

EditRand(tm)

A Data File Management Tool

Written and copyright by

Scott Whitehead  
CMD DATA SERVICES  
10413 Firethorn Lane  
Austin, Texas 78750-1740  
Compuserve ID 70701,2123

SYSTEM REQUIREMENTS:

IBM PC/XT/AT/PS2 or 100% compatible

DOS 3.0 or higher

400K available RAM

Hard disk or floppy disk

Color or monochrome monitor

## Copyright

Copyright CMD Data Services, 1988-1994. All rights reserved. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use without the written permission of CMD Data Services.

## Software License Notice

Information in this document is subject to change without notice and does not represent a commitment on the part of CMD Data Services. The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used only in accordance with the terms of the agreement.

## Licenses and Trademarks

EDITRAND is a trademark of CMD Data Services. dBASE is a registered trademark of Borland International. Microsoft is a registered trademark of Microsoft Corporation. All other brand and product names are registered trademarks of their respective companies.

## CMD Data Services Software License Agreement

READ THIS LICENSE AGREEMENT BEFORE USING THIS SOFTWARE. By using this software, you are accepting the terms of this agreement. If you do not accept the terms of this agreement, you may return this entire package, including all supplied disks and documentation, within 10 days for a full refund.

You may use this software on any computer, provided you have physical possession of the accompanying original documentation at all times during the use of the software. You may not temporarily transfer possession of the CMD Data Services software or the right to use the software to another individual. You may permanently transfer the software to another individual if all original materials, including disks and backup copies of the disks, and documentation are delivered to the transferee. You may terminate this license at any time by

destroying all original materials as well as all backup copies of the software.

You may transfer the software to a single floppy disk or single hard disk for the purpose of using the software. You may make two (2) backup copies of the software for security purposes.

#### Limited Warranty

This program assumes the user is thoroughly knowledgeable with DOS operations and the particular data that is edited, and that steps will be taken to prevent accidental loss of data by maintaining separate, timely backups. Every effort has been made to assure that this program satisfies the expressed goal of editing random data files. However, the Shareware evaluation (trial use) version is provided AS IS. CMD Data Services MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.

#### Additional Information

You should read the files README.DOC, WARRANTY.DOC, LICENSE.DOC, and SHR-WARE.DOC supplied with the software. It contains other important information about EDITRAND and shareware. README.DOC may include any additional information that did not make it into the documentation before it went to the printer. The supplied disk also contains soft copies of the forms found in Appendix B of this manual, such as the EDITRAND Bug Report/Enhancement Request Form. Such forms have the extension .FRM. You can print copies of these forms for your use.

\* \* \* \* \*

#### WARNING

EDITRAND is intended to be used with files that are evenly blocked, that is, whose total file size is evenly divisible by the user-specified record size. Although EDITRAND is capable of browsing and editing any DOS file, please be cautious since it is possible to alter the length of a file using EDITRAND, such as with the APPEND or DELETE record operations. Such an operation on a data file with a proprietary format or an executable (.EXE or .COM) could render it useless.

REMEMBER: IT IS POSSIBLE TO DESTROY ANY DOS FILE WITH THE EDITRAND PROGRAM IF THE EDITRAND USER IS NOT INTIMATELY FAMILIAR WITH THE DOS OPERATING SYSTEM AND FILE STRUCTURES.

It is strongly advised that any user of the EDITRAND program read this entire manual before attempting to use the program.

Be sure to read the tutorial in Appendix C in the back of the manual.

## EDITRAND Customization

If you have special editing requirements that EDITRAND does not address, CMD Data Services also provides customization of the EDITRAND program. For example, EDITRAND could be tailored to:

1. Intelligently load and handle specified proprietary data file formats.
2. Handle alternate character sets.
3. Generate indexes or format/prefill records and fields to specification.
4. Perform any specific editing through a defined function key or other triggering mechanism.

If interested, please submit a letter of intention, outlining your requirements to:

Scott Whitehead  
CMD Data Services  
10413 Firethorn Lane  
Austin, TX 78750-1740

Please include a summary technical description of your requirements. An immediate response will be sent to verify that your request has been received, and a detailed response, including cost estimates, will be forthcoming 2 to 4 weeks later.

\* \* \* \* \*

## EDITRAND Support

For technical assistance with EditRand, the preferred channel is on CompuServe, ID 70701,2123. Email is checked several times a week, and more thought can be given to questions and concerns before responding in this way.

For licensing or customization questions, or technical questions from registered customers, call 512-335-0579. For a bug report or enhancement request, it is preferable to fill out and send the supplied Report Form (see Appendix B).

NOTE: Technical assistance will NOT be given to UNREGISTERED users on the phone, but I will answer any and all questions from anyone on CompuServe.

EditRand

---

Table of Contents

INTRODUCTION.....	1
WARNING/DISCLAIMER.....	4
INSTALLATION.....	4
EXECUTING EDITRAND, switches.....	6
BROWSE MODE, description.....	14
SUMMARY OF BROWSE MODE FUNCTIONS.....	17
BROWSE MODE FUNCTIONS.....	18
LOAD new file to be edited.....	18
FILE SELECTION WINDOW.....	21
EDIT the current record.....	23
FETCH next or previous record.....	23
FETCH first or last record in file.....	23
FETCH absolute record.....	24
FETCH relative record.....	24
COUNT word occurrences.....	24
COUNT next word occurrences.....	24
SEARCH for a string.....	24
SEARCH for NEXT string occurrence.....	24
FETCH record via absolute byte number.....	25
INVERSE video rule lines.....	25
BOUNDS - Change string operation boundaries.....	25
ABORT program, lose changes and update ER.INI.....	25
QUIT program, save changes and update ER.INI.....	26
EDIT MODE.....	27
SUMMARY OF EDIT MODE FUNCTIONS.....	28
Basic Navigation Functions.....	31
GO to adjacent characters.....	31
GO to next or previous tab position.....	31
GO to first or last character within Field Bounds...31	
FETCH next or previous record.....	31
FETCH first or last record in file.....	31
FETCH record via absolute byte number.....	31
Primary String Functions.....	33
COUNT word occurrences.....	33
COUNT word occurrences, repeat.....	33
SEARCH for a string.....	34
SEARCH for NEXT string occurrence.....	34
REPLACE string.....	35
REPLACE NEXT repeat previous REPLACE command.....	39
Record Functions.....	40
CUT field bounds string into scratchpad.....	40
PASTE scratchpad into currently displayed record...40	
APPEND new record.....	41
DELETE record.....	42
UNDELETE record or UNDO changes.....	43

EditRand

---

Table of Contents

DELETE RANGE of records.....	43
DYNAMIC Field Boundaries.....	45
Change COLOR of record area.....	45
CASE Alter.....	46
FIND deleted record.....	47
SORT on field bounds interval.....	47
Editing Functions.....	49
TEXT/HEX entry mode.....	49
BINARY NUMBER.....	49
DISPLAY BINARY NUMBER.....	50
TAB stop on/off.....	50
TAB increment.....	52
TAB toggle tab stops off/on.....	53
TAB stops reset to null.....	53
BOUNDS change.....	54
BOUNDS Reset.....	55
CLEAR field bounds.....	55
CLEAR to end of record.....	55
UPDATE dBASE Header Record Count.....	55
Miscellaneous Functions.....	56
SAVE AS to another file.....	56
INVERSE video rule lines.....	58
EXIT to Browse mode.....	58
PURGE DELETIONS.....	58
TOGGLE prompt to purge deleted records.....	59
APPENDIX A - Ancillary String Functions.....	60
APPENDIX B - EDITRAND Forms.....	62
APPENDIX C - EDITRAND Tutorial.....	63

## INTRODUCTION

EDITRAND is a utility that allows the user to browse and edit any data file containing records of static, equal length. Most high level programming languages create this type of data file, and is typically referred to as a RANDOM file. This is due to the fact that any record can be accessed randomly, or directly, without the need to process the entire file searching for the desired record. Also, data files created by many database systems (such as dBASE III) can be handled by EDITRAND, even if they have a proprietary header (see page 9).

Each record is displayed with underlying rules to allow easy field and byte level checking. One record at a time is displayed in its entirety, along with pertinent information such as absolute record number, absolute and relative byte position, record and field boundaries, header offset, and hex/decimal representation.

EDITRAND can be used to analyze random data in ways not possible using the program that created the data. Using EDITRAND, data can be viewed exactly as it exists on disk, allowing speedy verification of program output and intelligent troubleshooting of problem situations. Avoid long hours of testing - use EDITRAND to peek at those random files and find the exact fields and records that are suspected of being corrupt or invalid. Have you found that an application will not allow a change that is desired? Load the data file into EDITRAND, swiftly make the desired changes, and put the data back into operation without a lengthy waiting period.

EDITRAND can be used to operate on any random data file. Key features:

- \* Browse and edit random data files produced by any program, regardless of record structure.
- \* Display an entire record on the screen at one time.



## EditRand

---

- \* Determine exact location of data fields with the use of ruled display and user definable tab stops.
- \* Full text, hexadecimal, binary numeric and ASCII data editing.
- \* Flat data files, data files with any proprietary header, or dBASE III format files.
- \* Cursor key navigation just like ordinary text editors.
- \* Perform standard editing functions such as Search, Replace, Word Count, Cut and Paste.
- \* Perform record functions such as Append, Delete, Copy, Move and Sort.
- \* Full featured binary number operations (IEEE or Microsoft Binary Format).
- \* Restrict operations to a range of records and/or range of columns (this allows the user to focus operations on a particular data field)
- \* Expand or collapse record size by inserting or deleting a new field anywhere in the record
- \* Undelete/Undo changes to records
- \* Create backup files for security purposes
- \* On line help screen showing all defined function keys
- \* Works on files up to 2 gigabytes in size
- \* Maximum record length is:
  - 1600 bytes per record in standard 80 X 25 text mode,
  - 3040 bytes per record in 80 X 43 mode (EGA required),

## EditRand

---

- 3600 bytes per record in 80 X 50 mode (VGA required).

Edit entry operations include standard text, hex, binary (string to binary number conversion), and full ASCII entry. String operations include Search, Search NOT, Search Next, Replace, Replace Global, Replace w/Confirm, Word Count, and Word NOT Count performed in text, hex, binary numeric or ASCII; also strings may be cut into a scratchpad to be inserted (pasted) elsewhere. In addition records may be copied, moved, deleted, appended, and sorted.

Central to most string operations is the concept of Bounds, which allows the user to focus on a range of records and a particular field across the range of records. By confining a string operation to such Bounds (such as Replace or Search), all strings outside the Bounds will be ignored. See the section on Bounds on page 54.

