IDENTIFICATION

IMEDIT -- Editor Program for Use with the Imlac Terminals
Status as of August 1972

Jack Haverty

16 August 1972

DRAFT

MOTIVATION

The IMLAC terminals are small computers with a reasonably sophisticated monitor/editor program which runs in them while connected as ITS terminals. In most cases, the monitor program in the IMLAC sends what is typed directly into the program in the PDP-1Ø, and displays on the screen whatever ITS sends back. The IMLAC looks like a normal teletype with a CRT display, since none of the editing functions of the IMLAC are used when it is used simply as a console.

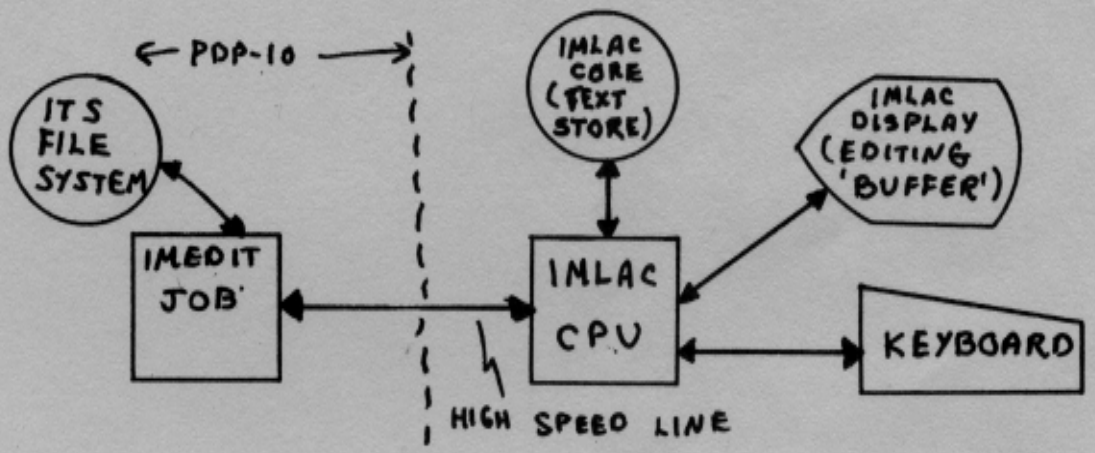
IMEDIT is intended as a means of allowing the use of the editor program in the IMLAC for simple editing of user's files. It is not intended to replace TECO; however, for many editing tasks it is easier to use than TECO. The basic advantage of IMEDIT however, is the fact that it occupies considerably less core than TECO and is simpler to learn for a novice. The buffer for editing is kept in the IMLAC core at all times; IMEDIT acts as a controller which interfaces the editor in the IMLAC core with the file system in ITS.

REFERENCES

1. IMLAC Corporation, User's Reference Manual, IMLAC PDS-1 Programmable Display System, 197Ø, SYS.52.Ø1.

I. IMLAC - PDP-1Ø RELATIONSHIP

We can visualize the data flow and location when using IMEDIT as pictured here:



IMEDIT, running in the PDP-1Ø, has two functions. It acts as an overseer of data transfers (i.e., pieces of files) between auxiliary storage and the IMLAC; it also permits the user to query ITS about various things such as directories, etc. These functions will be made clear in the description of each command.

When you are using IMEDIT, there are no buffers for your text stored in the PDP-1Ø core. To edit a file, you give IMEDIT a command telling it where to get the input file, and an optional command telling it where to send your output. The default for output is your disk area. To actually edit the file, you issue commands which cause IMEDIT to read a part of your input file, and send it directly to the IMLAC. The IMLAC stores the chunk of

file in its core, and uses that data to maintain the display. By positioning the cursor and typing, or hitting the "DEL" key, the text stored in the IMLAC core is changed, and these changes are also made on the screen. Whatever is on the screen at any time is exactly what is stored in the IMLAC core. When you are finished modifying the portion of the file appearing on the screen hitting "PAGE XMIT" causes the IMLAC to send the entire text which is on the screen back to ITS.

IMEDIT listens to the IMLAC whenever it is not sending, and considers anything it sees, which is not identifiable as a command, as part of your output file. When you hit "PAGE XMIT", it reads the stream of ASCII characters which the IMLAC is sending, and sends them directly to the output file.

