their format.

If you consider all text coming from word and data processors and categorize them first by length and frequency and second by regularity of format we can make the following generalizations.

Where to do the coding: at the EditWriter (EW) or at the word processor through the use of a Translation Table and the ICI.

TOTAL VOLUME COMPLEXITY	JOB LENGTH x FREQUENCY		
	Short	Medium	Long
Low	EW	ICI	ICI
Medium	EW	Either	ICI
High	EW	EW	ICI

It is quite possible to typeset a document using a Translation Table, without imbedding codes in the text. The process involves a careful analysis of the word processing or computer document to identify the aspects of the format, and/or content of the text, that can be uniquely associated with the proper typesetting codes. Once the relationships are identified, a Translation Table is created. It is important to understand that such a document can be edited without affecting the ability to re-typeset it if the format remains the same and if the elements of its content that are used to generate typesetting parameters also remain the same.

The following are examples of two short tables. The first table enables several of these paragraphs to be typeset. It uses the parts of the general purpose TTY text table (only the parts used are reproduced here) because it is a long job of low complexity.

The second table uses the format and content of a standard memo in order to automatically generate typesetting parameters. The assumption is that this text is transmitted with a bisync protocol. It would be appropriate to construct a customized table for this second job (of medium complexity) if the number of memos to be typeset was high (long total length).

Example One

```
/=//==//  
  
/V060606=BCF8CED63B0DD23BD42D2B2D/....Section Heads -- 3 returns become  
QL, TR, +LS, Sz 18, LS060  
/H:rule=90D63B27D42D2D2109842DF809842DD43B2B2DF890/  
          Inserts rule under Section Heads  
/V0606=BCF8CED63B15/.....Sub Heads -- Two returns become Quad  
QL, TR, +LS, Sz 14  
/V0609=BCF88A15D63B270B0B0B/.....Text -- Return, tab becomes QL,  
Tab Return, CT4, Sz 12, 3 Em spaces  
/H;ts=FA/.....Mnemonic for Tab Set  
/H;ct=8A/.....Mnemonic for Call Tab  
/H;tr  
=F8/.....Mnemonic for Tab Return  
/H;E=AC/.....Initiates Edit Mode  
/H;INIT=8CDE3B2D3B272D3B030F2D2D48/...Initializes the EditWriter  
  
/I;INIT;E;tr
```

Example Two

```
/=//==//  
  
/HMEMO=D203D63B272D78427870/. . . . . MEMO becomes Font 3, Size 12,  
                                         MEMO  
/V060606=3DF8CED63B2D/. . . . . Three returns become Quad Center,  
                                         Tab Return, Plus Line Space, Size 10  
/H:=65D23BBC/. . . . . Colon becomes colon, Font 1, Quad Left  
/V0606=BCF8CE/. . . . . Two req'd returns become Quad Left,  
                                         Tab Return, Plus Line Space,  
/V06=BCF8D203/. . . . . Req'd return becomes Quad Left,  
                                         Tab Return, Font 3  
/V05=1B/. . . . . Tab becomes Tab  
/H "=085050/. . . . . Space, quote becomes space, open quotes  
/H". =10103908/. . . . . Quote, period, space becomes close quotes,  
                                         period, space  
/H;ts=FA/. . . . . Mnemonic for Tab Set  
/H;ct=8A/. . . . . Mnemonic for Call Tab  
/H;INIT=8CDE3B273B272D0327272D2D48AC/.Initializes the EditWriter  
  
/I;INIT;ts108001400j;ct1
```

The text of the memo would be:

MEMO

TO: The Wicked Witch of the West
FROM: The Scarecrow
DATE: June 19, 1939
SUBJECT: Rude and uncalled for treatment by castle employees

Madam, I am writing in reference to the treatment I recently received from some people affiliated with your organization.

On my recent journey to Oz, I quite innocently wandered into one of your forests, and was immediately descended upon by a rather large number of your employees. Their approach to my predicament was, in my opinion, totally uncalled for. First they took my legs and threw them over there...then they took my arms and threw them over there. To add insult to injury, they flew away chanting "rest in pieces". Furthermore...

See also: "Recommended Techniques for Customizing Tables"

Representing Word Processor And EditWriter Codes

The codes and characters of both the word processor (or computer) and the EditWriter can be represented in two ways. First, they can be typed as they appear on the screen or they can be represented by hexadecimal numbers that are composed of pairs of hexadecimal digits. Hexadecimal numbers are used to represent all the codes that cannot be conveniently represented on the screen of the word processor.

Look at the EditWriter code set chart in the Appendix. You will see that the "Font" function is represented by a D2 hex. You read the "D" across the top of the chart, and the "2" vertically on the left side of the chart.

If you look at the EditWriter Keyboard Chart (See the Appendix), in the upper right corner where the Font key is located, you will see a "D2" in the box that represents that key.

Likewise, if you look at the EBCDIC/WP and ASCII code charts, you can see that there are Hex codes that represent the characters that you type on the screen of the word processor.

The ICI will translate from either the EBCDIC/WP or ASCII code sets, but the decision as to which Default Table is used depends on whether you select Asynchronous or Synchronous communications.

Take a look at the EBCDIC code chart. The letter "B" has a value of "C2". Look at the ASCII code chart. The letter "B" has a value of "42". If you are going to use the V or R designators, you will be using these charts to find out which Hex codes represent which characters. But remember that you use an EBCDIC code chart if you are using Bisync (Synchronous) communications, and the ASCII code chart if you are using TTY (Async) communications. Typically, you will use the EBCDIC code set with 2770 and EBCDIC/DP with 2780, 3741, and 3780.

Although hex codes can represent both characters and function codes, you will notice that hex codes most often will be used to represent functions.

The Hexidecimal Numbering System

The hexidecimal digits are:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

The corresponding decimal values are as follows:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

A Hex number (or code) can have any value between 00 (0 decimal) and FF (255 decimal).

Recommended Techniques For Customizing Tables

There are six General Purpose Translation Tables that can be used when a customized version does not exist for a specific word processor or computer. There are "text" and "tabs" tables for Asynchronous ASCII and Synchronous EBCDIC. The following is an explanation of the principles of code translation embodied in these tables.

Handling Tabs

When a word processor sends a tabular document, it may transmit tab codes, or it may send space characters to simply position the text to the right tab position.

Word processors that use the TTY protocol will generally send space characters instead of tab codes. Sometimes, they use the horizontal tab code (09 hex). The ICI normally compresses multiple adjacent space characters into a single space to facilitate justification. If you are not sure about the way the word processor sends tab codes, you would have these entries.

```
/V09=1B/.....Tab becomes Tab  
/H    =1B/  
/H    =1B/  
/H    =1B/  
Etc.  
/H          =1B/....3 to 20 adjacent spaces become Tab
```

When receiving text from a word processor using a Bisync protocol, the situation is generally the same and there will be need for entries that translate space characters into tabs. If in doubt about the way any particular word processor handles tabs, check with the vendor. However, the value and meaning of the tab code needs to be understood.

Word processors with Synchronous communications often transmit a horizontal tab code with the value 05 hex. Some also transmit an indent tab with the value 39 hex. The text will follow this pattern:

(05) This is text that is indented with a horizontal tab and which word wraps back to the left margin.

(39) This is text that is indented with an indent tab and which word wraps to the indent level.

In the first case you would want to translate the tab code into an indent—perhaps an Em Space. In the second case, you would want to translate the indent tab to a tab call. Devising a good series of translations depends upon finding some regular aspect of the format of the document that shows which tabs are to be preserved. If, as in the case with Wang, the carriage returns have the value 0D hex, we could have these entries:

```
/V05=1B/.....Horizontal tab becomes Tab  
/V39=1B/.....Indent tab becomes Tab  
/V0D=BCF8/.....Return becomes Quad Left, Tab Return  
/V0D05=BCF80B/.....Return, horizontal tab becomes Quad Left, Tab Return,  
Em space
```

Most word processors transmit space characters around numbers that are aligned on their decimal points. The best strategy for preserving decimal tabs is to convert multiple space characters into tab codes and then to create a series of ten entries that convert periods followed by a number into another tab call like this:

```
/H.0=8A27392D/.....Period, 0 becomes Call Tab 2, period 0  
/H.1=8A27393B/.....Period, 1 becomes Call Tab 2, period 1  
/H.2=8A273927/.....Period, 2 becomes Call Tab 2, period 2  
Etc. to 9
```

Of course, tab one would have to be defined as a right justified floating tab and tab two as a left justified floating tab. In this example:

1.23 34.8 6.900

becomes:

(tab 1)1(tab 2).23(tab 1)34(tab 2).8(tab 1)6(tab 2).900

The complete Translation Table is in the Appendix.

When aligning columns of numbers of unequal length, the word processor's required space can have exactly the same function as the EditWriter En space. Often, a required space is called a coded space.

What To Do About Line Endings

Perhaps the most important part of the translation from word processor to EditWriter format is the correct translation of the line endings that are indicated at the word processor.

Word processors format text for printers, and frequently do so using mono-spaced characters. Occasionally, word processors can output to proportionally spaced printers, and line endings will reflect that capability. When you typeset text on the EditWriter, you want to retain only those line endings that are required—not simply the line endings that were determined arbitrarily by the need to keep words from spilling over into the word processor's right margin. Those familiar with the typesetting process, realize that the EditWriter will make the line ending decisions based on the font, point size, and line length specified.

Line endings are indicated by carriage returns. Word processors that transmit using the TTY protocol, will send returns for every line whether blank or not. They may also follow the return code with a series of space characters that

represent the left and right margins. Those that use the Bisync protocols will often send two kinds of return codes: ordinary returns, which are translated to space by the Default Table, and required returns, which are automatically translated into the return function on the EditWriter).

Required Returns

An example of a required line ending is the end of this paragraph, which will not extend all the way to the right margin.

- A . Required returns end short lines.
- B . Required returns end paragraphs.
- C . Required returns do not normally occur in the middle of a paragraph or sentence.

The required return codes that should be retained are those that appear after the heading, after the third line of the paragraph, after the end of sentences A and B, and after the second line of sentence C. In addition, the fact that blank lines were skipped with return codes is also important to know, and should be reflected in the translation.

For an EBCDIC (bisync) device, the table entries might look like this:

```
/V06=BCF8/.....Return becomes Quad Left, Tab Return  
/V1E=08/.....Non-required return becomes space
```

Since ASCII (TTY) devices transmit only one type of return, the table entries might look like this:

```
/H  
=08/.....Return becomes space  
/H  
  
=BCF8CE/.....Two returns become Quad Left,  
Tab Return, Plus Line Space
```

Thus a short line, or a paragraph end, would only be recognized by the EditWriter if it was followed by 2 return codes. It should also be noted that a single ASCII return (0D

hex) is automatically translated to space by the ASCII Default Table and is, therefore, not required in the Translation Table.

Whenever you have a situation where a word processor or computer does not transmit tabs, you must map multiple adjacent word spaces into an EditWriter tab. When, in addition, you are receiving text that is image processed, with multiple spaces used to represent left or right margins, special table entries are needed to properly translate line endings. TTY table entries might look like this:

```
/H =08/.....Return and three spaces becomes space  
/H =08/.....Return and four spaces becomes space  
/H =08/.....Return and five spaces becomes space  
Etc., up to the number of spaces required to account for the longest possible margin and indented tab.
```

For a bisync transmission, a series of V format translations would be required like this:

2770 or 3780

```
/V1E404040=08/.....Non-required return and 3 spaces becomes word space  
/V1E40404040=08/....Non-required return and 4 spaces becomes word space  
Etc.
```

The process of handling returns properly is greatly simplified if the left margins of all documents are set to zero prior to transmission.

When copy that is primarily tabular in nature is transmitted from a word processor to the EditWriter, it useful to treat return codes differently.