The primary goal of this program is to be able to operate on random data files of unlimited size, so that the ability to edit a particular file is not dependent upon available Random Access Memory to store it in during the edit session. Therefore, this program edits directly to disk. This naturally presents an element of danger since changes made in this manner cannot be directly undone or aborted. To overcome this situation, a backup of the edited file is created from which record changes are undone and through which an entire edit session can be aborted. This backup is named by stripping off the name of the file to be edited, and giving it the extension .ERB. For example, the input file ACCOUNTS.DAT would cause the backup file ACCOUNTS.ERB to be created. If there is not enough disk space for the backup file, the program will terminate with an error message. Also, the sort function requires no additional RAM or disk space, as it sorts the main working file directly on disk after removing any deleted records. Sorting in this manner is typically slow, but again assures the user that file size is not an issue.

## EditRand

---

WARNING - PLEASE READ: EDITRAND is intended to be used with files that are evenly blocked, that is, whose total file size is evenly divisible by the user-specified record size. This typically means random data files. For example, any file created as RANDOM by the BASIC programming language, or as a file of Records in C or Pascal. Although EDITRAND is capable of browsing and editing any DOS file, please be cautious since it is possible to alter the length of a file using EDITRAND, such as with the APPEND or DELETE record operations. Such an operation on a data file with a proprietary format or an executable (.EXE or .COM) would render it useless.

REMEMBER: IT IS POSSIBLE TO DESTROY ANY DOS FILE WITH THE EDITRAND PROGRAM IF THE EDITRAND USER IS NOT INTIMATELY FAMILIAR WITH THE DOS OPERATING SYSTEM AND FILE STRUCTURES.

It is strongly advised that any user of the EDITRAND program read this entire manual before attempting to use the program.

DISCLAIMER: This program assumes the user is thoroughly knowledgeable with DOS operations and the particular data that is edited, and that steps will be taken to prevent accidental loss of data by maintaining separate, timely backups. Every effort has been made to assure that this program satisfies the expressed goal of editing random data files. However, the software and documentation have been provided "as is." No warranty is made by the author whether expressed or implied. Fitness for any task is solely determined by the user.

## INSTALLATION

## EditRand

---

To install EDITRAND, simply copy the file ER.EXE onto a working diskette or hard drive, depending upon your particular needs. In order that EDITRAND may be executed from any drive or subdirectory, make sure that the current PATH environment variable contains the subdirectory in which ER.EXE is installed.

Be aware that EDITRAND will create its defaults file ER.INI in the current default subdirectory any time it is executed. This is to allow customized settings for particular files or sets of files. For more information, see page 13.

## EditRand

---

### EXECUTING EDITRAND

EDITRAND can be executed two ways - by passing as arguments the name and record length of the desired file, or by simply executing with no arguments. When EDITRAND is exited, an initialization file called ER.INI is created within the current working subdirectory. This file contains various items of information about the previous edit session, such as file name and record length. It is this initialization file that makes it possible to execute EDITRAND with no arguments - the previously edited file will be reloaded using the same record length and general operating environment. There are two methods of executing EDITRAND:

Method one: ER filename recordlength [switches]

Method two: ER [switches]

The first time EDITRAND is executed method one will have to be used since there will be no initialization file present. Also, (if ER.INI is present) EDITRAND may be executed passing only the filename, in which case the record length used in the previous session (as found in ER.INI) will be used. The EDITRAND user must determine the proper record length used by EDITRAND. EDITRAND will not try to guess the correct record length, except to default to 1600 bytes if ER.INI is not found and no record length is specified.

Several command line switches are available when executing EDITRAND:

/N No Backup - Prevent a backup from being created. This switch is available if you need to edit a file but do not have sufficient disk space to create a backup. Use this switch with GREAT CAUTION as any changes made are permanent as soon as the changes are made:

YOU CANNOT ABORT FROM AN EDIT SESSION  
THAT IS STARTED WITH THE /N COMMAND LINE  
SWITCH. ANY CHANGES YOU MAKE WILL BE  
PERMANENT.

## EditRand

---

[A warning message is displayed on the bottom line]

For this reason, it is advisable that you make a separate backup (perhaps onto floppy disk) of the file to be edited prior to executing EDITRAND with the /N switch.

/B Browse Only - This is a variation of the /N switch without the danger involved. Use of the /B command line switch prevents a backup from being created but also disables edit mode. (Note that the browse mode function prompts on the bottom line and the help screen [Alt-H] are modified to reflect the lack of edit capabilities.) The /B command line switch should be used when it is only desired to verify the contents of a file without the need for alteration.

/E Erase Backup files - This command line switch will erase backup files after the session is finished with them. Backup files have the extension .ERB. Backup files will continue to be made for undelete and edit abort purposes, but will be deleted as different files are Loaded or when EDITRAND is exited. This means that although a file can be rescued from unintentional changes before ending its session, restoring a file after its session is ended is impossible since its backup will have been erased by EDITRAND.

The above switches /N, /B, and /E are mutually exclusive; only one at a time can be used. If more than one is specified, only the first is honored.

## EditRand

---

/M Use Microsoft Binary Format - The default binary numeric representation is IEEE. If you wish to use the older MBF representation use this switch. The string MBF will appear on the left of the status line. Please note that the binary format utilized by the application program used to maintain the data should be used when editing such files. In other words, if the parent application uses MBF format and you make changes to its data file using the default IEEE format, the next time the application accesses such an altered number, it will NOT be the value intended but something entirely different. This point does not apply to numeric fields that are maintained in pure ASCII format.

/I Use IEEE numeric format. Since IEEE format is the default format, the /I switch has effect only when loading a file after starting EDITRAND (using the "L" command), and the previous mode was MBF as imposed by the /M switch.

/D[T] Load File as dBASE III. This switch will cause EDITRAND to load the specified files as a dBASE III file. This causes the following effects:

1. The dBASE header is treated as an offset, the same as would occur if you used the /O switch.
2. The dBASE record length is used as the EDITRAND record length. After loading, you should see only one dBASE record at a time on the screen.

Once a file has been loaded as a dBASE III file, EDITRAND will attempt to load all subsequently loaded files as dBASE III files. To switch off dBASE mode, use the /On switch (/O0 for zero offset is also legal.)

## EditRand

---

[T] This optional switch will cause EDITRAND to read the field definitions from the dBASE header, and set the EDITRAND tabs to these fields. This will cause any previous tab definitions to be lost. EDITRAND will also read the names of the dBASE fields, their field type (like C for character), length and decimals and display them on the status line as you navigate in Edit mode. This switch must be used in conjunction with the /D switch and should appear as /DT.

NOTE: Other than interpreting the header and record length, EDITRAND performs NO dBASE-intelligent operations automatically. For example, EDITRAND will not update the header, recover damaged files, or generate indexes. It is up to the user of EDITRAND to understand dBASE file structures and make the changes manually using EDITRAND's basic functions. The only exception is the use of the Update dBASE Header function (see page 55); with this function you can update the record count as held in the dBASE header.

/On Create header offset. Use of this switch will cause EDITRAND to treat the first n bytes of the file as an offset header. This means that viewing of records will begin at byte number n + 1, and the first n bytes will not be visible or available for editing. \* A denotation of this offset appears on line 2 of the display, just to the right of the Record and Field boundary status indicators:

R:nnn-nnn F:nnn-nnn H: nnn

The maximum allowable offset is the pre-edit file size.

» Header offset is recorded in ER.INI.

---

\* If it is necessary to make changes to the header, you must reload the same file without the use of the /On switch.



## EditRand

---

The above switches /D[T] and /On are mutually exclusive. If both switches are specified, /D[T] takes precedence if the loaded file has a dBASE structure.

/Vn Video Mode. This switch can be used to force EDITRAND to use 25, 43 or 50 line mode. This switch is only valid if EDITRAND is running on a color monitor. Legal values for n are 25, 43, and 50. You must have an EGA or VGA to specify 43, and you must have a VGA monitor to specify 50. In 43 line mode, the maximum record length is 3040 bytes, and in 50 line mode the maximum record length is 3600 bytes. Also, if you attempt to load a file whose record length is greater than the current maximum record length, EDITRAND will adjust to 43 or 50 line mode if the monitor type will allow it. In other words, if you are running in 80 X 25 mode in DOS and load a file whose record length is 3450 bytes, EDITRAND will attempt to automatically switch to 50 line mode without the use of the /V50 switch.

/MONO Force Monochrome. EDITRAND attempts to automatically detect the type of monitor upon which it is running. However, there may be occasions when it is desirable to use monochrome color attributes even though the default monitor is color. For example, on a plasma screen that is detected as a color monitor, the combination of colors used may be difficult to look at; in such a case it may be preferable to force monochrome attributes. If monochrome is being used, color attributes can be restored by using the /MONO switch again.

» Monochrome setting is recorded in ER.INI.

Unless the /N switch is used, a backup of the edited file is created from which editing changes are undone and from which an entire edit session can be aborted. The name of the

## EditRand

---

backup file is extracted from the name of the file to be edited - the main file name is used and given the extension .ERB (stands for EDITRAND Backup). In other words, if the file to be edited is INVENTORY.PTF, the backup created will be called INVENTORY.ERB. If a file by the same name already exists, it will be erased WITHOUT WARNING.

Command line switches may be included with or without any other command line arguments (such as file name and record length). Switches remain in effect until EDITRAND is exited. However, switches can be changed or negated with the use of the Load function (see the explanation of the Load function on page 18). For example, if EDITRAND is started with the /N switch, no backups will be created for any files Loaded unless a new switch is specified.

If neither the /N nor the /B switches is used, EDITRAND will not permit loading of a file with the extension ERB. In other words, the file PARTS.ERB cannot be edited because its pertinent backup file name is also PARTS.ERB.

EDITRAND will attempt to create the backup file on the same drive and subdirectory as the target file ("target file" is the file that is being edited). If there is not enough disk space to create the backup in the target file drive/subdirectory, EDITRAND will then check to verify if the target file is on a drive other than the current default drive. If so, an attempt will be made to store the backup in the current subdirectory of the current default drive. This is done primarily to assist editing of large files on floppy disks, where insufficient disk space is available on the floppy disk. Be aware that this possibility exists if an attempt is made to restore from a backup file, as the desired backup may not be in the same location as the target file.

When loading a new file, whether from the DOS command line, or using the Load function, you may see the message:

WARNING - Last Record Truncated

If this message appears, it means that the size of the current file is not evenly divisible by the specified record size, and the last physical record is not a full record. If the /On header offset switch was used, this could have an

## EditRand

---

effect. Proceed with caution, as this is an indication that something may not be right - either the loaded file is not the file that was intended, or the correct header offset or record size was not used. It is possible to destroy a file using APPEND Record and DELETE Record in this circumstance.

RELATED FILES - If a data file edited by EDITRAND has pertinent, separate index files or other related data files created by another software program, EDITRAND operations have no dependency or requirements for such files. If changes made using EDITRAND cause such an index file or other file to become invalid, it is the EDITRAND user's responsibility to re-generate indexes or make changes to other files outside of EDITRAND.

See the Load command on page 18 for more information about files.