The IMLAC editor program (i.e., the monitor/editor in the IMLAC) is used as a console in non-local mode. In this mode, everything typed at the IMLAC is immediately sent to the PDP-1Ø, as well as placed in the IMLAC core (so it can be displayed). In using IMEDIT, we do not want everything which is typed to go in immediately to the PDP-1Ø, but instead have it put in the IMLAC core so it can be edited, if necessary, to correct mistakes. Because of this, it is necessary to shift the IMLAC into a mode called local mode; in this mode, only control characters are sent immediately to the PDP-1Ø. All other characters are placed in the IMLAC core, and displayed on the screen. These are two ways to send these characters to ITS. To send a single line, the "XMIT" key is used. This key can be considered as performing the same function in local mode as a carriage return does in the

normal mode of operation. When you hit the "XMIT" key, the IMLAC sends to ITS the characters on the screen since the last carriage return, or last character received from ITS. To send the entire page to ITS, hit "PAGE XMIT". This causes the IMLAC to transmit exactly what you see on the screen to the PDP-1Ø, except for the various comments displayed by the IMLAC monitor, e.g., --LCL--. Depending on the number of characters on the screen, this process takes from 1/2 to 2Ø seconds or so. While the data transfer is occurring, do not type anything. The string --WAIT-- will be displayed near the top of the screen. Occasionally, if the core situation in ITS is tight, the IMLAC will pause, waiting for IMEDIT to tell it to continue.

Note that in using IMEDIT, it is not necessary to have an input file if you simply want to input from the console. In this case, just type at the editor, compose the text on the screen exactly as you want it to look in the file, and "PAGE XMIT" it to IMEDIT, which will put it in the output file. This can also be done at any point in a file which you are editing.

Any command which takes an argument may be aborted if a G is typed instead of the argument. IMEDIT will point "Command aborted" and return to its input wait loop, in the same state as before you typed the erroneous command. This is very useful for the cases where you accidently hit the wrong key, which usually happens at the worst possible times.

II. USING IMEDIT - BASIC COMMANDS

IMEDIT is conceptually two distinct parts - one section controls the transmission of parts of a file to the IMLAC; the other governs the routing of the stream of characters coming in from the IMLAC. If you consider these two functions as separate, it is easier to understand the use of IMEDIT.

All commands in IMEDIT are control characters. Because of the characteristics of local mode in the IMLAC, any control character typed at the IMLAC is immediately sent to IMEDIT, running in the PDP-1Ø. When the command is typed, it will echo as an up-arrow followed by the corresponding letter, e.g., CTRL-A echoes as "↑A." A few commands take no arguments; these are immediately executed, and need not be followed by anything. Other commands take either a numeric or file-name argument. These commands should be followed by the argument, and terminated by "XMIT". E.g., to close and name your output file:

↑EFOO BAR(XMIT) .

Note that the argument may be edited if mistakes occur, before you type 'XMIT'. Because these commands may take arguments, if you use them without arguments they must be followed by a 'XMIT' to complete the command.

A. Commands Associated With Input Files

There are six basic commands used in manipulating input files. These perform three classes of functions: 1) specifying where the input file will be found, 2) manipulating pages of the input file without sending it to the screen, and 3) sending portions of the input file to the IMLAC for editing.

The R command is used to tell IMEDIT where to find an input file. This command may be issued at any time; if a previous input file is open, it will be closed. Note that this command has absolutely no effect on anything you have sent to your output file. When the ↑R command is given, IMEDIT will ask for an input file. The conventions for typing input files are identical to those of MONIT (same code!) i.e., "DEV: USR; NAME1 NAME2" where DEV: and USR; are optional. IMEDIT remembers the last device and names it has seen in any command, and if any of these are omitted in the file specification, the previous specification will be used. In particular, giving no information at all (i.e., just a 'XMIT') will cause the last file name seen to be opened as input; e.g., if you want to read a file you just finished outputting.

If you specify a "USR;", as, for instance, you would to read from another directory all successive references to DSK will be to that directory. To fix this problem, specify your own "USR;" (i.e., login name) to the next reference to the DSK.

In general, the ↑R command is analogous to TECO's ER command.

As in TECO, the input file to IMEDIT must be thought of as a linear stream of words. As you read the words, they are gone from the input; you cannot "back up" and read the same part of a file twice without reopening it as an input file. When a file is opened (with ↑R), IMEDIT's file reader is conceptually positioned at the beginning of the file. The other five commands in this section control what IMEDIT does with each chunk it

reads.