When using Asynchronous (ASCII) devices, every return code (0D hex) will be translated to a Tab Return (F8 hex) and Tab (1B hex). With EBCDIC devices, a required or code return (06 hex) becomes a Tab Return, Tab (F81B)

while an ordinary return (1E or 1F hex) becomes an EditWriter Return (29). When these translations are present in the table, you may typeset copy that looks like this:

Section	Number	Number	Number
ABC	123	456	789
DEF	012	345	678
GHI	901	234	567

without adding tabs or tab returns at the EditWriter.

Like tabs, most word processors accomplish the centering function by embedding the copy in space codes to properly position it between margins. This applies whether the transmission was in ASCII or EBCDIC. Occasionally, a word processor will transmit a code prior to a line that indicates that the line is to be centered. In this case, it is possible to translate the center code into a precedence C.

The Wang word processor, for example, transmits a center code which is assigned a 20 hex value.

/V20=845C/.....Center code becomes Precedence C

The line ending code would have to reintroduce a Precedence J to turn off this command.

Hyphenation

When receiving text from an Asynchronous word processor, there is no distinction made between required and discretionary hyphens. The Default Translation preserves all hyphens as required hyphens. When the text is rejustified on the EditWriter, many of these hyphens may need to be edited out. Alternatively, you can make a table entry that converts any hyphen that precedes a return code, to a Discretionary Hyphen. Such a table entry would be:

/H-
=05/.....Hyphen and return become Discretionary Hyphen

Some Bisync word processors, like the Xerox 850 and IBM Office System 6, make a distinction between types of hyphens. The Default Table translates a syllable hyphen (CA hex) to a Discretionary Hyphen, and preserves required hyphens (hex value 60) as part of the text.

Things To Remember About Quotes

A large part of the standard Translation Tables provided by Compugraphic, is taken up with the conversion of word processing quote symbols (which happen to be the same for open and close quotes). The approach taken by the Default tables is to convert all quotes to close quotes with specific exceptions. These are quotes that: (1) begin a line, (2) follow a space or tab, or (3) are adjacent to open brackets or parens. It's important to note that, the portions of the standard tables that deal with quotes, should be preserved in a custom table.

Dot Leaders

The standard Translation Table translates a series of four, seven or ten periods or hyphens to an EditWriter Insert Character command and a period. The number to search for is fairly arbitrary and can be adjusted to fit the job.

/H....=0939/.....Four periods become Insert Character, period

Paragraph Indents

Paragraph indents can be input in many ways at the word processor. Some word processor operators, for example, may indent a paragraph by typing 1 to 5 spaces before typing the first line, while others may use a horizontal tab.

Whatever the method, a paragraph indent is always preceded by a return and is, therefore, unique.

Thus, these table entries are suggested.

```
/H =BCF81D/.....Return and 3 spaces becomes Quad Left, Tab Return, En  
Space  
/H =BCF81D/.....Return and 2 spaces becomes Quad Left, Tab Return, En  
Space  
/H =BCF81D/.....Return and 1 space becomes Quad Left, Tab Return, En  
Space  
/H =0B/.....Two spaces become Em Space
```

As a result or these table entries, indents of 1 to 3 spaces will be translated to an En indent at the EditWriter. If the word processor operator input 5 spaces at the beginning of each paragraph, the ICI word convert them to an En and Em indent. If within the same table, 3 space characters have been equated to a tab, an indent of 6 spaces would become a tab.

Special Printer Control Codes

Because word processors are designed to run printers, printer control codes may be embedded in the text that is sent to the ICI. Most of these codes are not useful to the EditWriter, and are automatically discarded through the Default Tables. If desired, you can create Translation Table entries that will preserve them. An example would be to use stop codes to call for a font change.

Let's assume that you wish to typeset this paragraph:

(Stop)Special Printer Control Codes

(Stop)Because word processors are designed to run printers, printer control codes may be embedded in the text that is sent to the ICI. Most of these codes are not useful to the EditWriter, and are automatically discarded through the Default Tables. If desired, you can create Translation Table entries that will preserve them. An example would be to use stop codes to call for a font change.

Your Translation Table entries might look like this:

/V2F=D203/.....Stop code becomes Font 3
/V06062F=3DF8CED23B/....Two required returns and a stop code become Quad Center, Tab Return, Plus Line Space, Font 1

The result of these entries would be:

Special Printer Control Codes

Because word processors are designed to run printers, printer control codes may be embedded in the text that is sent to the ICI. Most of these codes are not useful to the EditWriter, and are automatically discarded through the Default Tables. If desired, you can create Translation Table entries that will preserve them. An example would be to use stop codes to call for a font change.

In addition, many systems transmit page or format codes, along with definition of tab fields. If this data is in a readily definable format it is quite easy to use it to define and call EditWriter parameters.

Upper Case to Lower Case

Many data processing systems will have files that need to be typeset, but whose text is all upper case. In some circumstances, the ICI can be set up to translate the alphabet to lower case. In the following example, the first character of every word (that is every character following a space) remains upper case, while all other alphabet characters are translated into lower case.

Note how the following table converts the upper case alphabet to lower case. Then we'll take it one step further and translate all upper case letters preceded by space to upper case.

```
/=//==//  
  
/A A= A/A B= B/A C= C/A D= D/A E= E/A F= F/A G= G/A H= H/A I= I/  
/A J= J/A K= K/A L= L/A M= M/A N= N/A O= O/A P= P/A Q= Q/A R= R/  
/A S= S/A T= T/A U= U/A V= V/A W= W/A X= X/A Y= Y/A Z= Z/  
/AA=a/AB=b/AC=c/AD=d/AE=e/AF=f/AG=g/AH=h/AI=i/AJ=j/  
/AK=k/AL=l/AM=m/AN=n/AO=o/AP=p/AQ=q/AR=r/AS=s/AT=t/  
/AU=u/AV=v/AW=w/AX=x/AY=y/AZ=z/  
/H  
  
=BC29CE/  
/H;INIT=8CDE3B2D3B272D3B27272D2D081BAC/  
  
/I;INIT
```

These entries will take you from copy like this:

IF TEXT OR DATA IS COMING TO THE EDITWRITER FROM A DEVICE THAT CREATES ONLY UPPER CASE CHARACTERS, YOU CAN CREATE A TRANSLATION TABLE THAT AUTOMATICALLY CONVERTS ALL BUT THE FIRST CHARACTER OF EACH WORD INTO LOWER CASE.

to copy like this:

If Text Or Data Is Coming To The Editwriter From A Device That Creates Only Upper Case Characters, You Can Create A Translation Table That Automatically Converts All But The First Character Of Each Word Into Lower Case.

Optionally, a Translation Table could be set up to translate all alpha characters into lower case, except where they follow a period and a space. The possibilities are endless, but the important thing to remember when creating these types of Translation Tables, is that you must not also use the spaces before words to simultaneously generate typesetting parameters.

Removing Page Headers

Frequently, when a word processor is transmitting a multi-page document, extraneous information will be transmitted between pages. This should be avoided if possible.

Typically, this extra material consists of such things as format lines, which define tab settings on the word processor, and/or headers, footers, and page numbers. This material is removed to simplify the editing and repagination process at the EditWriter.

The suggested approach is to use the appropriate code identification table (See the Appendix) with a multi-page test, and to look for patterns that can be deleted through the Translation Table.

Generating Multijust Commands

When typesetting tabular material that is generated on a word processor with two kinds of tabs—horizontal tabs (ordinary tabs) and indent tabs (coded or required tabs), the indent tab not preceded by a return can be translated to an EditWriter Multijust command, and the horizontal tab can be converted to a normal Tab command.

/V05=1B/.....Horizontal tab becomes Tab
/V39=37/.....Indent tab becomes Multijust

Margins, Line Feeds, and Double Spacing

One of the most confusing problems that can arise, occurs when the text being sent by a word processor is image processed. This means that the receiving system (the EditWriter) is treated like a printer that cannot handle tab instructions.

By way of explanation, let's assume that the left margin of the document being sent is set at 10. If the document was image processed, a string of 10 space characters (sometimes called "leading spaces") will be transmitted after the return code to account for the indent from the left side of

the paper. Right margins are treated in a similar manner. This means that Translation Table entries that include return codes may be incorrect. For example, you might have an entry in the table that looks like this:

/VODOD=BCF8CE/.....Two returns become Quad Left, Tab Return,
Plus Line Space

But the sending system may be sending ten spaces after each return, like this:

/VOD2020202020202020D2020202020202020=BCF8CE/

The EditWriter never sees two returns (0D0D) in a row and therefore, cannot replace the entry with the appropriate Quad Left, Tab Return, Plus Line Space.

Only caution and testing can avoid surprises like this. The best alternative is to use H format translations when possible, like this:

/H

=BCF8CE/.....Two returns become Quad Left, Tab Return, Plus
Line Space

Now, the EditWriter fills in the left side of the table entry with all the spaces, returns, and line feeds (if any) that it may receive. You don't have to worry about the details.

Similarly, if the sending device sends a line feed (0A hex) with each return code, the V format table entries might have to be written like this:

/VODOA0DOA=BCF8CE/.....Return, line feed, return, line feed becomes
Quad Left, Tab Return, Plus Line Space

Again, using H format entries solves this problem.

Note: H format statements cannot be used for return when the word processors Bisync protocol sends every line as an 80 character record.

Double spacing is trickier. Some word processors will send two carriage returns to end each line, and three or four

carriage returns to end a paragraph. Only testing with a sample document can solve this problem. But you may wind up with table entries like this:

/H

=08/.....Two returns to word space
/H

=BCF8/.....Three return to Quad Left, Tab Return

See the sections "Dealing with Tabs" and "Handling Line Endings" for more information on how image processing and sending spaces for tab codes complicates the task of defining line endings.

Preparing Copy For The ICI

A document—either text or data—can be structured as it is created, or modified afterwards, to lessen the amount of editing that needs to be done at the EditWriter. Whether this happens depends on the total labor savings, which can only be evaluated on a job-by-job basis.

Two major applications are worth exploring: (1) typesetting large, multi-page text jobs, usually manuals or books, and (2) typesetting reports, inventory lists, price catalogs, and similar data acquired in a data entry environment.

Large Text Applications

Two approaches should be considered for two different situations. First, there is the document that exists already and has been formatted for printer output. Secondly, there is the document that is input with the knowledge that it will be typeset. Printer output will only be used for proofing.

Approach 1

Take a representative sample of the document that is two pages long. Send it to the EditWriter as a test with a standard text table before it.

Look for cases where the text may be consistently mistranslated. Remember, the main point is to retain the basic format of the document—line endings, paragraph endings, tabs. Check to see that extraneous material does not show up at the top and/or bottom of pages. Many systems have options that send encoded tab and format information that would only be of interest to other, similar word processors.

Modify the standard Translation Table to minimize the amount of editing that will be required when the total document is transmitted.

Approach 2

Keyboard the document with rigorous consistency. Start paragraphs, do your tabbing, start headings, do section numbering etc., in a uniform manner. This will allow you to construct table entries that properly format the text automatically during transmission.

Don't input headers, trailers, page numbers, and footnotes in the text unless you are absolutely sure how the final document will be paginated. Remember, typeset pages will break differently than typewritten pages.

If the typographic format changes frequently and/or irregularly, you might consider the use of typesetting mnemonics to select point size, font, line length and spacing. Or alternatively, identify all the unique combinations of typesetting parameters and call them with format calls (see example).

/==//

/H*1=D63B27D203D43B272DD027272D2D/....*1 inserts header parameters
/H*2=D63B2DD23BD43B2D2DD027272D2D/....*2 inserts body copy parameters
/H
=BC29/.....Required return becomes Quad Left, Return

/I

The word processor copy would look like this:

*1Any Dummy Can Code Copy for Typesetting

*2(1)A Federal task force today released a 2,200 page study that reveals for the first time that using formats to embed typesetting codes is so simple that a 3 year old gorilla can do it.

and the output would appear as follows.

Any Dummy Can Code Copy for Typesetting

A Federal task force today released a 2,200 page study that reveals for the first time that using formats to embed typesetting codes is so simple that a 3 year old gorilla can do it.