Please note that throughout the remainder of this manual, various items are denoted with ». This indicates that the item described is saved in the initialization file ER.INI, to be reloaded and used the next time EDITRAND is executed. In summary, these items are:

- \* File name
- \* Record length
- \* Header offset
- \* Case setting
- \* Record View setting
- \* Tab stops
- \* Tab increment
- \* Field bounds
- \* Rule line video setting
- \* Monochrome setting
- \* Offset display format (hex or decimal)

## EditRand

---

- \* Offset shown from beginning of record or field
- \* Prompt to purge deleted records

By default, ER.INI is created in the current default subdirectory. This allows for multiple environments, with different tab settings, etc. That is, a copy of ER.INI is created everywhere you execute EDITRAND. However, it is possible to cause EDITRAND to use one, centrally located copy of ER.INI with the use of an environment variable. If you wish to take advantage of this option, include the following statement in your AUTOEXEC.BAT file:

```
SET ERINIPATH=d:\mypath
```

where d:\mypath is the full path of the subdirectory in which you wish EDITRAND to maintain a single copy of ER.INI.

## EditRand

---

### BROWSE MODE

When the program first loads into memory and displays the input file, the first record appears, and the program will be in Browse mode. Line 1 of the display contains the name of the current file, date and time the file was last modified (MM-DD-YY HH:MM:SS), size of the file versus record length, and current record number versus total number of records. For example:

```
C:\1099INFO.DAT 05-11-88 15:36:07 | Size: 7168/128 |
Record: 1/56
```

If there is insufficient space for the full file path name, the left end of the file path name will be truncated and "... " will be appended. For example, if the file to be edited is

```
E:\CUSTOMER\ACCOUNTS\RECVBLE\DELNQUNT\1988\MYERS.DAT
```

it may appear as

```
E:<>CVBLE\DELNQUNT\1988\MYERS.DAT
```

Line 2 of the display shows Record and Field Bounds and byte offsets (Beginning and Ending absolute byte positions of the current record). For example:

```
R:1-56 F:1-128 Beg:2817 End:2944
```

The numbers 1-56 following R: represent the currently defined range of records (also referred to as record bounds) and the numbers 1-128 following F: represent the currently defined field bounds. The beginning and ending byte positions can be displayed in hexadecimal representation with the Alt-F1 function key.

Line two can also display header offset if a header has been specified. See page 9 for a description of header offsets.

The number of lines devoted to record display can vary according to what display mode EDITRAND is in. For example, in standard 80 X 25 text mode, lines 3 through 22 are used; in 80 X 50 mode lines 3 through 47 are used. The status

## EditRand

---

line (the line above the bottom line of the screen) is used for messages and various prompts, and the bottom line is used primarily for function key prompts.

The data display area is set up as a set of lines depicting a record as broken up into contiguous 80 character segments. Each 80 character segment will be underlined with a rule by which the exact position of a particular field or byte can be readily ascertained. Example of a 400 byte record display can be seen in Figure 1, using a mock employee data record. Notice that the ending zero (0) of each 10 byte interval number will be highlighted when EDITRAND is executed. This is to indicate that the byte position of the zero is the actual ordinal position indicated by the number. For example:

```
|..180....|..190..
      ^
```

This position is byte 180

If the record length is not a multiple of 80, then the last rule line will be truncated.

To allow for longer record lengths, a single generic rule line is employed when there is not sufficient room in the record display area for standard rules. A generic rule has no numbers, only graduation marks at 5 and 10 byte intervals, and is located at the bottom of the record display area.

The rules by default will appear on a blue background (black on a monochrome monitor). They can be made to appear inversely on a white background by pressing Alt-I in either Browse or Edit mode. See INVERSE function on page 58.

No editing operations are allowed in Browse mode, but many other functions are available. The definitions of the Browse Mode functions follow.



EditRand

---

Summary of Browse Mode Functions

Description	Key Definition
LOAD new file to be edited	L
EDIT mode	E
FETCH next or previous record	->, <-
FETCH first or last record in file	Ctrl-Home, Ctrl-End
FETCH absolute record	nnnnnnn
FETCH relative record	+ - nnnnnnn
COUNT word occurrences	W
COUNT word occurrences (Repeat)	Ctrl-W
SEARCH for a string	S
SEARCH for next string occurrence	N
FETCH record via absolute byte number	Alt-B
INVERSE video rule lines	Alt-I
CHANGE string operation column boundaries	F10
Dynamic Fields toggle	Ctrl-F
ABORT program, lose changes and update ER.INI	A
ABORT program, lose changes and DON'T update ER.INI	Ctrl-A
QUIT program, save changes and update ER.INI	Q
QUIT program, save changes and DON'T update ER.INI	Ctrl-Q
HELP (show these definitions)	Alt-H



Browse Mode Functions

LOAD new file to be edited

"L"

A prompt will appear on the bottom line of the display, allowing specification of the new file to be edited. While typing a response, the Escape key may be pressed to abort the Load function and resume Browse mode. Current file will be updated on disk, and new file will be loaded. If the new file does not exist, or its file size is 0 (empty file), it will be treated as a new file, and will be displayed with one empty appended record. \* A new file will be denoted with [New] in front of the file name on line one of the display. The previous record length will be used on the new file. If a different record length is desired, the specified file name should be followed by one or more spaces and the new record length, such as

C:\CUSTOMER\QUARTER1\DIVIDEND.DAT 312

The command line switches, as previously discussed, can be used in conjunction with the Load function. The syntax is the same as when executing EDITRAND from DOS. For example:

C:\CUSTOMER\QUARTER1\DIVIDEND.DAT 312 /B

Any specified switches must be the last parameters as indicated in the above example. See the summary of defined switches below under FILE SELECTION WINDOW.

Special symbols:

- \* Asterisk stands for name of the same file currently being edited. Useful for viewing the same file

---

\* If a dBASE file is loaded with the /DT switch, and it contains a header only - no records, it will be loaded with one empty appended record.

## EditRand

---

with a different record length, for example

\* 256

# Reload previous file. Previously edited file (if any) will be loaded and displayed at last record edited.

If changes are pending from the previously edited file, an opportunity will be given to save those changes before loading the specified file. The following message will appear on the status line:

Save changes (Y/N/Escape) ?

If Y is typed in response to this question, all changes made to the previously edited file will be written to disk and a backup file will be kept (unless the /N or /E switches are in effect). If N is typed, all changes will be lost and no backup file will be kept. If the Escape key is pressed, the Load operation will be aborted.

EditRand

Path: C:\DATA\1993

.	D		08-12-89	04:15:48
..	D		08-12-89	04:15:48
1099A.ERB	A	81900	09-25-90	05:36:00
1099A.DAT		81900	09-25-90	05:36:00
CUSTOMER.DAT		124713	01-15-90	12:36:52
FISCAL87.REC		21800	09-15-90	15:20:04
FISCAL88.REC		34900	09-25-90	06:56:36
FISCAL89.REC		46100	09-25-90	06:38:00
BUDGET.DAT		102400	08-21-91	05:40:40
1099MISC.OUT		230600	01-23-90	05:32:36
TEST.DAT		13560	09-25-90	07:20:08
MYDATA.001	R	19250	04-07-90	13:51:10

Use ↑ ↓ Page Up, Page Down, Ctrl-Home and Ctrl-End to highlight the desired file or directory to expand. Enter Loads file or Expands directory. Alt-D deletes file. Tab alters fields.

2,106 dirs,files                   Item: 3                   3,485,696 bytes free

Mask: \*.\*                           [File specification including ? and \*]  
Switches: /B                       [/S toggles subdirectory display. ]  
Record Length: 420                 [Invalid if /DT switch is used. ]

Figure 2 - The File Selection Window

FILE SELECTION WINDOW <F8> The LOAD function can be used to view groups of files in a selection window. After the Load prompt appears, if F8 is pressed, a group of files will be displayed alphabetically in a window for browsing and selection. The actual files chosen for display depends upon the user-specified file designator and the number of files found. All MSDOS file designation conventions apply, such as the use of the asterisk "\*" and the question mark "?". For example, if the user types C:\EMPLOYEE\\*.DAT in response to the Load prompt, then presses F8, all files in the subdirectory C:\EMPLOYEE that have the extension .DAT will be selected for viewing. For each file that matches the file designator, the selection window will contain the file name, attribute, size, and date and time of last modification. If the specified file(s) are contained within a subdirectory, the full path will be not be shown as part of the file name(s) within the window; however, such path is displayed at the bottom of the screen. If F8 is pressed with no file designator, all files in the current default subdirectory will be displayed. Likewise if a subdirectory path is specified with no file name designation, all files in the specified subdirectory will be displayed. Please note that only normal, archived, and read-only files are displayed within the file selection window. The attribute column of the display (immediately to the right of the file name column) will indicate the type of file: blank for normal file, R for read only, A for archive, and D for subdirectory. A maximum of 1000 files can be displayed. If more than 1000 files match the specified designator a plus sign (+) will appear after the file count (see "count of the files" in the next paragraph).

Subdirectory navigation is possible in the File Selection Window. If a subdirectory is highlighted and carriage return is pressed, the highlighted subdirectory will be displayed using the indicated file mask. To change drives, you must exit the File Selection Window and respecify the file path. If

you do not want to see subdirectories in the File Selection Window, Tab to the Switches fields and include the /S switch (it will disappear after the files are redisplayed.) The current path and file mask will be reused, but no subdirectories will be shown. To bring subdirectories back to the display, repeat the procedure.

Below the file selection window is a smaller window containing usage instructions and a count of the files found for viewing and free disk space. Usage keys are up/down cursor, Page Up/Page Down and Ctrl-Home/Ctrl-End. When the desired file is highlighted, pressing carriage return will cause the selected file to be loaded. The file selection window may be aborted by pressing the Escape key; this will cause the selection window to disappear and allow the user to regain control of the Load prompt. If a file is selected, the currently defined record length will be maintained unless a new record length was specified after the file designator, such as

```
D:\INCOME\92\*.* 230
```

If a new record length is included, it must be preceded by a file designator. Also, if a switch is specified, it will take effect if the file is successfully loaded. Defined switches are discussed at the beginning of this manual. Briefly, the switches are:

```
/N          No backup.
/E          Erase backup files.
/B          Browse only.
/M          Use Microsoft Binary Format.
/I          Use IEEE numeric format (default).
/D[T]      Load File as dBASE [read fields
           as tab definitions.]
```



## EditRand

---

FETCH absolute record <nnnnnnnnn>

Display indicated absolute record. If specified record number is outside the currently defined record bounds, beginning or ending boundary record will be displayed. nnnnnnnn represents a maximum nine digit number.

FETCH relative record <[+ | -] nnnnnnnnn>

Display indicated relative record. If offset record number is outside the currently defined record bounds, beginning or ending boundary record will be displayed. nnnnnnnn represents a maximum nine digit number.

COUNT word occurrences <Alt-W | W>

Please see description under Edit Mode, on page 33. All Search and Count functions are identical under both Browse and Edit modes.