ASCII files in ITS, as operated on by TECO and IMEDIT, are logically divided into pages which are separated by form-feed characters. In TECO, the basic unit of a file on which you operate is a page, which is kept in core. In IMEDIT, no part of the text is kept in the PDP-1Ø core. The concept of pages does not directly apply in this case. However, it is useful to consider the "chunk" of text which are sent to the screen as "pages". To avoid confusion, these chunks of text will be termed "screenfuls" to distinguish them from pages defined as in TECO.

The buffer which is analogous to TECO's buffer is the IMLAC screen. The cursor on the IMLAC screen is a visible pointer. Using the keys on the IMLAC, this pointer is positioned in the buffer, and additions or deletions made, just as you do in TECO, except in the IMLAC editor you can see exactly what you're doing.

The next two commands of interest operate on pages of text, as opposed to screenfuls. The reason for keeping the page concept is because generally pages separate a file into logical parts, and in addition make a large file manageable.

The ↑P command causes IMEDIT to read text from its current position in the input file until it finds a form feed, and also send that text, unchanged, to the output file, including the form feed. It takes a numerical argument which is the number of form feeds to skip over; if no argument is given, a 1 is assumed. This enables you to skip over pages of your file which you don't want to change without having the page sent to the IMLAC and then transmitted back to IMEDIT. If, for example, you want to edit

the third page of your file from the page you are currently editing, you would type ↑P3 (XMIT) to do this. Note that if you currently have a page on the screen, which you want to precede the pages being passed over, you must send it to the output file by "PAGE XMIT" before giving the ↑P command.

The ↑X command acts exactly like ↑P, except it does not send the pages it reads to the output file. These pages are read and discarded. If, for example, you want to extract the 3rd page of a program, you would give a ↑X2(XMIT) command after the ↑R, to position IMEDIT's reader at the beginning of the 3rd page, so that the next input taken will come from page three.

The final three commands are used to control what is sent from the input file to the IMLAC buffer to be edited. These commands operate on single lines of a file, or "screenfuls" as defined above.

The ↑Y command caused IMEDIT to send a screenful of text from the input file to the IMLAC buffer (i.e., on the screen). The position in the IMLAC buffer where this screenful will go is at the current position of the cursor on the screen. Normally, this command will be used by clearing the screen, which puts the cursor at the upper left, then typing ↑Y to put a screenful on display, ready to be edited. However, you may also insert the screenful anywhere into the text already on the screen by positioning the cursor at that point before typing ↑Y. The ↑Y does not echo - if it did, you would have to delete the 2 characters "↑Y" from the screen, or they would be in your file when you sent the screen contents to IMEDIT.

Note that if you have just finished editing a screenful of text, and want to get the next screenful onto the screen, you must type "PAGE XMIT" to send the current screen contents back to the output file. If you first type the ↑Y, the screen contents are not automatically sent to the output. For this reason, the ↑Y command is analogous to TECO's "Y" command. "PAGE XMIT" can be thought of as similar to TECO's "PW" command.

The ↑L command operates identically to ↑Y except it sends only one line of the input file to the IMLAC buffer.

The ↑W command takes a one or two digit octal argument. Its function is to set the number of lines sent to the IMLAC by each ↑Y to the value of the argument. e.g., ↑W5 (XMIT) will cause IMEDIT to send 5 lines of text for each ↑Y received, until another ↑W command is given. Similarly, ↑W1 (XMIT) will make a ↑Y act exactly like a ↑L.

Other commands which are useful but not necessary are described later. The one most important concept in using IMEDIT is that what you see on the screen is what is in the IMLAC buffer, and it must be sent back to IMEDIT in order to have it appear in your output file, by hitting "PAGE XMIT".

B. Input File Pushing and Popping

Two commands in IMEDIT are used to provide an input pushing facility. At any time, you may "push" your current input file, and open a new one by typing ↑G, which will request names of the new input file. This is identical to a ↑R, except that the previous input file is still accessible, by typing a ↑T to return up a level. After typing a ↑G, all input will come from the new input file, until a ↑T command is executed; this may be done at up to 8 levels. For example, to insert the second page of "FILE 2" between the fourth and fifth pages of "FILE 1" the following sequence could be used:

```
*↑R  INPUT FILE: FILE 1 (XMIT)
*↑P  NUMBER OF PAGES TO OUTPUT FILE: 4 (XMIT)
*↑G  Input FILE PUSHED.
     CURRENT LEVEL IS 2

     FILE NAME: FILE 2 (XMIT)
*↑X  NUMBER OF PAGES TO DISCARD: 1 (XMIT)
*↑P  NUMBER OF PAGES TO OUTPUT FILE: 1 (XMIT)


*↑T  INPUT FILE POPPED
     CURRENT LEVEL IS 1
* ↑E  COPYING REST OF INPUT FILE
     OUTPUT FILE NAME: MERGED FILE (XMIT)
```

C. Commands Associated with the Output File

There are three basic commands used to manipulate your output file. The ↑O command is used to specify a location for the file (i.e., device); the ↑E and ↑F commands are used to close your output file.