Large Tabular Applications

Many organizations have vast quantities of data that is constantly updated, and frequently typeset. Examples include price lists, telephone directories, and parts lists. Two approaches can be used to typeset this data: (1) transmit a standard tabular Translation Table and the raw data files (sorted into appropriate order) to the ICI, and (2) to write a utility and an accompanying Translation Table that formats the data files for the ICI (similar to the way a report is sent to a computer printer).

Approach 1

The data will be sorted so it outputs in the appropriate sequence. The EditWriter will put the data into one or more tab columns, moving left to right, and then down the page.

If a field corresponds to the data in one tab position, and a record corresponds to a typeset line, then fields will be separated by tab codes, or 3 or more spaces. Records will be separated by carriage returns. Blank fields should be represented by a non-print character like a null (00 hex).

The standard Translation Table will convert the tabs or spaces into EditWriter tabs. All returns will become tab return followed by tab. The tab definitions will be done at the EditWriter.

Approach 2

In the second case, the EditWriter can be used almost as a "slave" unit, except possibly for determining page endings and queueing the documents to the photo unit. All required typesetting and formatting information is generated through the combination of a customized Translation Table and a formatting utility.

Consider the third alternative in "Large Text Applications". There, short two character format calls produce typesetting parameters. A similar technique may be used here, but it can be taken further. Because computer reports use the same heading information over and over, you can add table entries like this:

/=//==//

/A*3=Quarterly Financial Summary/
/H*4
=3D29/.....*4 inserts Quad Center, Return
/A*5=Territory(tab)Jan. 81(tab)Feb. 81(tab)Mar. 81(tab)Total/
/H
=F81B/.....Return becomes Tab Return, Tab
/V05=1B/.....Tab to Tab
/H;INIT=8CDE3B2D3B272D3B27272D2D081BAC/
/H;ts=FA/
/H;ct=8A/

/I;INIT;ts1000006001;ts206000400r;ts310000400r;ts414000400r;ts518000400r;ct1

The copy might look like this:

*3*4
*5
102 12,000 15,000 17,000 44,000
103 13,000 14,500 16,500 44,000
104 16,000 17,000 17,500 50,000
105 14,000 15,000 29,000
etc.

The first column position of the fourth row of data is blank. Sending two tab codes causes the position to be skipped.

FINAL ASS'Y NOMINAL THD. SIZE	PART NUMBER BASIC + LENGTH	NET PRICE EACH INSERT FOR QUANTITY INDICATED				
		1-9	10-24	25-49	50-99	100 & UP
1 1/8"-8	4490-18TN	1.125	18.99	15.19	10.63	7.44
		1.688	24.42	19.54	13.68	9.57
		2.250	30.07	24.06	16.84	11.79
						8.62
						10.61

The considerations mentioned above in sections (1), (2) and (3) still apply.

For Big Important Jobs Try A Small Test First

A representative sample—one or two pages—that has the tabs, margins, indents, and headings that you expect to see later, is a good test for a newly keyboarded Translation Table.

Sometimes word processors (or their operators) will be inconsistent in the way they transmit prepared text. If you get unexpected results, you may find it useful to consider these tactics:

- 1) Repeat the test in Terminal Mode. This will help you to see pretty much the same image that appears on the word processor's screen.

- 2) Special translation tables for code identification (See the Appendix) can help you to see exactly what codes the other system is transmitting. Have the appropriate table keyboarded at the word processor. Receive the sample text in the ICI Mode.
- 3) Compare a printed version of the test file with the text received through the ICI.

Coding At The Word Processor

There are many different ways to handle the process of coding text for typesetting. If you wish to utilize the power of the ICI to minimize the coding task at the EditWriter, there are several approaches which you can copy or use for inspiration.

The General Purpose Translation Table, provided by Compugraphic, give you the capability to code text at the word processor by utilizing a set of short mnemonics. Here is a portion of the standard table.

```
/=//==//  
  
/H;s=D6/.....;s## generates size change  
/H;f=D2/.....;f# generates font change  
/H;ls=D4/.....;ls### generates line space change  
/H;l1=D0/.....;l1#### generates line length change  
/H;;=BC/.....;; generates Quad Left Command  
/H;c=3D/.....;c generates Quad Center Command  
/H;m=0B/.....;m generates Em Space  
/H;  
=29/.....; return generates Return  
  
/I
```

Here is a short job using this type of coding.

```
;s10;f3;ls120;l12200
```

Coding With EditWriter Command Mnemonics;c;

;f1;mThe headline above will be centered in 22 picas. It will be set in 10 point bold face (font 3) on 12 points of line space. This body copy is set using font

(light face), and is indented one Em space. This coding technique lets anyone familiar with the EditWriter sit down and start coding a document for typesetting.;;;

And you can access all the command codes of the EditWriter without having to modify the standard table or make a new translation table.;;;

The typeset copy would look like this:

Coding With EditWriter Command Mnemonics

The headline above will be centered in 22 picas. It will be set in 10 point bold face (font 3) on 12 points of line space. This body copy is set using font 1 (light face), and is indented one Em space. This coding technique lets anyone familiar with the EditWriter sit down and start coding a document for typesetting.

And you can access all the command codes of the EditWriter without having to modify the standard table or make a new translation table.

If the text prepared at the word processor is fairly consistent in format or contains a well-defined set of formats, this easy ICI coding technique allows anyone to generate text that is ready for typesetting. Consider this table:

/=//==//
/HSTART =8CDE3B273B152D0327272D2D081BAC/....Size 12, Line Space 140,
Font 3, Line Length 22 picas
/HCENTER =8A3B/.....Call Tab 1
140, Line Length 22 picas
/HTEXT =8A27D63B2DD23B/.....Call Tab 2, Size 10, Font 1
/HBOLD =D203/.....Bold Face
/H BOLD=D203/.....Bold Face
/HITALIC =D227/.....Italic Face
/H ITALIC=D227/.....Italic Face
/HLIGHT =D23B/.....Font 1
/H;ts=FA/.....Tab Set

/I;ts100002200c;ts2000022001

Now an operator can code a document with commands like this:

START CENTER How To Code For The ICI

LIGHT TEXT With about 5 minutes of explanation, anyone capable of using a word processor can begin to BOLD quickly and accurately LIGHT code text that is to be typeset on the EditWriter.

Coding with formats can be streamlined even further if the formats do not need literal meanings. Data may be prepared and maintained on a data entry system. The formats are added by a utility that creates a temporary data file, which is sent to the ICI. The job could be done just as well on a word processor. Below is an example of how it could be done.

```
/=====/  
  
/H;ts=FA/.....;ts generates Tab Set  
/H;tr  
=F8/.....;tr, return generates Tab Return  
/H;f=D2/.....;f# generates Font  
/H;s=D6/.....;s## generates Size  
/H;ls=D4/.....;ls### generates Line Space  
/H;ll=D0/.....;ll### generates Line Length  
/H(1)=8A3B/.....Call Tab 1  
/H(2)=8A27/.....Call Tab 2  
/H(3)=8A03/.....Call Tab 3  
/H(4)=8A15/.....Call Tab 4  
/H(5)=8A21/.....Call Tab 5  
/H(6)=8A2B/.....Call Tab 6  
/H+=D22747D23B/.....Font 2, plus symbol  
/H#=D227862DD23B/.....Font 3, inch mark  
/V1E=FF/.....Non-required return is deleted  
/H  
=FF/.....Required return is deleted  
/H  
  
=F8/.....Two required returns become Tab Return  
  
/I;f1;s10;ls120;ll4500;ts100000906c;ts209060900c;ts318062606c;ts409060500c  
;ts514060400c;ts6f0308c;tr
```

The text for this job might look like this:

(1)FINAL ASS'Y(2)PART NUMBER(3)NET PRICE EACH INSERT FOR QUANTITY
INDICATED
(1)NOMINAL THD. SIZE(2)BASIC + LENGTH (6)1-9(6)10-24(6)25-49(6)50-99(6)100
& UP

(5)1.125(6)18.99(6)15.19(6)10.63(6)7.44(6)6.70
(1)1 1/8"-8(4)4490-18 TN(5)1.688(6)24.42(6)19.54(6)13.68(6)9.57(6)8.62
etc.

Transmitting The Table And Text

A table is written in the following format:

Keyword

Beginning code, type of translation code, codes sent by the word processor, separating code, EditWriter codes, ending code, comments etc.

End Statement

Initialization

text...text...text

The Translation Table has to be transmitted to the ICI before the text it is to encode. The following are all acceptable:

- 1) The Translation Table is written at the beginning of the job to be typeset—right at the top of the first page.
- 2) The Translation Table is page 1 of a document that is 2 or more pages long.
- 3) The Translation Table is a separate file that is transmitted first, before the text file.

A table can translate up to 2,000 characters. Comments are not included in that total. Note that the comments are placed after the ending code and before the beginning code of the next statement.

A table entry can have up to 64 characters representing the codes sent by the word processor, and up to 128 characters representing the codes to be saved by the EditWriter.

A Translation Table may be sent with each file if the ICI is set up in Unattended mode, and the word processor pauses and sends a TTY End of Transmission (EOT-04 hex), or Bisync End of Text (ETX-03 hex) code between transmissions.

Finding And Correcting Translation Table Errors

Table errors are going to occur when the operator of the sending system incorrectly keyboards the Translation Table that you have given them. They also are going to occur when the word processor or computer does something unanticipated or different than planned for in the table. Understanding what is happening in the Translation Table, and how to modify it becomes necessary.

Common Table Errors

Translation Table Errors frequently appear in the following cases:

- 1) A V format translation mistakenly contains alpha (not hex) codes (see "The Hexidecimal Numbering System").
- 2) The keyword (the initial 8 character sequence) is incorrectly entered (See "The Keyword").
- 3) Lower case "L" is used instead of "1" in V format statements.
- 4) Equal signs "=" are left out of statements (See "The Body of the Table").

- 5) Sequences of hexidecimal codes extend over the right hand margin so that they are word wrapped to lie partially on two different lines.
- 6) One of the two digits of a hexidecimal number is left out.
- 7) The letters used to write hexidecimal numbers or to designate types of translations are not capitalized.
- 8) There are spaces between hexidecimal digits.
- 9) Upper case O is used to represent zero in hexidecimal numbers.
- 10) The capital I at the end of the table is missing.

Finding Table Errors

The following are suggested techniques for locating table errors.

- 1) Make sure that the keyword contains eight characters and is arranged in the proper pattern.
- 2) Ensure that a slash (or the first character of your keyword) appears at the beginning and end of every translation statement.
- 3) Ensure that there are an even number of characters in all hex-based translation statements.
- 4) Check to see that an equal sign (or the second character in your keyword) appears once between fields in each hex statement.
- 5) Ensure that all designators and alpha characters in hex translation statements are capitalized.
- 6) Search the table for lower case L's and Oh's. Replace with ones and zeros.

- 7) If the error is not readily noticeable, move the /I (end of Table code) up to the 10th line of the table and transmit again. Consult your error reading. If no errors are reported, the first 10 lines of the Table are error-free. Move the /I down another 10 lines and repeat the process. Continue this procedure until the location of all table errors has been isolated. Once your attention is focused on the error, it is easily corrected.

APPENDIX

EditWriter Keyboard Chart

The following chart is used when you wish to find the hex value of a specific character or function on the EditWriter keyboard. When two hex numbers are shown on the same key, the lower value will represent the unshift position, while the higher value represents the shift condition. EditWriter function commands are not case sensitive.