COUNT next word occurrences <Ctrl-W>

Repeat previous count word operation. Please see description under Edit Mode, on page 33. All Search and Count functions are identical under both Browse and Edit modes.

SEARCH for a string <Alt-S | S>

Please see description under Edit Mode, on page 34. All Search and Count functions are identical under both Browse and Edit modes.

SEARCH for NEXT string occurrence <Alt-N | N>

Please see description under Edit Mode, on page 34. All Search and Count functions are identical under both Browse and Edit modes.

## EditRand

---

FETCH record via absolute byte number <Alt-B>

Fetch and display record containing the absolute byte as specified. If specified record number is outside the currently defined record bounds, beginning or ending boundary record will be displayed. Byte number can be specified in either decimal (the default) or hexadecimal (Hnnn).

INVERSE video rule lines <Alt-I>

Toggle the video mode of the underlying rules. Either regular color or inverse color is available depending upon the individual user's needs. Please note that a single generic ruler may be utilized if the current record length is too large for the record to fit in the record display area.

BOUNDS - Change string operation boundaries <F10>

Please see description under Edit Mode, on page 54.

DYNAMIC Field Boundaries <Ctrl-F>

Dynamic field boundaries can be toggled by pressing Ctrl-F. See page 45 for more information about dynamic fields.

ABORT program, lose changes and update ER.INI "A"

Abort EDITRAND: lose changes, update current operating parameters in ER.INI, and exit to DOS. All changes made will be lost. Abort is not available if program was executed with /N option (no backup). No backup file is kept.

ABORT program, lose changes, don't update ER.INI <Ctrl-A>



## EditRand

---

Same as typing A, except the initialization file ER.INI is not updated with the current operating environment. The next time EDITRAND is executed, the same operating environment will be loaded as the current invocation loaded.

QUIT program, save changes and update ER.INI "Q"

Quit EDITRAND: saves changes, update current operating parameters in ER.INI, and exit to DOS. ("save changes" does not apply if EDITRAND was executed with /B [browse only] option.) If changes were made to the previously edited file, a backup will be kept unless the /N or /E switches are in effect.

QUIT program, save changes, don't update ER.INI <Ctrl-Q>

Same as typing Q, except the initialization file ER.INI is not updated with the current operating environment. The next time EDITRAND is executed, the same operating environment will be loaded as the current invocation loaded.

Changes made to current file. If the current file has been altered, the file name on line one of the display will have the ASCII character 241 (the plus/minus sign) prepended to the file name.

## EditRand

---

### EDIT MODE

When in Browse mode, pressing E invokes Edit mode, and places the cursor on byte 1 of the currently displayed record. Notice that line two now has some additional information in the center area:

Hex:20 Dec:32

This information will remain until Edit mode is exited. As the cursor navigates about the various positions of the record, the value of the current byte is shown in both hexadecimal and decimal. In addition, the meaning of the byte offsets as shown on the right of line two change somewhat. Instead of representing the absolute byte offset of the beginning and ending of the current record (as in Browse mode), they now indicate the position of the cursor. Respectively they are:

- (a) Absolute byte position from the beginning of the file. If a header offset is in effect, this count will include the offset.
- (b) The byte Offset from the beginning of the record. This quantity can be toggled to represent the offset from the beginning of the current field with Alt-O.

For example, if the record length is 256, record 12 is currently displayed and the cursor is on byte 128, the following numbers would appear:

Abs:2944      Off:128

Both of these numbers change accordingly with every cursor movement.

In Edit mode, the Browse mode keys are no longer available. A new set of functions and their pertinent keys are defined in Edit mode (although some of the string functions are identical in both Browse and Edit modes). The Edit mode functions are discussed on the following pages and are grouped according to function type.

## EditRand

---

### Summary of Edit Mode Functions

Description	Key Definition
GO to first or last character within defined field Bounds Home, End	
FETCH next or previous record	Ctrl ->, <-
FETCH first or last record in file	Ctrl-Home, End
FETCH record via absolute byte number	Alt-B
COUNT word occurrences	Alt-W
COUNT word occurrences (Repeat)	Ctrl-W
SEARCH for string	Alt-S
SEARCH for next string	Alt-N
STRING replacement	Alt-R
STRING replacement (Repeat)	Alt-G
CUT into Scratchpad	Alt-C
PASTE from Scratchpad	Alt-P
APPEND record	Alt-A
DELETE record	Alt-D
UNDELETE record or undo changes to record	Alt-U
DELETE range of records	Ctrl-D
UNDELETE or undo changes to range of records	Ctrl-U
DYNAMIC field boundaries toggle	Ctrl-F

## EditRand

---

CHANGE colors of cursor area	Page Up/Down
CASE, change to Upper Case	Ctrl-P
CASE, change to Lower Case	Ctrl-O
CASE, change to miXed Case	Ctrl-X
CASE, toggle case of current character	Ctrl-C
FIND deleted record	Alt-Y
SORT (save changes, sort on bounds, redisplay)	Ctrl S
TOGGLE Hex/Text entry mode	F1
INSERT Binary number (Int, Long, Single, Dbl, Currency ?) F2	
SHOW decimal value of binaryInteger	F3
SHOW decimal value of binary Long	F4
SHOW decimal value of binary Single Precision	F5
SHOW decimal value of binary Double Precision	F6
SHOW decimal value of binary Currency	F7
TOGGLE tab stop on/off at cursor position	F9
TAB increment reset	Ctrl-F9
TOGGLE offset from beginning of field or record	Alt-O
TOGGLE decimal/hex offset display	Alt-F1
TOGGLE tab stops off/on	Alt-F9
TAB stops reset to null	Alt-T
CHANGE Bounds (string operation column boundaries)	F10

## EditRand

---

RESET Bounds to physical maximums	Alt-F10
CLEAR field bounds area	Alt-F
CLEAR to end of record	Alt-E
UPDATE dBASE header record count	Alt-Q
SAVE AS - process deletions, save to different file	Alt-V
INVERSE video rule lines	Alt-I
EXIT to Browse mode	Escape
PURGE Deletions	[internal]
TOGGLE prompt to purge deleted records	Alt-Z
HELP (show these definitions)	Alt-H

## Basic Navigation Functions

GO to adjacent characters <all four cursor keys>

Place cursor on adjacent characters. Cursor keys behave the same as in an ordinary text editor, with the exception that they can never cause adjacent record to be fetched and displayed (see other function below). Also, carriage return will cause cursor to be placed in left-most position of next display line.

GO to next or previous tab position <tab | shift tab>

Place the cursor on next or previous tab position. Tab positions are defined/undefined using the F9 key (see description on page 50). If no tab positions are defined, then the standard tab increment is used (see standard tab increment on page 52).

GO to first or last character within Field Bounds<Home | End>

Place cursor on first or last byte within currently defined field bounds.

FETCH next or previous record <Ctrl+[Right|Left] Cursor>

Display adjacent record. Ctrl + Left cursor navigates toward beginning of file; Ctrl + right cursor navigates toward end of file. Beginning and end of file are determined by Record bounds, not true beginning and end of file.

FETCH first or last record in file <Ctrl-Home, Ctrl-End>

Display first or last record in file as determined by Record bounds.

FETCH record via absolute byte number <Alt-B>

## EditRand

---

Fetch and display record containing the absolute byte as specified. If specified record number is outside the currently defined record bounds, beginning or ending boundary record will be displayed. If an offset header has been defined (with the /On switch or through the use of /D dBASE file), absolute beginning of the file is the first displayable byte after the header.

The specified offset is in decimal by default. A hexadecimal location may be specified by preceding the number with H, for example HE3A1.

Primary String Functions

COUNT word occurrences

<Alt-W>

Count number of occurrences of specified string within currently defined Bounds. Counting begins at current cursor position (in Browse mode, counting begins at beginning field bound of currently displayed record). String delimiters are not necessary, but may be used (slashes or double quotes) if significant leading or trailing blanks are used. When specifying a string, you may optionally specify Case (Alt-C), View records (Alt-V), Hex (F1), Binary numbers (F2), or full ASCII entry (Alt-nnn); refer to Appendix A for usage of these ancillary functions.

NOT - It is possible to count the number of records that do not contain the specified word by enclosing the string in delimiters and specifying N as the last character. For example:

Count: /STOCK/N

The last two characters must be /N - no intervening blanks. This example will cause all records that do not contain the word STOCK to be counted. (Note the subtle difference between standard COUNT and COUNT NOT: the number returned by COUNT NOT will never exceed the total number of records in the file.) Record and field bounds also apply to COUNT NOT. COUNT NOT starts at the BEGINNING field bound of the current record.

COUNT word occurrences, repeat

<Ctrl-W>

This will begin the previously defined word count again, from the current cursor position.



SEARCH for a string

<Alt-S>

Search for specified string within currently defined Bounds. Searching begins at current cursor position (in Browse mode, search begins at beginning field bound of currently displayed record). String delimiters are not necessary, but may be used (slashes or double quotes) if significant leading or trailing blanks are used. When specifying a string, you may optionally specify Case (Alt-C), View records (Alt-V), Hex (F1), Binary numbers (F2), or full ASCII entry (Alt-*nnn*); refer to Appendix A for usage of these ancillary functions.

NOT - It is possible to search for a record that does not contain a specified string by enclosing the search string in delimiters and specifying N as the last character. For example

Search: /427.16/N

(The last two characters must be /N - no intervening blanks.) This example will locate the next record (if any) that does not contain the decimal number 427.16. Record and field bounds also apply to SEARCH NOT. SEARCH NOT starts at the BEGINNING field bound of the current record.

A common situation in which Search NOT is useful is when it is known (or suspected) that a certain field should always contain the same information. Using Search NOT such fields that do not contain the desired information can be readily located. Another situation could be when it is known that certain fields were left blank when records within a data file were initially created. Using Search NOT, such fields (perhaps all blanks) can be readily located, provided the field bounds are properly set.

SEARCH for NEXT string occurrence

<Alt-N>

Search for next occurrence of a previously specified string within currently defined Bounds. Searching begins after previously found string, or beginning

## EditRand

---

field bound of currently displayed record if record navigation has occurred since last search. If string is found, cursor will be positioned on the first character of target string (in Browse mode, string will blink on the displayed line). If string is not found, record from which search initiated will be maintained on the display.

If original search was specified as a SEARCH NOT, SEARCH NEXT will also be treated as a SEARCH NOT.

REPLACE string

<Alt-R>

Strings can be replaced in the same manner as many ordinary text editors. A prompt will appear on the bottom line allowing entry of the target and replacement strings. Syntax is:

```
/target string/replacement string/[D|L|U|X|P|Ca][*,?]
```

Note the use of slashes as delimiters - double quotes may also be used as delimiters. Optional switches placed after the last delimiter are:

- D The letter D will cause the deletion of any record that contains the target. This is constrained by the current field boundaries.
- U The letter U will cause the target to be converted to upper case.
- L The letter L will cause the target to be converted to lower case.
- X The letter X will cause the target to be converted to a miXture of upper and lower case. Specifically, the first letter of each word will be capitalized, and all other characters will be converted to lower case.