Any character which IMEDIT receives from the IMLAC, which it does not recognize as part of a command, is automatically routed to the output file. If no output file is currently open, one will be opened on DSK. You may close an output file at any time, but a new one may not be opened until the old one is closed.

The ↑O command is followed by an argument of the standard file specification form (as described under the ↑R command). The device ("DEV:") must be given to the ↑O command, while the file names are ignored, but become the "remembered" names. When this command is given, IMEDIT opens its output file on the specified device; all output through the next ↑E or ↑F command will go to that device.

The ↑O command cannot be given after any output has been done. If no ↑O command has been issued when IMEDIT has text to output, a file is opened on DSK to receive your output. This file must be closed (by a ↑E or ↑F) before you can change the location of the output file. Note that it is necessary to give this command only when you don't want your output on DSK.

The ↑E and ↑F commands are analogous to TECO's EE and EF commands. Each is followed by an optional file specification to tell IMEDIT what name to give your output file. If no name is given, the current "remembered" name is used, replacing the old

file with the new one. ↑F will close the output file immediately, and give it the specified name. ↑E will read from the input file, and copy everything it reads to the output file before closing and naming the output file.

Note that it is not possible to give a new "USR;" to either of these commands. The output file will not be in the new directory. If you want your output in a different directory from your input, use the ↑O command to specify the "USR;" name.

Note also that it is possible to give a new DEV: to either of these commands, if, for example, you want your output on tape, and forgot to give a ↑O command. IMEDIT will close the output file on the disk, copy it over to the new device, and then delete it from the disk. Do not try to give a new device to these commands if you did use the ↑O command, as unpredictable lossage may occur.

The ↑F command causes the output file to be closed immediately and renamed, as TECO's EF command does. The ↑E command causes IMEDIT to copy the rest of the input file over to the output file and then close it. This has the same effect as the user giving enough ↑P commands to exhaust his input file, and then giving a ↑F. Essentially, these commands function exactly as the corresponding TECO commands, with the added flexibility of being able to specify a new output device.

D. Miscellaneous Commands

Other commands in IMEDIT which are useful are summarized here:

↑C - takes no argument, causes IMEDIT to send a description of current commands to the screen for reference.

↑D - followed by standard file specification, causes that file to be deleted. If no device is given, the last device given to a ↑U command is assumed.

↑K - followed by an ASCII string and (XMIT), exists from IMEDIT and passes string to IMEDIT's superior, e.g., HDDT(XMIT) will cause DDT to be loaded. This command is the "standard approved method" of getting out of IMEDIT.

↑B - followed by a single digit and (XMIT), causes corresponding microtape to be flapped; i.e., TECO ↑2EK is the same as ↑K2 (XMIT).

↑U - followed by "DEV: (XMIT)", causes directory of that device to be sent to the IMLAC screen.

When using the ↑U command, IMEDIT will send a screenful of the directory at a time, and wait for a response without first typing an "*". Typing any control character except ↑M or ↑G will cause the next screenful of the directory to appear on the screen. Control-M will return you immediately to the IMEDIT listener and type the "*" indicating it is ready for another command. There are a few other commands currently implemented, but these are not necessary for most editing. All current IMEDIT commands are summarized on the last page for convenience.

E. Search Commands

There are four search commands: ↑S, ↑N, ↑V, ↑Q. Each command is followed by an argument giving the strings to look for. The behavior of the four commands is as follows:

↑S -- search until a match is found or a form feed is encountered. Send to the output file everything from the input file, up to but not including the entire line in which the match is found. If no match is found, print an appropriate message on the console. If a match is found, print the entire line on the console and position the input reader at the beginning of the next line.

↑N -- same as ↑S, except does not stop searching until end of file on input, or a match occurs.

↑V -- same as ↑S, except does not send anything to the output file.

↑Q -- same as ↑N, but does not send anything to output file.