EDITWRITER KEYBOARD HEX CODES

90 CHAR COMP	88 MEAS DEPTH	80 CTL	D0 LINE LENGTH	C8 LINE SPACE	C0 FWD																								
92 SP	8A CALL TAB	82 CHAR COMP SET	D2 FONT	CA PT SPACE	C2 BACK																								
94 AUTO END	8C INIT	84 PR UC	D4 LINE SPACE	CC + PT SPACE	C4 ONLY																								
96 AUTO HY	8E TAB	86 PR LC	D6 SIZE	CE + LINE SPACE	C6 FLASH ONLY																								
			*7B *3B	67 127	# 43 } 03	55 15	© 61 1 21	□ 6B ■ 2B	* 4F ► 0F	% 4D ► 0D	• 71 ' 31	- 6D " 2D																	
9C DEL CHAR	9A DEL LINE	98 DEL PAR	BC OL	1B TAB EL	+ 47 \$ 07	- 7B 1 3B	z 67 2 27	- 43 3 03	x 55 4 15	= 61 5 21	t 6B 6 2B	: 4F 7 0F	§ 4D 8 0D	“ 71 9 31	! 6D 0 2D	! 75 1 35	! 63 ? 23	(7F) 3F	FC CANC CHAR	B6 SRCH	AE DEL FILE	A6 DUP DISK	9E ERASE DISK						
A0 INSERT	A2 ↑	A4 SCRL UP	3D QC	01 THIN SPACE	1D EN SPACE	6E 0 2E	66 W 26	E 42 E 02	R 54 R 14	T 60 T 20	Y 6A Y 2A	U 4E U 0E	I 4C I 0C	O 70 O 30	P 6C P 2C	- 53 - 13	6F 2F	37 MULTI JUST	FA TAB SET	F0 REP SRCH	E8 REPL FILE	E0 SET FILE	D8 DISP						
A8 ← HOME	AA →		1F OR	0B EM SPACE	SHIFT	A 46 A 06	S 4A S 0A	D 52 D 12	F 5A F 1A	G 74 G 34	H 68 H 28	J 56 J 16	K 5E K 1E	L 64 L 24	! 65 ! 25	50 5 10	29 RET	17 INS SPACE	F8 TAB RET	F2 SAVE BLOCK	EA ENTER FILE	E2 INS FILE	DA TYPE SET						
B0 LIMIT	B2 ↓	B4 SCRL DOWN	09 INS CHAR	73 ½ 33	- 51 · 11	SHIFT	Z 62 Z 22	X 7A X 3A	C 5C C 1C	V 7C V 3C	B 72 B 32	N 58 N 18	M 78 M 38	! 59 ! 19	79 39	/ 44 ½ 04	SHIFT	½ 40 ½ 00	05 DH	F4 INS BLOCK	EC CALL FILE	E4 CALL INDEX	DC FILE FEED						
BA CLR MEMORY						48 SPACE BAR												EE REJUST						DE EXECUTE					

EditWriter Hex Chart

The EditWriter Hex Chart is used when you wish to know which EditWriter key or function corresponds to a given hex value. Read the first hex digit across the top of the chart, and the second hex digit down the left side. The intersection of the two rows will contain the appropriate character or function.

		FIRST HEX DIGIT															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
SECOND HEX DIGIT	0	1/4	'	t	o	3/4	'	T	O	CTL	CHAR COMP	INSERT	LIMIT	FWD 1 UNIT	LINE LENGTH	SET FILE	REPL SRCH
	1	THIN SPACE	*	5	9	THIN SPACE	EN -	=	@								
	2	e	d	z	b	E	D	Z	B	CHAR COMP SET	LTR SPACE	CRSOR UP	CRSOR DOWN	BACK 1 UNIT	FONT	INSERT FILE	SAVE BLOCK
	3	3	HYPH	?	1/3	÷	EM —	&	2/3								
	4	1/2	r	I	g	/	R	L	G	PREC U C	AUTO LINE END	SCRL UP	SCRL DOWN	SPACE ONLY	LINE SPACE	CALL INDEX	INSERT BLOCK
	5	DISC HYPH	4	;]	DISC HYPH	TIMES ×	:	[
	6	a	j	w		A	J	W		PREC L.C	AUTO HYPH	DUP DISK	SRCH	FLASH ONLY	SIZE		
	7	\$	INSERT SPACE	2	MULTI JUST	+	INSERT SPACE	±	MULTI JUST								
	8	WORD SPACE	n	h	m	WORD SPACE	N	H	M	CLEAR VERT MEAS	DEL PAR	CRSOR LEFT		LINE SPACE		REPL FILE	TAB RETURN
	9	INSERT CHAR	,	RETURN	•	INSERT CHAR	,	RET	•								
	A	s	f	y	x	S	F	Y	X	CALL TAB	DEL LINE	HOME	CLR DISP MEM	- PT SPACE	TYPE SET	ENTER FILE	TAB SET
	B	EM SPACE	TAB EL	6	1	EM SPACE	TAB EL	†	-								
	C	i	c	p	v	I	C	P	V	INIT	DEL CHAR	CRSOR RIGHT	QUAD LEFT	PT SPACE	FILM FEED	CALL FILE	CNCL CHAR
	D	8	EN SPACE	0	QUAD CENTER	8	EN SPACE	°	QUAD CENTER								
	E	u	k	q		U	K	Q		AUTO TAB	ERASE DISK	DEL FILE		LINE SPACE	EXE-CUTE	REJUST	
	F	7	QUAD RIGHT	!)	‡	QUAD RIGHT	EN LDR	(

ASCII Code Set Chart

The ASCII Code Set Chart relates to Async communications. To find the ASCII value, scan the chart for the character or command in question. When the character has been located, read the first hex digit across the top of the chart, and the second digit down the left side.

ASCII CODE SET

FIRST HEX DIGIT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	space	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								
A	LF	SUB	*	:	J	Z	j	z								
B	VT	ESC	+	;	K	[k	{								
C	FF	FS	,	<	L	\	l	l								
D	CR	GS	hyphen	=	M]	m	}								
E	SO	RS	.	>	N	^	n	~								
F	SI	US	/	?	O	_	o	DEL								



Codes with "Bit 8 = 1" for even parity

ASCII Default Table

To determine the default translation of an ASCII character, find the hex value of the character or function (see the ASCII Code Set Chart). Locate the corresponding spot in the Default Table to read the character that the EditWriter will store. If both characters are the same, the EditWriter will handle the translation automatically. If you wish to translate the character differently than the ICI, it may be done through the Translation Table.

ICI Default ASCII Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			space	0	@	P	[°] degree	p								
1			!	1	A	Q	a	q								
2			' quote	2	B	R	b	r								
3			÷	3	C	S	c	s								
4			\$	4	D	T	d	t								
5			§	5	E	U	e	u								
6			&	6	F	V	f	v								
7			,	7	G	W	g	w								
8	CNCL CHAR		(8	H	X	h	x								
9)	9	I	Y	i	y								
A		*	:	J	Z	j	z									
B		+	;	K	[k	^{3/4}									
C		,	×	L	^{1/2}	l	‡									
D	space	-	hyphen	=	M]	m	^{1/4}								
E		.	†	N	±	n	^{1/3}									
F		/	?	O	^{EM}	—	o									

EBCDIC/WP Code Set Chart

The EBCDIC Code Set Chart relates to Synchronous communications. To find the EBCDIC value, scan the chart for the character or command in question. When the character has been located, read the first hex digit across the top of the chart, and the second digit down the left side.

EBCDIC/WP CODE SET

FIRST HEX DIGIT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE			SP	&	RHY							2	3	1/4 0
1	SOH	DC1			RSP		/	a j		^o degree		A J		NSP	1	
2	STX	DC2		SYN				b k s			B K S		2			
3	ETX	DC3	WUS	IRT				c l t			C L T		3			
4								d m u			D M U		4			
5	HT	NL	LF					e n v			E N V		5			
6	RCR	BS	ETB	NRS				f o w			F O W		6			
7	DEL		ESC	EOT				g p x			G P X		7			
8				SBS				h q y			H Q Y		8			
9	SPS			IT				± i r z			I R Z		9			
A	RPT	UBS	SW	EOP	[]	1/2	:				SHY					
B			CU2		.	\$,	#								
C	FF				§	*	%	@								
D		IGS	ENQ	NAK	()	—	'									
E		IRS			+	;	¶	=								
F		ITB	BEL		!	¢	?	"								

EBCDIC Default Table

To determine the default translation of an EBCDIC character, find the hex value of the character or function (see the EBCDIC Code Set Chart). Locate the corresponding spot in the Default Table to read the character that the Edit-Writer will store. If both characters are the same, the Edit-Writer will handle the translation automatically. If you wish to translate the character differently than the ICI, it may be done through the Translation Table.

ICI Default EBCDIC Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0					space	&	hyphen							$\frac{2}{3}$	$\frac{3}{4}$	$\frac{1}{4}$	0
1					EN		/	a	j	$^{\circ}$ degree			A	J	TH	1	
2								b	k	s			B	K	S	2	
3				RET				c	l	t			C	L	T	3	
4								d	m	u			D	M	U	4	
5				+LS				e	n	v			E	N	V	5	
6	RET	CNCL CHAR						f	o	w			F	O	W	6	
7								g	p	x			G	P	X	7	
8				+PT				h	q	y			H	Q	Y	8	
9	-PT	space						\pm	i	r	z		I	R	Z	9	
A					[]	$\frac{1}{2}$:						Disc Hyph				
B					.	\$,	\div									
C					$\$$	*	\ddagger	@									
D					()	$\frac{EM}{—}$,	quote									
E	space				+	;	\times	=									
F					!	\dagger	?	‘ quote									

ASCII Code Identification Table (Debug Table)

The ASCII Code Identification Table is a big help when you're having trouble with your Translation Table because you don't know what hex values are being transmitted by the sending device. For example, does the sending device transmit a carriage return (0D) the end of each line, or a line feed (0A), or maybe a carriage return as well as a line feed (0D0A)? The answers to these questions are important if you're going to write an effective Translation Table.

Sending a Debug Table prior to the document to be analyzed, allows the EditWriter operator to see exactly what is being transmitted. The alpha-numeric characters and punctuation will be displayed normally, but function commands such as returns, line feeds, tabs, etc., will be displayed in hex. If an Async device is transmitting a carriage return and line feed on the end of every line for instance, the EditWriter screen will display the line of text followed by “-0D--0A-”.