By default, a word to be converted is defined as either the beginning of the current field or any word preceded by a space or other user-defined characters. The user may define what characters should be treated as word

## EditRand

---

separators for this purpose, with the use of the environmental variable ERMIXSEP. For example:

```
SET ERMIXSEP=123456789 !
```

The above DOS command (executed prior to executing EditRand) tells EditRand to convert any word that is preceded by any number, space or exclamation point.

- P The letter P will cause the target to be overwritten with the contents of the scratchpad. This option will do nothing if the scratchpad is empty.
- Ca The letter C indicates that the target is to be filled with the character indicated by a. a can be any ASCII character. If the target is a discrete string, then the string will be replaced with a string of a the same length as the target string; if column replacement is being used, then entire field bounds area will be filled with the character a.

NOTE: The above five options D, U, L, X, P, and Ca do not require a discrete replacement string - the replacement is implicit. Therefore, the syntax for these options is the same as above except the replacement string is not specified:

```
/target//[D|U|L|X|P|Ca]
```

Also, D, U, L, X, P and Ca are mutually exclusive. Only one may be used at a time. If more than one is specified, the leftmost will be used.

- \* Use of the asterisk causes a global replacement to occur. All occurrences of the target string will be replaced with the replacement string.

? Use of the question mark causes the program to stop and query the user before each replacement is performed. Valid responses to this query are Y (yes), N (no), or Q (quit). If used in conjunction with the \* switch, all occurrences of target string will be located and queried until end of file or the user responds with Q (quit).

Both \* and ? may be used simultaneously. If neither switch is used, only the next occurrence of the target string will be replaced.

Replacement begins at the current cursor position of the current record. Search for the target string only occurs within the currently defined field and record bounds. When specifying a string, you may optionally specify Case (Alt-C), View records (Alt-V), Hex (F1), Binary numbers (F2), or full ASCII entry (Alt-nnn); refer to Appendix A for usage of these ancillary functions. Hex and binary functions can be invoked only when the cursor is in position one of the target string specification (no text has yet been typed); notice that the Hex and binary prompts disappear after the first character has been typed.

Another function available is F7-Replace At Cursor. This prompt will appear only when the cursor is in the second position (after the first delimiter has been typed). By pressing F7 a special icon appears "{C}" which signifies that the replacement string should be inserted at the current edit cursor position. (Merely typing {C} will not enable Replace At Cursor; F7 must be pressed.) For example:

Replace: /{C}/Hello/

This replacement will cause the string Hello to be inserted into the record at the current cursor edit position. (Current edit cursor position can be verified by looking at the number after Off: on the right side of column 2). Subsequent use of Alt-G (replace aGain) will insert Hello anywhere the

## EditRand

---

cursor happens to be in any record. The switches ? and \* may also be used with Replace At Cursor. Replace At Cursor is useful for inserting the same value into the same field across an entire file or group of records. For example, if the field of interest is located between bytes 10 and 25, and it is desired to blank out this field, first perform the following:

- \* Assure that the current record and field bounds are properly set
- \* Assure that the cursor is on byte 10 before pressing Alt-R

The following command will accomplish the desired function:

```
Replace: /{C}/          /*
```

Notice the use of the global replace character \*. This will cause the field between bytes 10 and 25 to be blanked out in all records between the current record and the end of file (or end record bound). Please note that if the {C} icon is backspaced, Replace At Cursor will be disabled.

When specifying a Replace string in Hex or Bin mode, it is possible to specify a non-Hex or non-Binary target or replacement. In other words if you have pressed Alt-R then either F1 or F2, the target or replacement must still be hex or binary, but one or the other can be a normal character string. For example, in hex mode:

```
/0C 0A/\!!!/
```

This will cause the next occurrence of carriage return/line feed to be replaced with a pair of exclamation points. The backslash is used as a special character that tells EditRand to treat what follows as literal characters (that is, don't try to interpret them as hex characters.) The reverse would be

EditRand

---

/\!!!/0C 0A/

REPLACE NEXT repeat previous REPLACE command <Alt-G>

Pressing Alt-G will repeat the previously specified  
REPLACE command.

Record Functions

CUT field bounds string into scratchpad <Alt-C>

The string within the field bounds of the currently displayed record is cut into the scratchpad. Displayed record is unaltered. The current value of the scratchpad and its length are always denoted on the right side of the status line while in edit mode. (Scratchpad string is within curly braces {}, and its length is shown in square brackets [].) If the scratchpad string is more than 8 bytes long, only the left 4 bytes and right 4 bytes are shown with an intervening arrow. Please note that if a new file is loaded (Load option in browse mode), the scratchpad string is maintained.

The only way to nullify the scratchpad string is to CUT from a deleted record.

PASTE scratchpad into currently displayed record <Alt-P>

The string within the scratchpad is inserted into the current record beginning at the current cursor position. If the current cursor position is such that there is insufficient space to insert the entire scratchpad string into the current record, then as much of the scratchpad as can fit will be inserted. If no string has been CUT into the scratchpad (with Alt-C), no action is taken. A scratchpad string cannot be inserted into a deleted record.

The CUT and PASTE operations are useful for copying records (or portions of records). Record move operations are more restrictive, but can be accomplished in the following manner:

1. CUT the record to be moved into the scratchpad (make sure the current field bounds are set correctly)

2. DELETE the same record (the one that was just CUT)

Now, either

3. APPEND a new record and specify that the scratchpad be inserted into it.

or

3. Determine the desired target record position, and make it the currently displayed record.
4. PASTE the current scratchpad into the displayed record. If you are moving the entire record, make sure the cursor is in position 1 of the record. This will effectively delete the old record contents of the target record location since you are pasting over a pre-existing record, so beware.

APPEND new record

<Alt-A>

A prompt will appear asking how many records are to be appended. One or more new records will then be created at the end of file. You can optionally specify that the new records be filled with the scratchpad if it is not empty; otherwise the new records are filled with all spaces, and inserted at the end of the file. This function can be invoked no matter which record is currently displayed as long as the end record bound is the last physical record. The last new record becomes the currently displayed record, ready for data entry.

Records cannot be appended if the current end record bound is not the last physical record in the file (see the discussion of BOUNDS on page 54). If append is completed, end record bound is changed to reflect to the newly appended record.



## EditRand

---

WARNING: If the size of the file currently being edited is not evenly divisible by the record length, the last record will be truncated, preventing cursor movement beyond the end of file. This is okay unless you Append a record, in which case the last record will be padded out to produce a file size that is evenly divisible by the record length; then the new record will be appended. For example, if a file called JOBS.FIL contains 12900 bytes, and is loaded with a record length of 128, the last record will contain only 100 bytes. Appending a record onto this file will increase the file not by 128 bytes, but by 156 bytes: the 28 bytes necessary to fill out the last record plus the 128 for the new appended record. This action represents an effort by the program to maintain an evenly blocked data file. If EDITRAND is used to edit files other than evenly blocked data files, it may not be desirable to Append records for this reason.

DELETE record

<Alt-D>

DEFINITION: Deletion String is a character string inserted into a record to indicate that it is deleted. The Deletion String is defined as a string of nulls (ASCII 0) equal in length to the current record length, with the Greek character Theta (ASCII 237) inserted in position 1.

Delete the currently displayed record. Record is changed to the Deletion String, and further editing of the record is not allowed. Record can be undeleted using Alt-U unless the /N (no backup) option is in effect.

WARNING: If, by rare chance, there are legitimate records in the data file whose contents are also all nulls with ASCII 237 in position 1, they may be interpreted as deleted records and purged during a future purge operation (see PURGE DELETIONS on page 58).

## EditRand

---

When a record is deleted, it will continue to exist (as a deleted record) within the file until you direct EDITRAND to purge deleted records. EDITRAND will prompt you to purge deleted records each time you load a new file or exit from the program.

UNDELETE/UNDO changes to current record <Alt-U>

Restore the current record or field to pre-edit contents. If field bounds are not set, the entire current record is restored; if field bounds are set, you are prompted for Record or Field undo. If field is selected, only the current field is restored to the current record - the remainder of the record is left unaltered.

This operation is performed by reading the same record from the backup created and writing its contents back into the displayed record. This feature will be unavailable if the /N (no backup) option is utilized [see NOTE below]. This obviously makes the /N option rather dangerous. Also, undo is not available for newly appended records. See APPEND on page 41.

NOTE: When a record is fetched and displayed, a temporary copy is made (internally). As changes are made to the displayed record, such changes can be undone by use of this temporary copy, regardless if the /N option or append operation is in effect. However, once any subsequent disk I/O operations are carried out (such as record fetch, Sort, Save As, etc) this temporary copy is destroyed and cannot be used to undo changes. The use of this temporary copy for undo purposes is transparent to the user - the EDITRAND program will decide whether to undo from the temporary copy or from the backup file copy.

DELETE RANGE of records <Ctrl-D>

This function will delete all records that are within the currently defined record bounds. Any

## EditRand

---

records that are already deleted will remain deleted. Before proceeding, the function will ask Delete records lll through uuu (Y/N) ? where lll and uuu are the currently defined beginning and end record bounds. Press Y to perform the deletions, or N to abort deletions. Deletions may be interrupted with the Escape key. This function can be undone with its complimentary function, UNDELETE RANGE, although any records that were deleted manually (Alt-D) prior to invocation of DELETE RANGE will also be undeleted.

WARNING: Use this function cautiously if the /N switch (no backup) is in effect, as any records deleted without the presence of a backup file cannot be undeleted with any of UNDELETE RANGE (Ctrl-U), UNDELETE record (Alt-U), or session Abort.

UNDELETE/UNDO RANGE of records

<Ctrl-U>

This function will undelete or undo changes to all records that are within the currently defined record bounds. The function will ask "Undelete Only/Undo All records lll through uuu (O/A/Esc) ?" where lll and uuu are the currently defined beginning and end record bounds. Press O to only undelete records that contain the deletion string, or A to undo all changes (deleted records as well as textual changes.) Function may be aborted with the Escape key.

If you chose to undo all changes (responded with A) and field bounds are not set, the operation will begin; if field bounds are set, you are prompted for Record or Field undo. If Field is selected, only the current field is restored to each record - the remainder of each record is left unaltered.

The undelete/undo operation is performed from the backup file. This function will not work if the /N switch (no backup) is in effect. Also, any records appended since the start of the session cannot be undone, since there are no corresponding records in the backup file.

DYNAMIC Field Boundaries

<Ctrl-F>

Dynamic field boundaries can be toggled by pressing Ctrl-F. When dynamic field boundaries are on, the field boundary setting will change automatically as the cursor crosses a tab setting. This feature is handy if you want editing operations to always be confined to the field level. Any part of the current record can still be edited, but such operations as Search, Replace, Wordcount, and character insert and delete are confined to the current field boundary setting.