Arguments to Search Commands

All search commands require an argument. Nothing happens until you hit "XMIT". Arguments are of the form:

%string1%string2%...stringn%(XMIT)

The % represents a delimiter, which may be any non-control character which does not appear in any of the strings. E.g., use . | % $ @ etc.

Imedit will search for the first occurrence of any of the strings specified in the argument.

The argument strings are saved in a search list, which is remembered after the search. A search command given with no arguments (i.e., just a "XMIT") will search for the list of strings last seen in an argument.

E.g.,

↑S /FOO/(XMIT) searches for FOO

and

↑S(XMIT) will then also search for FOO

A special feature is provided to allow you to add strings to a search list, without deleting strings already there. To do this, make the first string in the argument the null string: I.e.,

↑S /FOO/ searches for FOO

↑S//BAR/ then searches for FOO or BAR

III. USING THE EDITOR FUNCTIONS OF THE IMLAC

In the following discussion, the IMLAC is assumed to be in local mode; IMEDIT, when it starts, automatically places the IMLAC in this mode.

Any new text which the IMLAC receives, either from its keyboard or from ITS will be inserted in the screen buffer of the IMLAC at the current position of the cursor. To add text to a page of your file which is currently on the screen, or to a completely blank screen, you merely position the cursor at the point where you want the characters inserted, and type them in. The cursor will move as you type, to position itself after the letter just typed, so that long strings of text are inserted by just positioning the cursor at the point where the string is to begin, and typing it in.

Similarly, hitting the 'DEL' key will cause the IMLAC to delete the letter immediately to the right of the cursor. This also causes the entire contents of the screen to be moved up one character, compacting the text. This is generally only visible on the line in which the cursor is positioned, since the carriage return and line feed cause the next line to stay at the left edge of the screen. SHIFT "DEL" will delete to the left of the cursor. Holding either of these keys down for more than 1/2 second will cause them to automatically repeat, and will stop when you release the keys.

Positioning the cursor is done by using 5 of the keys at the right hand section of the IMLAC keyboard. The "HOME" key causes the cursor to go to the bottom of the screen, for inputting text

at the end of the current screen contents. The four keys labeled with arrows cause the cursor to move

a) left 1 character ←

b) right 1 character →

c) to the left end of the previous line |

d) to the left end of the next line |

Using these keys, it is possible to position the cursor anywhere in the text file on the screen, and add or delete characters as desired.

For making large movements of the cursor, it is possible to multiply the effects of the keys by holding them down, similar to the "DEL" and "SHIFT-DEL" cases.

While some of these functions may appear complex to use, the IMLAC is very easy to use. The best way to learn appears to be to play with it for 1Ø minutes or so.

Typing CTRL-Ø at any time will clear the screen (and lose any text that was on the screen). If you want what is on the screen to go to your output file, be sure to 'PAGE XMIT' it back to IMEDIT before clearing the screen.

For a description of the editing facilities of the IMLAC, see SOME NOTES ON SIC, by Larry Rubin.

IMEDIT COMMANDS - VERSION 61 AND UP

↑B - followed by number, flaps microtape

↑C - prints listing of available commands.

↑D - followed by standard file specification, deletes the file.

↑E - like TECO EE - followed by file specification; if none given, previous name given to ↑O command or name of input file is used.

↑F - like TECO EF -- name conventions same as ↑E

↑G - pushes states of current input file, allowing user to open a new input file and later return to original input file at the same point where he left.

↑K - followed by string, kills Imedit and passes string to its superior (same as TECO ↑H).

↑L - causes Imedit to send next line of input file to the screen.

↑N - search to end of file, output all passed over.

↑O - followed by device and optional name, directs output to that device.

↑Pn - reads from input file through n form feeds, and sends to output device without sending to Imlac screen.

↑Q - search to end of file output nothing.

↑R - causes Imedit to ask for an input file.

↑S - search to end of page, output all passed over.

↑T - pops input file -- undoes what a ↑G does.

↑U - followed by "DEV:" lists directory of that device.

↑V - search to end of page, output nothing.

↑Wnn - sets number of lines sent to screen at any time by ↑Y, ↑U commands to nn.

↑Xn - reads from input file through n form feeds, but does not send to output file -- i.e., discards that part of the input file.

↑Y - send 3Ø lines, or number last specified by a ↑W command to the screen for editing. If a form feed is encountered, stop at the form feed. Note that form feeds appear on the screen as three '#' characters.

To insert a form feed into your file, insert 3 '#' characters instead. All commands except ↑C, are ended by a "XMIT" to cause the command to be executed.