Note: This table is used when dealing with “Standard ASCII”, that is, seven data bits and one parity bit.

```
/=//==// Standard test table -- ASCII Debug 0011A  
  
/R00=.00-/ /R01=.01-/ /R02=.02-/ /V03=48392D0329/  
/R04=.04-/ /R05=.05-/ /V06=48392D3B29/ /R07=.07-/  
/R08=.08-/ /R09=.09-/ /R0A=.0A-/ /R0B=.0B-/  
/R0C=.0C-/ /V0D=48392D5229/ /R0E=.0E-/ /R0F=.0F-/  
  
/R10=.10-/ /R11=.11-/ /R12=.12-/ /R13=.13-/  
/R14=.14-/ /R15=.15-/ /R16=.16-/ /R17=.17-/  
/R18=.18-/ /R19=.19-/ /R1A=.1A-/ /R1B=.1B-/  
/R1C=.1C-/ /R1D=.1D-/ /V1E=48393B4229/ /R1F=.1F-/  
  
/V20=48/ /R21=!/ /R22=/ /R23=#/  
/R24=$/ /R25=%/ /R26=&/ /R27='/  
/R28=(/ /R29=)/ /R2A=*/ /R2B=+/  
/R2C=,/ /R2D=-/ /R2E=./ /R2F=.2F-/  
  
/R30=0/ /R31=1/ /R32=2/ /R33=3/  
/R34=4/ /R35=5/ /R36=6/ /R37=7/  
/R38=8/ /R39=9/ /R3A=: /R3B=;/  
/R3C=.3C-/ /R3D=.3D-/ /R3E=.3E-/ /R3F=?/  
/R40=@/ /R41=A/ /R42=B/ /R43=C/  
/R44=D/ /R45=E/ /R46=F/ /R47=G/  
/R48=H/ /R49=I/ /R4A=J/ /R4B=K/  
/R4C=L/ /R4D=M/ /R4E=N/ /R4F=O/
```

```

/R50=P/      /R51=Q/      /R52=R/      /R53=S/
/R54=T/      /R55=U/      /R56=V/      /R57=W/
/R58=X/      /R59=Y/      /R5A=Z/      /R5B=[/
/R5C=.5C-/   /R5D=]/     /R5E=.5E-/   /R5F=.5F-/
/R60=.60-/   /R61=a/     /R62=b/     /R63=c/
/R64=d/     /R65=e/     /R66=f/     /R67=g/
/R68=h/     /R69=i/     /R6A=j/     /R6B=k/
/R6C=1/     /R6D=m/     /R6E=n/     /R6F=o/
/R70=p/     /R71=q/     /R72=r/     /R73=s/
/R74=t/     /R75=u/     /R76=v/     /R77=w/
/R78=x/     /R79=y/     /R7A=z/     /R7B=.7B-/
/R7C=.7C-/   /R7D=.7D-/   /R7E=.7E-/   /R7F=.7F-/
/R80=.80-/   /R81=.81-/   /R82=.82-/   /R83=.83-
/R84=.84-/   /R85=.85-/   /R86=.86-/   /R87=.87-
/R88=.88-/   /R89=.89-/   /R8A=.8A-/   /R8B=.8B-
/R8C=.8C-/   /R8D=.8D-/   /R8E=.8E-/   /R8F=.8F-
/R90=.90-/   /R91=.91-/   /R92=.92-/   /R93=.93-
/R94=.94-/   /R95=.95-/   /R96=.96-/   /R97=.97-
/R98=.98-/   /R99=.99-/   /R9A=.9A-/   /R9B=.9B-
/R9C=.9C-/   /R9D=.9D-/   /R9E=.9E-/   /R9F=.9F-
/RA0=.A0-/   /RA1=.A1-/   /RA2=.A2-/   /RA3=.A3-
/RA4=.A4-/   /RA5=.A5-/   /RA6=.A6-/   /RA7=.A7-
/RA8=.A8-/   /RA9=.A9-/   /RAA=.AA-/   /RAB=.AB-
/RAC=.AC-/   /RAD=.AD-/   /RAE=.AE-/   /RAF=.AF-
/RB0=.B0-/   /RB1=.B1-/   /RB2=.B2-/   /RB3=.B3-
/RB4=.B4-/   /RB5=.B5-/   /RB6=.B6-/   /RB7=.B7-
/RB8=.B8-/   /RB9=.B9-/   /RBA=.BA-/   /RBB=.BB-
/RBC=.BC-/   /RBD=.BD-/   /RBE=.BE-/   /RBF=.BF-
/RC0=.C0-/   /RC1=.C1-/   /RC2=.C2-/   /RC3=.C3-
/RC4=.C4-/   /RC5=.C5-/   /RC6=.C6-/   /RC7=.C7-
/RC8=.C8-/   /RC9=.C9-/   /RCA=.CA-/   /RCB=.CB-
/RCC=.CC-/   /RCD=.CD-/   /RCE=.CE-/   /RCF=.CF-
/RD0=.D0-/   /RD1=.D1-/   /RD2=.D2-/   /RD3=.D3-
/RD4=.D4-/   /RD5=.D5-/   /RD6=.D6-/   /RD7=.D7-
/RD8=.D8-/   /RD9=.D9-/   /RDA=.DA-/   /RDB=.DB-
/RDC=.DC-/   /RDD=.DD-/   /RDE=.DE-/   /RDF=.DF-
/RE0=.E0-/   /RE1=.E1-/   /RE2=.E2-/   /RE3=.E3-
/RE4=.E4-/   /RE5=.E5-/   /RE6=.E6-/   /RE7=.E7-
/RE8=.E8-/   /RE9=.E9-/   /REA=.EA-/   /REB=.EB-
/REC=.EC-/   /RED=.ED-/   /REE=.EE-/   /REF=.EF-
/RF0=.F0-/   /RF1=.F1-/   /RF2=.F2-/   /RF3=.F3-
/RF4=.F4-/   /RF5=.F5-/   /RF6=.F6-/   /RF7=.F7-
/RF8=.F8-/   /RF9=.F9-/   /RFA=.FA-/   /RFB=.FB-
/RFC=.FC-/   /RFD=.FD-/   /RFE=.FE-/   /RFF=.FF-
/I

```

EBCDIC Code Identification Table (Debug Table)

The EBCDIC Code Identification Table is a big help when you're having trouble with your Translation Table because you don't know what hex values are being transmitted by the sending device.

Sending a Debug Table prior to the document to be analyzed, allows the EditWriter operator to see exactly what is being transmitted. The alpha-numeric characters and punctuation will be display normally, but function commands such as required carriage returns, record separators, tabs, etc., will be displayed in hex. If a Bisync device is transmitting a required carriage return at the end of a paragraph for instance, the EditWriter screen will display the paragraph of text followed by "-06-".

```
/=//==// Standard Test (De-bug) Table -- EBCDIC  
  
/R00= .00-/ /R01= .01-/ /R02= .02-/ /V03=48392D0329/  
/R04= .04-/ /R05= .05-/ /V06=48392D2B29/ /R07= .07-/  
/R08= .08-/ /R09= .09-/ /R0A= .0A-/ /R0B= .0B-/  
/R0C= .0C-/ /V0D=48392D5229/ /R0E= .0E-/ /R0F= .0F-/  
  
/R10= .10-/ /R11= .11-/ /R12= .12-/ /R13= .13-/  
/R14= .14-/ /R15= .15-/ /R16= .16-/ /R17= .17-/  
/R18= .18-/ /R19= .19-/ /R1A= .1A-/ /R1B= .1B-/  
/R1C= .1C-/ /R1D= .1D-/ /V1E=48393B4229/ /R1F= .1F-/  
  
/R20= .20-/ /R21= .21-/ /R22= .22-/ /R23= .23-/  
/R24= .24-/ /R25= .25-/ /R26= .26-/ /R27= .27-/  
/R28= .28-/ /R29= .29-/ /R2A= .2A-/ /R2B= .2B-/  
/R2C= .2C-/ /R2D= .2D-/ /R2E= .2E-/ /R2F= .2F-/  
  
/R30= .30-/ /R31= .31-/ /R32= .32-/ /R33= .33-/  
/R34= .34-/ /R35= .35-/ /R36= .36-/ /R37= .37-/  
/R38= .38-/ /R39= .39-/ /R3A= .3A-/ /R3B= .3B-/  
/R3C= .3C-/ /R3D= .3D-/ /R3E= .3E-/ /R3F= .3F-/  
  
/V40=48/ /R41= .41-/ /R42= .42-/ /R43= .43-/  
/R44= .44-/ /R45= .45-/ /R46= .46-/ /R47= .47-/  
/R48= .48-/ /R49= .49-/ /R4A= .4A-/ /R4B= .4B-/  
/R4C= .4C-/ /R4D= .4D-/ /R4E= .4E-/ /R4F= .4F-/
```

/R50= .50-/	/R51= .51-/	/R52= .52-/	/R53= .53-/
/R54= .54-/	/R55= .55-/	/R56= .56-/	/R57= .57-/
/R58= .58-/	/R59= .59-/	/R5A= .5A-/	/R5B= .5B-/
/R5C= .5C-/	/R5D= .5D-/	/R5E= .5E-/	/R5F= .5F-/
/R60= .60-/	/R61= .61-/	/R62= .62-/	/R63= .63-/
/R64= .64-/	/R65= .65-/	/R66= .66-/	/R67= .67-/
/R68= .68-/	/R69= .69-/	/R6A= .6A-/	/R6B= .6B-/
/R6C= .6C-/	/R6D= .6D-/	/R6E= .6E-/	/R6F= .6F-/
/R70= .70-/	/R71= .71-/	/R72= .72-/	/R73= .73-/
/R74= .74-/	/R75= .75-/	/R76= .76-/	/R77= .77-/
/R78= .78-/	/R79= .79-/	/R7A= .7A-/	/R7B= .7B-/
/R7C= .7C-/	/R7D= .7D-/	/R7E= .7E-/	/R7F= .7F-/
/R80= .80-/	/R81=a/	/R82=b/	/R83=c/
/R84=d/	/R85=e/	/R86=f/	/R87=g/
/R88=h/	/R89=i/	/R8A= .8A-/	/R8B= .8B-/
/R8C= .8C-/	/R8D= .8D-/	/R8E= .8E-/	/R8F= .8F-/
/R90= .90-/	/R91=j/	/R92=k/	/R93=l/
/R94=m/	/R95=n/	/R96=o/	/R97=p/
/R98=q/	/R99=r/	/R9A= .9A-/	/R9B= .9B-/
/R9C= .9C-/	/R9D= .9D-/	/R9E= .9E-/	/R9F= .9F-/
/RA0= .A0-/	/RA1= .A1-/	/RA2=s/	/RA3=t/
/RA4=u/	/RA5=v/	/RA6=w/	/RA7=x/
/RA8=y/	/RA9=z/	/RAA= .AA-/	/RAB= .AB-/
/RAC= .AC-/	/RAD= .AD-/	/RAE= .AE-/	/RAF= .AF-/
/RB0= .B0-/	/RB1= .B1-/	/RB2= .B2-/	/RB3= .B3-/
/RB4= .B4-/	/RB5= .B5-/	/RB6= .B6-/	/RB7= .B7-/
/RB8= .B8-/	/RB9= .B9-/	/RBA= .BA-/	/RBB= .BB-/
/RBC= .BC-/	/RBD= .BD-/	/RBE= .BE-/	/RBF= .BF-/
/RC0= .C0-/	/RC1=A/	/RC2=B/	/RC3=C/
/RC4=D/	/RC5=E/	/RC6=F/	/RC7=G/
/RC8=R/	/RC9=I/	/RCA= .CA-/	/RCB= .CB-/
/RCC= .CC-/	/RCD= .CD-/	/RCE= .CE-/	/RCF= .CF-/
/RD0= .D0-/	/RD1=J/	/RD2=K/	/RD3=L/
/RD4=M/	/RD5=N/	/RD6=O/	/RD7=P/
/RD8=Q/	/RD9=R/	/RDA= .DA-/	/RDB= .DB-/
/RDC= .DC-/	/RDD= .DD-/	/RDE= .DE-/	/RDF= .DF-/
/RE0= .E0-/	/RE1= .E1-/	/RE2=S/	/RE3=T/
/RE4=U/	/RE5=V/	/RE6=W/	/RE7=X/
/RE8=Y/	/RE9=Z/	/REA= .EA-/	/REB= .EB-/
/REC= .EC-/	/RED= .ED-/	/REE= .EE-/	/REF= .EF-/
/RF0=0/	/RF1=1/	/RF2=2/	/RF3=3/
/RF4=4/	/RF5=5/	/RF6=6/	/RF7=7/
/RF8=8/	/RF9=9/	/RFA= .FA-/	/RFB= .FB-/
/RFC= .FC-/	/RFD= .FD-/	/RFE= .FE-/	/RFF= .FF-/
/I			

Kerning Table

Below is a sample of the type of entries that would be added to a Translation Table if you wished to kern automatically as a job is being transmitted.

/=//==//

/HTA=60C2C246/
/HTa=60C2C206/
/HTo=60C2C230/
/HTe=60C2C202/
/HTy=60C2C22A/
/HTr=60C2C214/
/H--T=53C2C260/
/HT--=60C2C253/
/HT.=60C2C239/
/HT,=60C2C219/
/HVA=7CC2C246/
/HVa=7CC2C206/
/HVe=7CC2C202/
/HVo=7CC2C230/
/HV.=7CC2C239/
/HV,=7CC2C219/
/HWA=66C2C246/
/HWa=66C2C206/
/HWe=66C2C202/
/HWo=66C2C230/
/HWr=66C214/
/HW.=66C2C239/
/HW,=66C2C219/
/Hw.=26C2C239/
/Hw,=26C2C219/
/HAV=46C2C27C/
/HAT=46C2C260/
/HAW=46C2C266/
/HYo=6AC2C230/
/HYe=6AC2C202/
/HY.=6AC2C239/
/HY,=6AC2C219/
/HLY=64C2C26A/
/HLT=64C2C260/
/Hr.=14C2C239/
/Hr,=14C2C219/
/Hy.=2AC2C239/
/Hy,=2AC2C219/

/I

Translation Table For Decimal Tabs

The following table is a sample of the type of entries that would be needed to handle decimal tabs.