While dynamic field boundaries are in effect, the current field boundaries on line 2 are highlighted in cyan (color monitor).

When dynamic field boundaries are in effect, the current field in the record display area is highlighted using a different foreground and background color than the rest of the record. As the cursor moves from one field to another and the field bounds change automatically, so will the highlighted color area.

If you attempt to redefine boundaries (F10) while dynamic field boundaries are in effect, you will be permitted to alter only the record boundaries.

Change COLOR of record area

<Page Up, Page Down>

If a field bound is set, the current field will have a different foreground and background color than the rest of the record. The foreground and background colors of the area the cursor is currently on can be changed with the use of the Page Up and Page Down keys.

If you are not satisfied with the default colors, experiment with the Page Up and Page Down keys until you find a combination you like.



## EditRand

---

Note that you can globally convert a range of records to upper, lower, or mixed case using the Replace command. See REPLACE on page 35.

FIND deleted record <Alt-Y>

Locate next deleted record. Search begins at currently displayed record and continues to end of file. This function will locate any record that contains the deletion string (see Delete Record Alt-D). Note that as you browse records, a deleted record is denoted with the word Deleted on the status line.

SORT on field bounds interval <Ctrl S>

This function will sort the file based on the content of the currently defined field boundaries. Also, only those records within the currently defined record boundaries will be sorted; all records outside the currently defined record boundaries will maintain their position within the file. Before proceeding with the sort, a prompt will appear, allowing you to select ascending or descending sort order. The sort can be aborted at this point by pressing the Escape key.

Please BEWARE that performing a sort causes ALL editing changes to become PERMANENT (the same affect as though you had Quit the program) - all changes are written to the permanent version of the file, and all deleted records are purged before the sort actually occurs. After the sort, the backup file will reflect the new contents of the file, NOT the previous contents. The sort operation can be interrupted by pressing the Escape key, but the file ordering will likely be scrambled.

The type of sort used by EDITRAND does not require any additional RAM or disk space since the main working file is sorted in place on the disk. However, if there are any pending deleted records, such records will be purged. The purge operation

## EditRand

---

DOES require building a second copy of the file - if there is not sufficient disk space for this temporary second copy, the purge operation AND the sort will fail. See PURGING DELETIONS on page 58.

Once the sort gets under way, EDITRAND passes through the entire file comparing different pairs of records, swapping their locations if they are not in proper order (current pass number is displayed on the bottom line). The number of passes and total length of time required for the sort depends upon the number of records sorted and how chaotic the pre-sort ordering is (a file that is already nearly sorted will not require as long to sort as a file whose ordering is totally random). In general, however, this type of direct-disk sort is relatively slow but gains the advantage of being able to sort a file of unlimited size.

## Editing Functions

Note that as soon as the cursor leaves the first record altered while in Edit Mode, the file name on line one of the display will have the ASCII character 241 (the plus/minus sign) prepended to the file name. This allows quick acknowledgment of file edit changes.

TEXT/HEX entry mode <F1>

Data can be typed directly into the currently displayed record in either Text or Hex mode (Text is the default). In text mode, normal alphanumeric keys can be used to enter data directly into the record. Also, while in text mode, any of the 255 ASCII characters can be typed by pressing the Alt key and typing a number between 0 and 255 on the numeric keypad. When the Alt key is released, the representative character is inserted into the current position of the record.

If hex mode is desired, press the F1 key (notice that the string "Hex Mode" appears momentarily on the status line, and that the F1 definition on the bottom line is changed from "F1=Hex" to "F1=Text"). While in hex mode, only the characters 0 - 9 and A - F are legal. A hex character is inserted into the current position by typing two of these characters (one is not sufficient). When the first character is typed, it appears in reverse video; when the second character is typed, it is appended to the first to form the desired hex character, and this character is inserted into the record in its ASCII representation. To switch back to Text mode, simply press the F1 key again.

BINARY NUMBER <F2>

This function is used to insert binary numbers into the currently displayed record. Integers, Longs,





already defined at the current position, it will become undefined. The tab stop status for each byte position is denoted in the following manner:

Rule lines are visible: if the current byte position is defined as a tab stop, the rule for that position will be overlaid with a "T"; if the byte position is not currently defined as a tab stop the standard rule character for that position will be seen instead.

Rule lines are not visible; generic rule line is at bottom of record display area: if the current byte position is defined as a tab stop, a "T" will appear in the generic rule line directly under the current cursor position.

If one or more tab stops are defined, pressing the tab key will cause the cursor to be placed on the next defined tab stop after the current cursor position. Alternately, pressing the shift/tab key will cause the cursor to be placed on the next defined tab stop before the current cursor position. If no tab stops are currently defined or tab stops are turned off (with Alt-F9), the tab increment will instead be used to jump the cursor ahead (tab key) or backward (shift/tab key) the number positions as currently defined by the tab increment (see TAB increment function below; if the tab increment has never been set, its default value will be 8). Tab stops are not restricted in any way by the field boundary settings. If tab stops are currently turned off (Alt-F9), pressing F9 will generate an error message on the status line. Tab stops must be turned on for F9 to function.

The principal usefulness of setting tab stops is to define field boundaries of the current record layout. One method would be to have a hardcopy of the record layout next to the machine and locate the beginning of each field with the cursor keys; at each beginning field position, press F9 to define a tab stop. Thereafter, you may jump around to the

## EditRand

---

various fields by using tab and shift/tab. Tab stops will not have to be rekeyed since tab stops are one of the items that are saved into the defaults file ER.INI.

NOTE: If you change files that have varying field layouts you will have to reset the tab stops. The last defined set of tab stops will be the one that is written to ER.INI as you exit the program. The exception to this is when various files are maintained in separate subdirectories; since ER.INI is typically created in the current default subdirectory, maintaining data files in separate subdirectories will allow multiple EDITRAND environments to exist simultaneously in the form of different copies of ER.INI. See page 13 for more information about ER.INI.

» The tab stops are saved in ER.INI.

### TAB increment

<Ctrl-F9>

This function is used to change the tab increment. A prompt will appear at the bottom of the screen. Press Escape to abort and save previous value. The default value is 8. The value of the tab increment is used to move the cursor forward (tab key) or backward (shift/tab), but only if:

- \* no tab stops are defined, or
- \* tab stops are currently turned off (see TAB stops off)

If tab stops have never been defined, have been reset to null (with Alt-T), or have been turned off with (Alt-F9) then pressing tab or shift/tab will cause the cursor to move the number of spaces as defined by the current tab increment.

» The tab increment is saved in ER.INI.

## EditRand

---

TAB toggle tab stops off/on <Alt-F9>

Press Alt-F9 to turn tab stops off or on. An indicative message will appear on the status line. If tab stops have been defined (with F9), then subsequently turned off with Alt-F9, the tab increment will turn on. When tab stops are turned off, they are still defined and will be saved to ER.INI if EDITRAND is exited. When tabs stops are again turned on, the tab increment is turned off. While tab stops are turned off, the tab stop define/undefine key (F9) cannot be used.

Notice that the tab stop markers (T) within the rules will reflect whether tabs are turned on (Ts will appear) or turned off (Ts will disappear).

TAB stops reset to null <Alt-T>

Pressing Alt-T will cause all currently defined tab stops to become null - no tab stops will be defined thereafter. There is no way to undo this function.

TOGGLE offset from beginning of field or record <Alt-O>

Pressing Alt-O will toggle the quantity displayed after Off: on line two:

- \* The number of bytes the cursor is from the beginning of the current record (the default), or
- \* The number of bytes from the beginning of the current field, if field bounds are set.

» The offset display type is saved in ER.INI.

## EditRand

---

TOGGLE decimal/hex offset display <Alt-F1>

Pressing Alt-F1 will toggle the position numerics on line two between decimal (the default) and hexadecimal.

» The offset display representation is saved in ER.INI.

BOUNDS change <F10>

This function allows the record and field boundaries to be altered. A dual prompt will appear at the bottom of the screen, one for record bounds and one for field bounds. Pressing carriage return at either prompt will cause the previous setting to be maintained. Pressing Escape will abort the change. For each setting, both the beginning and end value must be specified, separated by a space. If an illegal set of values are specified, an error message will be displayed and the old values will be kept.

If the currently displayed record is outside the new record bounds, the nearest boundary record will be fetched and displayed. Records outside of the record bounds are not available (cannot be fetched) for ANY purpose. To access such records, the record bounds must be again altered to include such records. If the end record bound is not the last record in the file, records cannot be appended. If the end record bound IS the last record in the file, the end bound is incremented by 1 for each new record appended.

The following edit functions are constrained by the current record and field bounds: SEARCH, COUNT, REPLACE, FETCH, APPEND, FIND (deleted record), and SORT. See the description and usage for each function for more information. Also when in edit mode, the Insert and Delete keys will have no effect if the cursor is outside of the currently defined field bounds. Field bounds can be set to change automatically to reflect tab settings with Dynamic Field Boundaries (see page 45).

## EditRand

---

Note that the new values will be reflected on the left side of the screen on line 2: R:nnn-nnn for record bounds and F:nnn-nnn for field bounds.

» The bound settings are saved in ER.INI.

BOUNDS Reset <Alt-F10>

Pressing Alt-F10 will cause field and record beginning and end bounds to be reset to physical minimum and maximum. If dynamic field boundaries are in effect, only record bounds will be reset.

CLEAR field bounds <Alt-F>

Pressing Alt-F will insert blanks within the currently defined field bounds area of the currently displayed record, regardless of the cursor position.

CLEAR to end of record <Alt-E>

Pressing Alt-E will insert blanks within the currently displayed record, beginning at the current cursor position extending to the end of the record. Field bounds have no effect upon this function.

UPDATE dBASE Header Record Count <Alt-Q>

If edit file has been loaded as a dBASE file with the use of the /D switch, you can update the record count by pressing Alt-Q. The current number of records shown on line one will be the number that is used, even if record deletions are pending.

Miscellaneous Functions

SAVE AS to another file

<Alt-V>

SAVE AS allows the current file to be output in several ways:

- \* Make a copy of the file and give it a different name.
- \* Copy only portions of the current file, and put the portions in another file. This is done by setting the field and record boundaries to either include or exclude the desired portions.
- \* Make a copy of the current file, adding an additional field to the new output file.
- \* Optionally add carriage return/line feed pairs to each output record. This facilitates editing the output file with an ordinary text editor, or to create a new data file to be used by some other program.
- \* Send the whole file, or portions of it (as constrained by field and record boundaries), to a parallel printer, such as LPT1: or LPT2:.. When printing in this manner, EditRand automatically appends a carriage return/line feed pair to each output record.

If you wish to send to a serial printer, first output to a file, then use the DOS PRINT command to send the file to your serial printer.

Several prompts will appear to guide you through the selection process. Each prompt will allow 2 or more possible responses. The default response will be shown in square brackets. This is the response that will be used if you just press Enter. At any

## EditRand

---

prompt, you can press the Escape key to abort the entire operation.