/=//==//

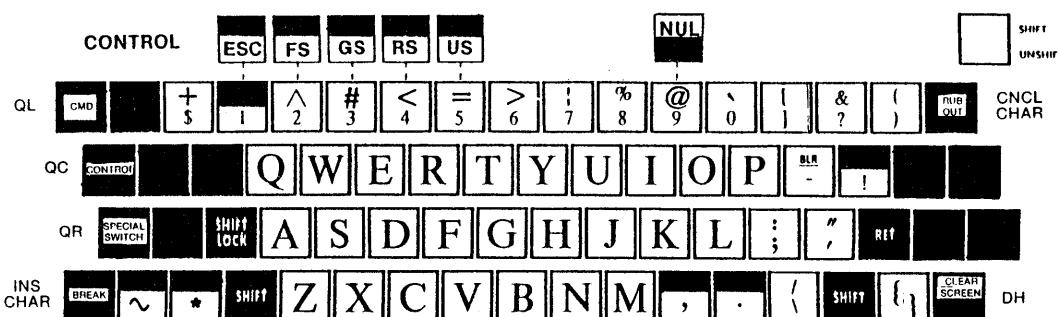
```
/H.0=8A27392D/  
/H.1=8A27393B/  
/H.2=8A273927/  
/H.3=8A273903/  
/H.4=8A273915/  
/H.5=8A273921/  
/H.6=8A27392B/  
/H.7=8A27390F/  
/H.8=8A27390D/  
/H.9=8A273931/  
/H          =8A3B/  
/H 0=8A3B2D/  
/H 1=8A3B3B/  
/H 2=8A3B27/  
/H 3=8A3B03/  
/H 4=8A3B15/  
/H 5=8A3B21/  
/H 6=8A3B2B/  
/H 7=8A3BOF/  
/H 8=8A3B0D/  
/H 9=8A3B31/  
/H  
=F8/  
/H;t=FA/  
/H;s=D6/  
/H;f=D2/  
/H;l=D4/  
/H;ll=D0/
```

/I;s12;l140;f1;l14500;t1F0400R;t2F0300L

EditWriter Keyboard Chart For Async Terminal Mode

The following keyboard layout outlines the position of the ASCII character and function codes when in Async Computer Terminal Mode.

FUNCTIONS OF THE EDITWRITER KEYBOARD IN TERMINAL MODE USING THE ICI



SPACE BAR			
[QL] com	Clears EditWriter screen and Activates ICI	[SHIFT DH]	Clear EditWriter screen when in terminal mode .
[QL]e	Exits communications and restores normal EditWriter operation	[CNCL CHAR] (rub out)	Sends ASCII rubout character.
[QL]c	Used in asynchronous terminal mode to initiate local copy.	[INS CHAR] (break)	Transmits break signal when in terminal mode .
[QL]s	Transmission status	[QC] (control)	ASCII control key used in conjunction with other keystrokes listed in the ASCII Control Code Chart on page 4 of the ICI Applications Manual
[QL]t	Stores the EditWriter screen data on the disk. If used when in receive to disk mode , this sequence will store the balance of the copy on the EditWriter disk and restore terminal mode .	[QC]q	X-ON starts transmission
[QR] (receive to disk)	In asynchronous terminal mode this key sends a return and places the ICI in the receive to disk mode .	[QC]s	X-OFF stops data transmission
		[QC]h	Back space
		[QL]n	Turns local copy off in terminal mode .

NOTE: Characters in brackets represent the actual EditWriter function key.

GLOSSARY OF TERMS (“Pardon me...You said what?”)

Acoustic Coupler—A device that converts machine codes into audible sound that can be carried over telephone lines to a similar device in a remote location. Couplers are always Asynchronous and generally operate at speeds of 0 to 300 baud. They can also use speeds of 1,200 and 2,400 baud. Acoustic couplers are just what they say they are...acoustic...they pick up sound. If operated in a noisy environment they may pick up surrounding sound and attempt to convert it to machine codes. This results in parity errors at the receiving end. If you have a choice, a modem is always preferable to a coupler.

See Also: Modem

ASCII—American Standard Code for Information Interchange. The code set used for Asynchronous communication.

See Also: Appendix “ASCII Code Set Chart”

Asynchronous—A mode of transmission in which a transmitted character does not necessarily have a timing relationship to any other transmitted character. Each character transmitted is framed with start and stop bits that tell the receiving device where the character begins and ends. The bits within each do have a specific timing relationship with each other.

See Also: Bisync, Protocol, Synchronous

Attended Operation—In Attended operation, the EditWriter will automatically exit the communications program at the end of a transmission (when an EOT is received) and return to normal EditWriter operation.

Baud—The transmission speed per second is often defined as a Baud Rate or Bit Rate. As part of the ICI Menu on the EditWriter you must fill in a Bit Rate (i.e. 300, 1,200, 2,400 etc.).

See also: Bit

Bisync—Binary Synchronous Communications. A set of synchronous, half duplex protocols developed by IBM. The ICI supports four different Bisync protocols: 2770, 2780, 3741, 3780.

See Also: Synchronous

Bit—Abbreviation of Binary Digit. Seven bits are necessary to make one character in the ASCII code set. Eight bits are required to create one character in the EBCDIC code set.

Bit Rate—The number of bits that can be transmitted in any given time frame. Bit rate is usually expressed in bits per second (bps).

Buffer—A specific area in the EditWriter that is used to temporarily store data, before it gets stored permanently on the disk.

Carrier—The high pitched tone that is used to transmit data from one modem to another. You may hear this tone when you dial a telephone that has a modem attached to it.

Clock—A source of specifically spaced timing pulses. Used to transmit data from and receive data to the EditWriter.

Data Communications—The transfer of information from one communicating device to another.

Data Communications Equipment (DCE)—This is typically the modem or any substitute for a modem, such as; null modems, line drivers, etc. One end of the 20 foot cable that is provided with the ICI is labeled DCE. This end is used to either plug directly into the modem (data communications equipment), or into the end of the null modem cable labeled ICI.

See Also: Null Modem

Data Phone—A telephone device designed for data communications. A data phone will normally have a data button or exclusion key.

See Also: Modem.

Data Set—Another name for a modem.

See Also: Modem.

Data Terminal Equipment (DTE)—This is typically the terminal in which the data is being transmitted or received. One end of the 20 foot cable that is provided with the ICI is labeled DTE. This end is used to plug into the EditWriter (data terminal equipment).

Data Transmission—The actual transmission of the data from one device to another.

DC1—Is an X-on command sent by the EditWriter.

See Also: X-on/X-off

DC3—Is an X-off command sent by the EditWriter.

See Also: X-on/X-off

Delimiter—Used in the Translation Tables. The delimiter can be any character. These delimiters must be determined in the keyword at the beginning of each Translation Table (i.e. / = // = //). We generally use a character that does not appear very often within text. In the define statement /AKat = Cat/, the slashes are delimiters.

See Also: Keyword, Separator

Duplex—See Full Duplex and/or Half Duplex.

EBCDIC—Extended Binary Coded Decimal Interchange Code. A code set developed by IBM in which each character consists of eight bits. This code set is widely used for Binary Synchronous Communications (Bisync). There are two very similar versions of EBCDIC call EBCDIC/WP and EBCDIC/DP

Echo—Applies to Computer Terminal Mode. Characters typed at the EditWriter (terminal) are not seen unless the receiving device is capable of echoing them. Local echo mode can be initiated at the EditWriter by striking [QL]c.

ENQ—“Enquiry” applies to Bisync. Simply put, the sending device enquires, “Is anybody there...??”. If a proper response is received, transmission will begin.

EOT—The “End of Transmission” code terminates the transfer of data from one system to another. In Synchronous communications an EOT is generated automatically. In Asynchronous an EOT may have to be generated manually by the operator of the sending system (code,D), or by the operator of the EditWriter, [QL]t.

FE—Framing Errors

Frame—A fixed amount of space and/or pulse that all the bits of a character must be in. A character is framed by start bits and stop bits. Used in Asynchronous only.

See Also: Start/Stop Bits

Log-on—A procedure that is used mainly with computers to insure that you have proper authorization to be accessing files on a computer. This mainly consists of secret passwords, identification numbers, etc.

Menu—Simply put... a list of choices. The ICI's "menu" allows the operator to select the various modes of operation. For example, once in the menu the operator can choose either Synchronous or Asynchronous communications which, in turn, selects either the EBCDIC or ASCII code sets.

Mnemonic (coding)—Mnemonic codes are multi-character sequences that represent Edit-Writer functions. There are EditWriter function codes like "size" and "font" that cannot be found on the sending system. But if the operator of the sending system uses mnemonic coding in their table and work, they can generate these commands. For example: ";" followed by 2 digits gives you a size change, ";" followed by one digit gives you a font change.

Modem—A device that turns machine codes (electrical signals) into audible sound that can be carried, long distances, over telephone lines.

See Also: Null Modem, 103J, 212A, 201C.

Modem Eliminator—Is used when cable connecting two devices that are set up to talk only to modems. It eliminates the modems. In the case of the ICI, the Null Modem cable is the modem eliminator.

Noise—Undesirable or unintended electrical signals (static) on the telephone lines that can distort and interfere with the data signals. In Asynchronous communications on the Edit-Writer, this may appear as question marks on the screen or Parity Errors (PE:).

Non-Transparent—Used in Bisync only. The EditWriter supports non-transparent as well as transparent communications, but always requires the EBCDIC code set.

Null Modem—A null modem (no modem) will be used in applications where the sending and receiving devices will be cable-connected. This cable makes the ICI look like a real modem to the sending system.

Off-line—Pertaining to equipment or devices that are not connected to a central processing unit or to a transmission line.

On-line—Pertaining to equipment or devices that are directly connected to a central processing unit or to a transmission line.

Page Image—See: Image Processing

Printer Control Codes—Control codes on the word processor or computer that give commands to the printer. Some of these codes are transmitted during a telecommunications session, and are of use to the EditWriter.

Printer Image—See: Image Processing

Parity Check—An error detecting method that applies to Asynchronous communication.

PE:—During an Asynchronous transmission, the EditWriter keeps track of the number of parity errors that occur during the session. When an error is encountered, the EditWriter will place a question mark on the screen in place of the questionable character. If, at the end of the transmission, you have a number in this field, use the SEARCH routine to locate the question marks and replace them with the proper character.

Point to Point—A connection made between two, and only those two, devices.

Protocol—This term refers to a set of parameters, or rules, used during a transmission. Both the sending and receiving devices must operate using like protocols (or play by the same set of rules) for a transmission to be successful.

RS-232C (EIA Standard)—A standard method adopted by the EIA to insure uniformity and compatibility of interfacing between data communicating equipment. This is a type of cable (connection) that has been generally accepted by most manufacturers of data communications and business equipment.

Separator—Used in Translation Tables. It can be any character, but we suggest you use a character that is not used frequently. In the equation /AKat = Cat/, the equals symbol is the separator, it separates one side of the equation from the other. This separator must be determined in the keyword at the beginning of each Translation Table (i.e., /=// = //).

Short Haul Modems—These are special modems used to connect only two devices together. These modems generally do not use phones, or the telephone company's lines. They generally use a different type of cable or dedicated telephone lines.

Software—Part of the equipment that you can't actually touch, like the programming.

Start/Stop—Used to refer to Asynchronous communications. Each character has a start and stop bit to outline the character, so the receiving end knows where each character begins and ends.

Stop bits—Used in Asynchronous. You can have either 1 or 2 stop bits.

See Also: Start/Stop

Synchronous—Sending all bits of a character and all characters of a transmission at a constant (synchronized) rate. The EditWriter's Synchronous Mode of operation can support (IBM compatible) 2770, 2780, 3780, 3741 protocols and also uses the EBCDIC code set.

Teletype (TTY)—Referring to a Teletypewriter.

See Also: TTY

Terminal—Any device that can send and/or receive data by a communications channel.

Terminal Mode—A mode of operation on the EditWriter that allows the keyboard to be active. Mainly used to log-on to computers and to request files from the computers.