Below are the prompts you will see, in order (you may not see all of the following prompt - the ones you actually see will depend on how you respond):

1. Save As. Enter the name of the file you wish to output to. This would be LPT1 or LPT2 if you are going to print the file.
2. Purge deleted records ? This prompt will appear only if you have deleted records in the current file. If you answer Yes, the output file (or printer) will not contain any of the deleted records. This will have no effect on the current file.
3. Copy Entire file, or insert New field at cursor position? Respond with Entire (the default) to output the whole file, or respond with New to insert a new field. You will be allowed to specify the length of the new field, and if it should go before or after the current cursor position. This prompt appears only if there is no field bound set.
3. Output Entire file, or record/field bounds Inclusive/eXclusive? Respond with Entire (the default) to output the whole file, or respond with Inclusive to output only those records that fall within the current record boundaries and only the field within the current field boundary. Respond with eXclusive to output all parts of the current file that fall outside of the current field and record boundaries. This prompt will not appear if field and record boundaries are set to physical maximums.
4. Add carriage return/line feeds to each record? The default response is No. Press Y if you want carriage return/line feeds.



## EditRand

---

This prompt will not appear if you specified a printer.

5. Output file header ? This prompt will appear only if the current file has an offset header. You can tell this by looking for H: nnn on line 2 of the display. If you are editing a dBASE III file, there will be a header. Answer no if you want to output data only. Keep in mind that saying No will cause the output file to lose all control information that is present in the current file's header.

INVERSE video rule lines

<Alt-I>

Toggle the video mode of the underlying rules. Either regular color or inverse color is available depending upon the individual user's needs.

» The rule line video setting is saved in ER.INI.

EXIT to Browse mode

<Escape>

Press the Escape key to exit from Edit mode and place program in Browse mode. Exiting to Browse has no effect on changes made in Edit mode, and Edit mode can be resumed at any time.

It is necessary to Escape to Browse mode to exit the program (Quit or Abort) or Load a new file.

PURGE DELETIONS

Purging deletions is not a directly accessible function, but one that occurs as a result of several other functions, such as Save As, Load, and Quit. EDITRAND will ask if you wish to purge deleted records, but only if it has detected such records. EDITRAND detects deleted records either from direct deletions by the user (Alt-D) or if any record with

## EditRand

---

the deletion string is viewed by EDITRAND during the edit session.

The purging of deleted records requires the creation of a temporary file to hold all non-deleted records as the current file is processed. For this reason it is possible to receive an insufficient disk space message when purging deletions. If this occurs, two options are available: Save As to a different drive that contains sufficient free space, or undelete all records with Ctrl-U (Undelete/Undo Range of records).

During the purge operation, two running numeric totals appear on the bottom of the screen. These two numbers appear as a pair such as 89, 76. The first number indicates the record that is currently being inspected for deletion status. The second number indicates the total number of non-deleted records that have been written to the temporary file. (Remember that this temporary file, if no errors occur, will become the new copy of the file that is being edited.) This means that when processing is completed, the second number will be the total number of records in the current file minus the number of deletions that have been made.

All records in the current file that match EDITRAND's idea of a deleted record (the Deletion String is defined as all nulls with ASCII 237 in position 1) will be purged, whether as a result of an actual delete operation (Alt-D) or some other external non-EDITRAND operation.

TOGGLE prompt to purge deleted records <Alt-Z>

It is possible to turn off the prompt to purge deleted records. This will disallow purging of deleted records. To turn the prompt back on, press Alt-Z again.

» Prompt setting is recorded in ER.INI.

APPENDIX A - Ancillary String Functions

The following functions are available when invoking SEARCH, COUNT and REPLACE:

CASE           The left side of the status line will display the current case setting. Pressing Alt-C will toggle the setting. Use of the SEARCH and COUNT functions are the only way to reset case.

» Case setting is recorded in ER.INI.

VIEW           The right side of the status line will display the current View setting. When View is ON, all records being searched are displayed as they "flash" by, and record byte offsets on line 2. This function serves only to let you get a general idea of the content of the records as they are being searched. In most cases, the records will not be on the screen long enough to be useful. Pressing Alt-V will toggle the View setting.

» View setting is recorded in ER.INI.

HEX            Press F1 to engage hex search mode. You must press F1 while the cursor is in position one (before typing any text). To ensure proper interpretation of the hex string, type two-byte hex digits separated by a single space. For example, if searching for a pair of tab characters, type 09 09, not 0909. Subsequent SEARCH NEXT or REPLACE NEXT will use the same hex string.

Binary           number. To search for a binary number, press F2 while the cursor is still in position one. A prompt will appear on the status line allowing you to specify whether to search for an Integer (2 bytes), Long (4 byte integer), Single Precision (4 byte float), Double Precision (8 byte float), or

Currency (8 bytes). After specifying which type, a prompt will appear on the bottom line allowing entry of a decimal number. Press carriage return to complete. The specified decimal number will be converted to binary form using either IEEE (the default) or Microsoft Binary Format (must use the /M switch), and the search will begin. Subsequent SEARCH NEXT or REPLACE NEXT will use the same binary string. (NOTE: Microsoft Binary Format will be in effect only if the string MBF appears on the left side of the status line.)

Alt-*nnn* Any of the 255 ASCII characters, except ASCII 0, may be typed with the use of the Alt key and the numeric keypad. This function is not available if either of the Hex or Binary functions are in effect.

APPENDIX B - EDITRAND Forms

The accompanying forms are for your use. You may make copies or use the softcopy equivalent of these forms supplied with the EditRand distribution disk.

EDITRAND Registration Form

This program is not public domain or freeware. It is marketed on the shareware principle and a payment of \$29 is required after an initial trial period of 30 days. With your payment of \$29 you will receive (1) A printed, bound up-to-date copy of the documentation and the latest version on disk without the UNREGISTERED VERSION reminder; (2) the next release free; (3) a copy of the NUMRECS utility. NUMRECS will tell you the number of records of any size that will fit on the total and available disk space of any drive (or all) drives.

Please use this form to submit your payment. If you are already a registered customer, please use this form to purchase additional copies, or pass on to a friend or associate.

Date: \_\_\_\_\_

Name: \_\_\_\_\_

Address: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Phone: \_\_\_\_\_

Each registered version of EditRand is personalized with the licensee's name or company name. Enter the name you wish to use below (must be a person's name or a company name):

Licensee: \_\_\_\_\_

Current Version of EDITRAND: \_\_\_\_\_

From where did you receive your sharware copy of EditRand ? \_\_\_\_\_  
\_\_\_\_\_

Licenses ordered:	Media size	quantities	\$ Total
\$29 each	3.5"	(720K) _____	_____
	5.25"	(360K) _____	_____

International Orders add \$5 ground, or \$15 air: \_\_\_\_\_

Total amount of your order: \$ \_\_\_\_\_

Comments: \_\_\_\_\_

Mail this form with payment to: Scott Whitehead  
CMD Data Services  
10413 Firethorn Lane  
Austin, TX 78750-1740

| Thank you for your registration and payment! |

EDITRAND Report Form

Please use this form (or a copy of it) to report bugs, enhancement requests, or just general comments and correspondence. If there is not enough room on this sheet, please attach any additional pages required.

Date: \_\_\_\_\_ EDITRAND Version: \_\_\_\_\_

Name: \_\_\_\_\_

Address: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Phone: \_\_\_\_\_

Check one:      BUG Report \_\_\_\_\_      Enhancement Request \_\_\_\_\_

For enhancement, 1. through 4. are not required.

1. Operating System/Version: \_\_\_\_\_

2. Machine manufacturer, model: \_\_\_\_\_

3. Memory configuration: \_\_\_\_\_

4. What other programs were running (TSRs, network, etc):  
\_\_\_\_\_  
\_\_\_\_\_

5. Description: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Please mail to:      Scott Whitehead  
                          CMD Data Services  
                          10413 Firethorn Lane  
                          Austin, TX 78750-1740

## EditRand

---

### Index

43 or 50 line mode 10  
Abort 6, 7, 10, 25  
ABORT program 17  
Alt-nnn 33, 34, 37, 61  
Ancillary string functions 60  
APPEND record 4, 28, 41, 54  
Arguments 6  
Asterisk 18, 21, 36  
Backup 3, 11  
Binary number 3, 29, 34, 37, 49, 50, 60  
Bounds 3, 14, 25, 54  
Bounds Reset 55  
Browse mode 15, 35, 58  
Browse Mode Functions Summary 17  
Browse Only 7, 23  
Carriage return/line feed 56  
Case 34, 37, 60  
CHANGE bounds 17, 29  
CHANGE colors of cursor area 29  
CLEAR field bounds 30, 55  
CLEAR to end of record 30, 55  
Color 29, 45  
Command line switches 11, 22  
COUNT 17, 24, 33, 54, 60  
COUNT NOT 33  
COUNT word occurrences 28  
COUNT word occurrences (Repeat) 28  
CUT into Scratchpad 28, 40  
Data Display 15  
DBASE 2, 8, 9, 22  
DBASE Fields 9  
DBASE file structures 9  
DELETE range of records 28, 43  
DELETE record 4, 28, 42  
Deletion String 42, 44, 47, 59  
DISCLAIMER 4  
Dynamic field boundaries 25, 28, 45, 54, 55  
Dynamic Fields 17  
EDIT 17, 23, 27  
Edit entry mode (Text or Hexidecimal) 49  
Edit Mode Functions (Summary) 28  
Edit Mode Navigation 31  
ER.INI 6, 9, 10, 12, 17, 25, 26, 46, 52, 53, 54, 55, 58,  
59, 60  
Erase Backup files 7

---



## EditRand

---

### Index

ERINIPATH environment variable 13  
ERMIXSEP environment variable 36, 46  
Executing EDITRAND 6  
EXIT to Browse mode 30  
FETCH 17, 54  
Fetch absolute byte 25  
Fetch absolute record 24  
FETCH first or last record 31  
FETCH first or last record in file 28  
FETCH next or previous record 28, 31  
Fetch record 23  
FETCH record via absolute byte number 28, 32  
Fetch relative record 24  
Field boundaries 51  
Field bounds 54, 55  
Figure 1 - Example EditRand Screen 16  
Figure 2 - The File Selection Window 20  
File Selection Window 21  
FIND deleted record 29, 54  
Force Monochrome 10  
Generic Rule 15, 25, 51  
Global replacement 36  
Header Offset 9  
Help 7, 17, 30  
Hex 3, 27, 33, 34, 37, 49, 60, 61  
Hexidecimal 14, 32, 38  
Index files 12  
Initialization file 12  
Insert Binary Number 49  
INVERSE video 17, 25, 30, 58  
Last Record Truncated 11  
Load 11, 17, 18, 21, 40, 58  
Load File as dBASE III 8  
Microsoft Binary Format 8  
Move record 40  
No Backup 6, 25, 42, 43  
PASTE from Scratchpad 28, 36, 40, 41  
Printing 57  
Purge 43, 47, 59  
PURGE Deletions 30  
Purging deletions 58  
Quit 26, 58  
QUIT program 17  
Read-only 21  
Record bounds 23, 54