Time-sharing—Used mainly with computers. Allows several different users to share the computer for different purposes at the same time.

Translation Table—This is what allows the ICI to be “user programmable”. It is a series of search and replace strings. Searching for the codes on one side of the “separator” and replacing them with the codes on the other side. Most of our text Translation Tables search for two returns in a row and replace them with a Quad Left, Return for the end of paragraphs. This table is input as a file on the sending system and is sent first to the EditWriter, followed by the text files.

Transparent Mode—Used in Bisync only. The EditWriter supports transparent communications but always requires the EBCDIC code set.

TTE—Translation Table Errors. A message automatically gets appended to the end of each transmitted file that tells you how many table errors were detected.

TTY—Teletypewriter equipment. Also used to describe the protocol that the EditWriter and other devices support in Asynchronous communications (simulating a teletypewriter).

Unattended Operation—An automatic mode of operation that allows the transmission and reception of data without operator intervention. In Unattended Operation, the EditWriter will remain in the ICI Mode until the operator exits communications manually by striking **[QL]e**.

X-on/X-off—Commands that are sent out from the receiving device. The EditWriter will send out an X-off, DC3, (wait) command when the ICI buffer is full. The sending system should stop sending information until it receives an X-on, DC1, command. This is to ensure that no data is lost.

103J—An Asynchronous, full-duplex modem that transmits 0 to 300 baud. This modem is compatible with another 103J or 212A modems.

212A—A dual purpose, full-duplex modem. The 212A can be set up to transmit Async at 0 to 300 baud and at 1,200 baud. It can also be set up to transmit Async at 0 to 300 baud and Synchronous at 1,200 baud. It is compatible with another 212A or 103J modem.

201C—A half duplex, synchronous modem with a fixed speed of 2,400 baud. The 201C is only compatible with another 201C.

Glossary of Word Processor Terms For EditWriter User's

Adjust = Rejust—A text editing feature in which the system automatically adjusts the right hand margin for insertion or deletion of copy during playback. Word and sometimes page wrap-around is automatically performed as needed.

Automatic Carrier Return = Automatic Line Ending—Automatic performance of a carrier return when the last word which will fit onto a line is typed. The system generally has a buffer to hold the word currently being typed until it judges whether to place the word on the current line, or to wrap it onto the next line. Systems which automatically perform carrier returns are speedier on input, since the operator may type text at a uniform speed without pausing at each line end to perform a return.

Automatic Centering = Quad Center/Neat Center—The automatic ability to center a word or text segment. This function is usually implemented by a keystroke(s) that instruct the system to center the previously typed or next text segment. Text may be centered between margins or a designated point. Also, some systems can center between tab settings for the centering column headings of a tabular matter.

Automatic Decimal Tab = Quad Right within Tab—Automatic alignment of columns of decimal figures on the decimal point. The typist can type numbers without regard for alignment, with the system performing the aligning chore.

Automatic Headers/Footers = Insert Block—The ability to place header/footer text at the top or bottom of each page of a multi-page document. The operator specifies the text once, and the header/footer (usually document title, company name or confidentially requirements) is automatically added during printout. Changes may be made to the main document text without affecting the headers and footers.

Automatic Input Underlining = Insert Rule—Ability of the system to cause text to be underscored without the operator having to strike the underscore key once for each underlined character.

Automatic Line Spacing = *Mixed Line Space*—Different line spacings (single, double, triple, etc.) are permitted without performing physical setting changes on the printer. This enables the typist to input text with combinations of spacing without stopping and resetting the printer during playback printout.

Automatic Margin Adjust = *Mixed Line Length*—Margins changes are performed by a single command, automatically changing line endings without further intervention, or the operator may effect changes with a multi-step process (by line, by paragraph, etc.) During margin adjust procedures, temporary hyphens are normally dropped (unless they occur at the end of the line), and the system may give the operator an opportunity to make new hyphenation decisions to afford a “tight” line.

Automatic Page Numbering—Sophisticated word processing systems (some video display standalones or shared-logic systems) are automatically able to generate page numbers within documents. When text is rearranged and page numbers change, the system can generate a new set of correct page numbers. This feature enables the operator to input text without regard to final page endings. The system will create pages of the desired length, and number them appropriately.

Automatic Repeat Key = *Insert Character*—A “live” typewriter key such as the underscore which will continue to operate as long as the key is depressed.

Automatic Typewriter—The simplest form of word processor. Used for straight, repetitive output with little or no text editing.

Automatic Typing—Producing final typed copy automatically from a word processing system with operator intervention necessary only to enter variable information. The term power typing sometimes applies to the same concept.

Automatic Word Wraparound = *Automatic Line Ending*—Automatically moves a word, which does not fit on the line being typed, to the next line. Frequently combined with the auto carrier return feature. Also used to denote systems which can wrap words during margin adjust procedures.

Automatic Widow Adjust—System prevents the first line of a paragraph, or a title heading from being the last line on a page. It may also prevent a last line from being the first line on a new page. Such a feature is especially desirable if the system paginates or repaginates automatically.

Backspace Correction = *Delete Character, Line & Paragraph*—The modes or increments provided by the system during the input process to allow the operator to delete a character (C), word (W), line (L), sentence (S), paragraph (P) or page.

Bi-directional Printout—With bidirectional or bustrophedon printing, the system prints line 1 from left-to-right and line 2 from right-to-left, saving time by avoiding unnecessary carriage (or element) movement. A few systems which employ bidirectional printing will also check for the closest margin before deciding to print a line left-to-right or right-to-left; this can save a few seconds in printing a segment immediately following a very short line.

Boilerplate = *Save Block/Insert Block*—Sections of standardized text that are stored and can be retrieved for inclusion in customized documents with little or no modifications.

Boldface—Boldface type is heavier in face than the text type with which it is used.

Bullets—Large dots used to draw attention to, or set off, paragraphs from the rest of the text.

Capacity = *Disk Capacity*—The amount of text which may be stored on one unit of mag media (cards, cassette, etc.), generally expressed in number of characters, or pages. In some cases, the media is formatted into lines, pages, documents, or files.

Carbon Ribbon—A Mylar ribbon backed with a carbon film for typewriters, producing a cleaner print impression than that usually achieved with fabric ribbon.

Carriage—The portion of a typing mechanism primarily concerned with controlling the paper; in character printers, the portion of the machine which usually moves horizontally.

Cathode Ray Tube (CRT) = *EditWriter Screen*—An electronic vacuum tube, such as a television picture tube, that can be used to display text and graphic images.

Character Density = *Characters Per Pica*—Number of characters per unit length.

Communications Hardware = *ICI Modem, Cable*—Communications hardware encompasses devices for transmission, switching, and termination of communications signals, such as data terminals, modems, acoustic couplers, PBX equipment, store-and-forward message switching computers, etc. Important to the understanding of networks are the interfaces between the host computers and the communications network, the interfaces between the users terminal devices and the communications network, and the message switching computer.

Continuous Form—Form manufactured from a continuous web which is not cut into units prior to execution. Materials may be carbon-interleaved, noncarbon interleaved or carbonless. It may be zig-zag folded (flat pack) or non-folded (roll).

Copy—A dual media word processing systems, the process of copying information from one media transport to the other.

Cursor—A light dot or blinking indicator to indicate position on a CRT.

Cursor Positioning—Describes the motion of a cursor (a lighted position indicator, indicated by a blinking underline or a reversed character). Most systems employ a series of arrow keys for up, down, left and right movement. Some systems use a Home key to position the cursor at the upper left corner of the screen. A few systems also offer a Reverse Home to move the cursor to the lower right corner of the screen. Some systems use a code key plus alpha-numeric or function keys for cursor movement. A number of systems only permit cursor movement horizontally, along a fixed line.

Daisy Wheel—A print element for certain character printers where the characters are engraved at the end of spokes, the entire print “wheel” resembling a daisy.

Data Entry—Equipment and procedures designed to capture data in machine readable form for computer processing.

Data Processing—An operational sequence, usually mathematical, performed on facts and figures. The data processing system operates upon input data, processes files and produces output under the control of a stored program. The numerical methods of processing data differentiate it from the syntactical processing of words in word processing systems.

Deletion—In text editing, the ability of the operator to remove any select portion of copy without substituting new.

Density—Used generally to refer to the blackness or darkness of a typed, printed, or carbon image. Density control is the operator control which affects the hammer or type impact; on a typewriter, often called “impression control.”

Disc Storage—A high capacity access storage device in which data is written on and read from the surfaces of a stack of revolving record-like discs which have been coated with magnetic material.

Distributed Logic—A system configuration where numerous word processing stations share computer power, storage and peripherals, with each individual station having some intelligence.

Dual Column = Multiple Column Position Option—This refers to formatting capabilities in a word processing system to permit data recorded in a single column format to be printed in two columns on a page.

Editing—Changes or re-arrangements in the text, including reading back, scanning, deleting, substituting, inserting and re-formatting.

Element—Refers to the removable element in some character printers containing the type set. Most commonly, includes the “golfball” element for the IBM Selectric and the “daisy wheel” element for the Diablo and Qume printers.

Elite—Typewriter spacing at 12 characters per inch. Also known as 12-pitch.

Entry = Enter File—Usually used to describe original typing and entry of text into a system.

Execution = Execute—Of an instruction, the set of elementary steps (or primitives) carried out by the computer to produce the result specified by the operation code of the instruction.

Fabric Ribbon—Usually a nylon typewriter ribbon.

Fast Forward—A tape recorder feature which permits the tape to be rapidly run in normal play direction for search purposes.

Font—A character set of a particular type size and style.

Form Letter—A letter sent to a variety of recipients having the same or much the same body; minor variations within the body may be provided. Form letters may be preprinted; however, word processing applications for form letters allow each letter to be individually typed, providing a more professional appearance.

Format—1.) The type style, size, page margins, printing requirements, etc. of a document, form or any printed piece. 2.) In data processing, organization of data within a computer, especially in preparation for printing it out.

Format Display—The ability of a word processing system to display on a CRT or print on a printer the control codes for recorded format.

Global Search and Replace—The ability of a system to search for repeated occurrences of a character string (typically up to 32, 64 or 128 characters long). In some instances, the system can automatically delete all occurrences of a string or replace all occurrences of one character string with another character string. In other cases (EditWriter), the system merely locates the string for operator-selected deletion or replacement. A few high powered

systems can apply logical considerations to making the replacement or perform multiple searches simultaneously.

Hard Copy—Typewritten or printed copy of any description (as opposed to “soft copy,” which may be stored on a medium or displayed on a video display, but does not exist on paper).

Hot Zone—Hyphenation technique where the operator specifies a fixed or variable length zone next to the right hand margin. Any word that begins prior to the hot zone and which will not end before encountering the margin causes typing to halt as soon as the hot zone is entered.

Indent—Refers to sections of text where a temporary left margin is set up by a given amount for each line. Margin indent is handled automatically by most word processing systems.

Indicator—A light, usually on the operator’s console, that is turned on to indicate a particular condition occurring in the system.

Information Processing—A term used to encompass more than data processing or word processing. Inlcudes all business and scientific operations performed by a computer.

Input—Material entered into a word or data processing system for processing.

Insert—Add characters, words, sentences, or paragraphs into copy.

Intelligence—Refers to the level of complexity of automatic functions provided by a system. “High intelligence” infers a sophisticated control logic (usually an integral mini or microprocessor) capable of providing many automatic word processing functions.

Justification—“Full justification” allows the operator to output text with even left- and right-hand margins, with equal line length calculated automatically by the system.

Keyboarding—Inputting information by manipulating a keyboard. Somewhat synonymous with “typing,” except “keyboarding” does not have the same connotation or producing printed output.

Logging = Enter File—A method of recording, cataloging and/or filing documents or media regarding a word processing installation.

Logic—1.) The science dealing with the criteria or formal principals of reasoning and thought. 2.) The systematic scheme which defines the interactions of signals in the design of a system. 3.) The basic principals and application of truth tables and interconnection

between logical elements required for arithmetic computation in an automatic data processing system.

Menu Prompt—A number of video display systems allow the operator to enter into a dialog or conversation with the word processor; operator actions are called for by the system in a question-answer mode. This type of dialog can be particularly useful in training new operators, or in helping part-time operators through a job. Some video systems offer Menus which list operator options at each step in the word processing process.

Magnetic Media = Disk Storage—Refers to magnetic cards, tape or disc storage media (as contrasted to, for example, paper tape).

Memory—An integral component of many word processing systems where information is temporarily stored. The memory can act as a buffer for reading or writing to I/O devices; may act as temporary storage for text entered from the keyboard; may act as temporary storage for text being edited or formatted; or may hold the word processing program.

Off-Line Printing—Ability of a system to print one page of a document simultaneously while the operator is entering or editing another page of the same document or another document.

Optical Character Reader (OCR)—A device or scanner which can read printed or typed characters and convert them into a digital signal for input into a data or word processor. OCR units in word processing applications usually read special machine-readable type fonts. The use of such equipment allows an ordinary typewriter fitted with a font to serve as an input station for a word processing system. Pages produced on the typewriters are fed into the OCR and converted into a digital form. Such digitized text may either be entered directly for text edit and format, or stored on mag media for future processing.

Output—The process of transferring information from internal storage to an external source, to the printing device or storage medium. Also refers to that information itself.

Peripherals—Devices (such as printers, OCR readers and communications) which may be configured with word processing systems as options, extending their capabilities. More sophisticated systems can frequently share a peripheral between multiple stations, making the use of a high speed printer or other expensive pieces of equipment cost-effective.

Pica—Typewriter spacing at 10 characters per inch horizontally. Also known as 10-pitch.

Platen—Rubber-covered cylinder around which a form or paper sheet is passed for writing in most character printers. Provides support to oppose the impact of the type, and, in conjunction with pressure rollers, drives and controls form advance (PRESSURE FEED) or by pins engaging marginal punching (PIN FEED).

Prerecorded—Text stored on magnetic media for subsequent playout. (May be a repetitive letter or a letter created from Boilerplate. Variable information, either prerecorded or keyboarded, may be combined with such prerecorded text.)

Printout Queueing—Allows a number of documents to be lined up or queued for subsequent printout while the operator goes on to perform other tasks. Such printout queueing may be quite primitive, employing only a single printer and handling one page or one document at a time. Other queueing may be very sophisticated, with multiple printers and print queues for each printer, capable of processing large documents or allowing multiple documents in the queue. Some systems also allow documents to be deleted from the queue and/or priority documents to be processed ahead of a normal first-in, first-out queue.

Program—A sequence of instructions executed by a digital computer designed to accomplish some specific processing task.

Proportional Spacing—Typed, printed or displayed text where each alphanumeric character is given a weighted amount of space. For instance, an "I" might be two units wide, an "L" four units wide, and a "W" five units wide. Such output has a print-like appearance, especially when combined with a character spacing scheme employing sophisticated intercharacter spacing.

Read—The process of transferring information from some external storage medium into a system.

Response Time—The time a system requires to respond to an operator command in supplying stored data or completing a processing cycle.

Reverse Index/Line Feed—Half-line vertical spacing for subscripts and/or superscripts.

Scroll—A capability in most video display systems to allow the operator to view a larger body of text than can be displayed at one time. Controls are provided to move the text past the "window" both horizontally and vertically. During entry, text will "scroll" off the top of the display as new lines are entered at the bottom of the display.

Software—The machine programs which instruct and control computer operations. Also used in word processing to mean all non-hardware parts of the system, including manuals, training, etc.

Stop Code—A reference code recorded on magnetic media which causes the system to stop during printout. Used to allow the operator to perform such manual procedures as changing fonts or paper on the printer.

Storage Capacity—Total amount of text stored per unit of media (card, cassette, diskette) which may be accessed by the system without changing media. A magnetic card system has about 5,000 to 10,000 characters “on-line.” A magnetic tape cassette would hold up to 300,000 characters. Diskettes hold about 250,000 to 300,000 characters, but in many cases the word processor’s operating system software is also stored on the diskette, so only 60 to 100 pages of storage are available. Discs can hold much larger quantities of data, frequently 2 to 50 million characters or more on each disc.

Sub/Superscript Printout—The ability of the printer to print characters at a fractional increment (sometimes adjustable) above and below the line for footnotes, formulas, etc.

Switch Code—A code which permits switching between media stations of a word processing system, allowing the system to combine such separately stored text as an address list with a repetitive letter.

Text Editing—A general catch-all phrase to cover a wide variety of word processing systems and procedures involving text editing.

Track—The path on magnetic tape or card along which a single channel of sound or codes is recorded.

Turnaround Time—Elapsed time between dispatch and receipt of material back at starting point.

Word Processing—A system of trained personnel, specific procedures and automated equipment that provides more efficient and economical business communications; usually involves the transformation of information into readable form.

Word Processing/Office Systems (WP/OS)—A term referring to the total information handling system of an organization, including word processing, administrative support, micrographics, photographics, data processing, communications, etc.

Index

A

- Advantages of the ICI 7
- Aligning Numbers in Tabs 82, 95–97, 116
- Alpha to Alpha Translations (A) 68
- Alpha to Hex Translations (H) 69
- ASCII 17, 35, 64
- Assigning File Names 20, 22, 25, 27, 44
- Asynchronous Communications 35
- Asynchronous Mode 6, 16, 21, 35
- Attended Operation 20, 22, 25, 43

B

- Backspace 27, 76, 108
- Baud Rates 20, 22, 24, 36, 42
- Batch Receive Mode (see: *ICI Mode*)
- Bisync Protocols (see also: *Synchronous Mode*) 17, 41
- Bit Rates (see: *Baud Rates*)
- Block Size 41
- Break Signal 26

C

Cabling	45
Cancel Character	27
Caps To Lower Case	89
Carrier Detect (CD/LSD).....	46-48
Centered Lines	86
Character to Character Translations	64
Character/Line Limit	27, 44
Clear To Send (CS).....	46-48
Clearing the Screen	26, 40
Code Charts	106, 107, 109
ASCII.....	107, 108
EBCDIC	109, 110
EditWriter	105, 106
Code Identification Tables (<i>see: Debug Tables</i>)	
Coding at the Word Processor (<i>see also: Mnemonic Coding</i>)	98-101
with Formats	95-97
with Numbers	100, 101
with Words	70, 79, 99,
Command/Control Keys	26, 40
Command Key Sequences	40, 117
Common Table Errors	102
Communications Menus	19, 21, 23, 29-32
Comparing Translation Types	73
Computer Terminal Mode	6, 16, 23, 38, 117
Conversational Mode (<i>see: Computer Terminal Mode</i>)	
Conversion Tables (<i>see: Translation Tables</i>)	
Correcting Table Errors	102

D

Data Communications Equipment (DCE)	48
Data Phone 2400.....	51
Data Phone 300/1200	50
Data Set Ready (DSR).....	46-48
Data Terminal Equipment (DTE)	48

Data Terminal Ready (DTR).....	46-48
Debug Tables	111, 113
Decimal Tabs	95-97, 116
Default Tables	9, 17, 64, 108, 110
Define Statements.....	68-72
Deleting Codes	76
Delimiter	67
Designator.....	67
Diagnostics	72, 55-58, 111, 113
Disk Capacity	16, 44
Dot Leaders	87
Double Spacing	91

E

EBCDIC/DP	43, 80
EBCDIC/WP	43, 80
Echo	26, 38
EIA Standards (see also: <i>RS-232C</i>).....	45, 51
End of Table	73
End of Transmission (EOT).....	43
Entering File Names (see: <i>Assigning File Names</i>)	
Entering Communications	19, 21, 24
Equations	68-72
Errors (see: <i>Troubleshooting</i>)	
Errors, Communications	58
Errors, Tables	102-104
Even Parity (see: <i>Parity</i>)	
Exiting Communications	20, 22, 26, 39
Extended ASCII	35, 37
External Timing	48, 53

F

File Capacity	16, 44
File Naming Conventions (see also: <i>Assigning File Names</i>)	44
Finding Table Errors	103

Format Lines	88, 90, 91
Formats	95, 96
Framing Errors (FE).....	28

G

General Purpose Translation Tables (see: <i>Translation Tables</i>)	
Generating File Names (see: <i>Assigning File Names</i>)	

H

Hardwire.....	45
Headers	90
Hex Codes	81, 105-110
Hex to Alpha Translations	72
Hex to Hex Translations	70
Hexidecimal Numbering System	81
Horizontal Tabs	82
Hyphens	86

I

IBM OS/6	29
ICI Mode	6, 16, 19, 38
ICI Self Test	17, 56-58
Indent Tabs	82
Initialization of Typesetting Parameters	74
Inputting Copy	93
Inputting Translation Tables	101-104
Interactive Mode (see: <i>Terminal Mode</i>)	
Internal Modems	45

K

Kerning	115
Keyboard Charts	105, 117
Keyword	27, 66

L

Leaders	87
Line Drivers	51
Line Endings (see: <i>Returns</i>)	
Line Feeds	91
Line/Character Limit	27
Loading the Table	63, 101
Local Echo (see: <i>Echo</i>)	
Locating Table Errors (see: <i>Finding Table Errors</i>)	
Logging On	23

M

Margins	63, 91
Menus	19
Modem Elimination (see: <i>Null Modem Cable</i>)	
Modems	48/51
Modes of Operation	16
Mnemonic Coding	78, 98, 100
Multijust Commands	91

N

Null Modem Cable	51-53
------------------------	-------

O

Odd Parity (see: <i>Parity</i>)	
Opening Files (see: <i>Assigning File Names</i>)	

P

Pacing the Transmission (see: <i>X-on/X-off</i>)	
Page Headers	90
Paragraph Endings (see: <i>Returns</i>)	
Paragraph Indents	87
Parameters (see: <i>Initialization of Typesetting Parameters</i>)	
Parity	22, 24, 37
Planning Your Work	8
Precedence of Translations	75
Preparing Copy	93
Preparing to Communicate	16
Printer Control Codes	88
Processing Copy	33
Prompts (see: <i>Screen Prompts</i>)	
Protective Ground (PG)	46
Protocols	16

Q

Quotes	87
Queueing the Photo Unit (see: <i>Processing Copy</i>)	

R

Receive Clock (RXC)	46-48
Receive Data (RXD)	46-48
Recommended Techniques for Customizing Tables	81
Record Length	41
Request to Send (RTS)	46-48
Return Codes	65, 78, 83, 92
RS-232C	5, 45-48
Rubout	26
RVI	41

S

Screen Prompts	19-25
Search and Replace	67
Selecting Bit Rates	20, 22, 24, 36, 42
Selecting Protocol	17, 19, 22, 24
Self Test Procedures	56-58
Separator	67
Sequence of Translations	75
Signal Ground (SG)	46-48
Space	84
Space Compression	77
Start/Stop	35
Stop Bits	22, 24, 37
String Translations	68-72
Synchronous Communications	41
Synchronous Mode	16, 19, 41

T

Table Errors	27, 102-104
Table Format	66
Tabs	72, 82, 85, 95-97, 101
Tabular Coding with Numbers	82, 97
Techniques for Customizing Tables	81
Terminal Mode	6, 23, 38
Terminal Port Connections	54
Testing	97
Timing	53
Translation Sequence	75
Translation Tables	9, 18, 61-104
Translation Table Errors	27, 102-104
Translation Types (see: <i>Designator</i>)	
Transmit Clock	46-48
Transmit Data (TXD)	46-48
Transmitting the Table and Text	101
Troubleshooting Guide	55-60
TTE (see: <i>Translation Table Errors</i>)	
Typesetting (see: <i>Processing Copy</i>)	

U

Unattended Operation	20, 22, 25, 43
Underscored Text	76
Upper Case to Lower Case	89

V

Voltage Requirements	46
Vydec 1400 Communications Menu	30

W

WACK.....	41
Wang Communications Menu	31
Word Processors	18, 28
Work Flow	8-15

X

Xerox 850 Communications Menu	32
X-on/X-off	35, 40, 54