## EditRand

---

### Index

Record Length 14, 25  
Record Number 14  
Relational/Related Files 12  
Reload previous file 19  
Replace 3, 35, 54, 60  
Replace At Cursor 37  
Replace Next 37, 39, 60, 61  
RESET Bounds to physical maximums 30  
Rules 15  
Sample EDITRAND screen 16  
SAVE AS 30, 58  
Scratchpad 3, 36, 40, 41  
Search 3, 17, 24, 28, 34, 54, 60  
Search for Binary number 60  
SEARCH NEXT 28, 34, 35, 60, 61  
SEARCH NOT 34, 35  
SHOW decimal value of binary number 29  
SORT 29, 47, 54  
Special symbols 18  
STRING replacement 28  
STRING replacement (Repeat) 28  
TAB increment reset 29  
TAB increment set 52  
TAB stop on/off 50  
TAB stops reset to null 29, 53  
TAB stops toggle 53  
TOGGLE decimal/hex offset display 29, 54  
TOGGLE Hex/Text entry mode 29  
TOGGLE offset from beginning of field or record 29, 53  
TOGGLE prompt to purge deleted records 30  
TOGGLE tab stop on/off at cursor position 29  
TOGGLE tab stops off/on 29  
UNDELETE or undo changes to range of records 28  
UNDELETE range of records 44  
UNDELETE record 28, 43  
UNDO range of records 44  
UNDO record 43  
UPDATE dBASE header record count 30  
VGA monitor 10  
Video Mode 10  
View records 34, 37, 60  
WARNING 4, 7, 11, 23, 42, 44

## EditRand Tutorial

---

## EditRand Tutorial

This tutorial will demonstrate many of EditRand's useful functions. To begin, follow these steps:

- a. Create a new subdirectory and make it the default. For example:

```
MD ERTEST
CD ERTEST
```

- b. Now copy the necessary files from the EditRand distribution diskette:

```
COPY A:\1099.DAT
COPY A:\1099.DBF
COPY A:\ER.INI
```

The file 1099.DAT contains sample 1099 data as reported to the Internal Revenue Service by businesses each year. The file 1099.DBF is the same data in a dBASE III format. The file ER.INI contains settings necessary to this tutorial.

In the following exercises, the file 1099.DAT will be used. The file 1099.DBF is used only in the section titled The FILE SELECTION SCREEN.

- c. Copy the EditRand executable program, ER.EXE, from the distribution disk onto your hard disk or floppy disk. You can locate ER.EXE anywhere you wish, but if it is not in the current default subdirectory, make sure it is in a subdirectory that is in the current PATH. The currently defined PATH can be checked with the PATH command.
- d. Execute the EditRand program in the following manner:

```
ER 1099.DAT 420
```

This will cause EditRand to load the sample file using a record length of 420 bytes. Also, since

## EditRand Tutorial

---

the supplied EditRand initialization file ER.INI was copied into the current subdirectory, the tab settings in it will be used for 1099.DAT.

You are now ready to begin this tutorial. It is best to follow the outlined steps chronologically, since some operations depend on previous steps.

## EditRand Tutorial

---

### 1. FOR STARTERS

You should see on your screen a 420 byte record. This is record 1 of the file 1099.DAT. At the bottom of the screen are several defined function keys. There are more function keys than this, but these are the most important ones. To see a brief description of the rest of them, press Alt-H. Press any key to restore the EditRand main screen.

Currently EditRand is in BROWSE mode and no editing is possible. You can perform Search and Word Count, as well as fetch other records, but no alterations are possible.

Try fetching other records:

- a. Press right cursor. You should see record 2 appear. To be sure, look at the right of the top line of the display. To the right of Record: you will see ccc/ttt where ccc is the current record and ttt is the total number of records in the file.
- b. Press Ctrl-End. The last record of the file should appear.
- c. Press Ctrl-Home. The first record of the file should re-appear.
- d. To switch to EDIT mode, press "E". The cursor should appear on byte (character) one of the current record.

### 2. TABS and BOUNDARIES

Look at the Record and Field Boundary definitions on line two, denoted by

R: 1-80 F: 1-420

This says that record and field bounds are both set to physical maximums and all string operations will operate on the full extent of the file. (There are 80 records and their record length is 420 bytes.)

## EditRand Tutorial

---

The supplied ER.INI initialization file caused the file to be loaded with tab stops defined for the file 1099.DAT. Notice the character "T" at various points on the display in the rulers under the record lines.

- a. Press Tab once and notice that the cursor has moved to the next tab stop. Press Tab several more times to get a feel for the functionality.
- b. Now press Shift-Tab until the cursor returns to the first Tab stop.
- c. Press the End key. The cursor should be in the last position of the record, byte 420. This can be verified by looking on the right side of line two - the offset field should say Off: 420. This means the cursor is this many bytes from the beginning of the current record.
- d. Press the Home key. The cursor should be in position 1.

Now you will experiment with setting Record and Field Boundaries:

- e. Press the F10 key. A prompt will appear at the bottom of the screen, allowing you to specify both Record and Field Boundaries. Type "10 70" for Record Boundary and press enter. Now type "51 140" for Field Boundary and press the enter key. Bytes 51 to 140 represent the numeric, or money portion of the 1099 records.
- f. Notice that the beginning record has changed to 10. Press Ctrl-End and see that you cannot fetch any record beyond record 70. Press Ctrl-Home to return to record 10.
- g. Press the Home key to position the cursor on byte 51. Note that with Field Boundary set, no matter where the cursor is, pressing Home or End will always place the cursor at the beginning or end of the current Field Boundary area. Also, you are

## EditRand Tutorial

---

free to set the field boundary to any area, regardless of the tab settings.

- h. Press Alt-S. You will see a prompt to specify a search string. Type MONROE and press Enter. EditRand will search only records 10 to 70 of the file, and will only look in the Field Boundary area of each record. You should receive a message telling you that the string was not found. Now change the Field Boundary, using F10. Press Enter for Record Bounds (this will cause the previously specified Record Boundary to remain in effect), and type "162 201" for the Field Boundary (the First Payee Name field). Now press Alt-N to repeat the previous search. The string MONROE should be found in record 68. This demonstrates how EditRand can focus on a specific field and ignore other areas of the file.
- i. Now press Alt-F10 to reset Record and Field Boundaries to physical maximums.

Now to demonstrate the power of Dynamic Field Boundaries. First turn Dynamic Field Boundaries on by pressing Ctrl-F. A message should appear on the status line (last line from the bottom) indicating that Dynamic Field Boundaries are turned On. If you have a color monitor, you can tell by looking at the field boundaries on the left of the second line of the display, after F: - the field boundaries will appear as cyan. Now to demonstrate the power of Dynamic Field Boundaries.

- j. Press Tab key one time. Notice that the Field Boundary (as denoted to the right of F:) has changed to reflect the current tab setting to either side of the cursor position.
- k. Now press the right cursor key until the cursor crosses over the next tab. Notice that the Field Boundary changed automatically when the cursor traveled into the next field. A field is defined as the area between any two tab stops. Generally, a field is considered a single item of information that is distinct from any other item of

## EditRand Tutorial

---

information in the record, such as Name, Address, or Social Security Number.

1. Now press the End key. When Dynamic Field Boundaries are On, EditRand always looks at the tab settings immediately to either side of the current cursor position, and uses those as its Home and End locations. Press the Home key to return to the beginning of the current field.

Now to highlight a few other useful functions. With Dynamic Field Boundaries still set On, do the following:

- m. Press the delete key several times. Notice that characters shift to the left, but only within the current field. This would be true anytime Field Boundaries are set, whether manually (with F10) or with Dynamic Field Boundaries.
- n. Now press the insert key and press the space bar (or any type any characters) and notice that characters to the right of the cursor key shift to the right, but only up to the next tab setting to the right.

Note: If no Field Boundary was set, use of the delete and insert key would affect all characters from the cursor to the end of the record.

- o. Now press Alt-U to undo the alterations you just made. When prompted for Record or Field Boundaries, type F for Field Boundaries. The field you just altered will be restored to its previous contents; if you had made changes to any other portion of the current record, they would have been left as you made them, or you could have responded with R to Undo the entire record.



## EditRand Tutorial

---

When you executed EditRand, you may have noticed a brief message that said "Copying to 1099.ERB". The file 1099.ERB is an exact copy of the file 1099.DAT and is used for backup purposes. EditRand is able to undo changes to records by making a copy of the same record in the backup file and copying it back into the current record.

- p. Using the Tab key, locate the cursor on the field that reads "SOME TOWN". Press Alt-R to perform a replacement. Now, suppose the task you wish to perform is to change "SOME TOWN" to your town's name. Two approaches are possible:

1. Tell EditRand to replace all occurrences of "SOME TOWN" with your town's name. Type:

```
/SOME TOWN/MY TOWN/*
```

and press Enter. (The asterisk tells EditRand to perform this replacement across all records currently defined by Record Boundaries.) All occurrences of SOME TOWN have been changed to MY TOWN. This works fine if you know this was exactly what you wanted. But what if several records had some other name, beside SOME TOWN, in them?

2. Press Ctrl-U, All Records, Field Bounds to restore the town name field to its previous contents. Then press Ctrl-Home to fetch the first record (actually record 10.) Now tell EditRand to insert your town's name in ALL records at this position (or field), regardless of what was there before. After typing the first slash delimiter press F7 and the Replace at Cursor symbol "{C}" appears. Finish the replacement string as you normally would:

```
/{C}/MY TOWN /*
```

This will cause "MY TOWN " (or whatever town name you wish) to be inserted at the same

position across all records. Notice the two extra blanks after MY TOWN - this is so the last two characters of SOME TOWN will be overwritten with blanks. Another variation would be:

```
{C}/MY TOWN /*?
```

This would cause EditRand to pause at each record before inserting "MY TOWN " and ask if you wish to perform the replacement. This way you can be very selective.

- q. Now press Ctrl-Home to go back to record 10. Again position the cursor on the same field that contains the new town name you just inserted. Suppose you want to change "MY TOWN" to mixed case. Press Alt-R and type:

```
{C}//X*
```

This replacement string says to convert to lower case, but capitalize the beginning letter of all words found across all records. No replacement string was needed since it is implicit that all characters are to be converted. You could have typed:

```
/MY TOWN//X*
```

to convert only occurrences of "MY TOWN" to lower case, or even:

```
/MY TOWN/My Town/*
```

This is the same result as above, but caused you more keystrokes.

Other useful replace options:

L Cause the target to be converted to lower case.

U Cause the target to be converted to upper case.

## EditRand Tutorial

---

Ca Fill the target area with a string of the character following the C ("a" could be any character.)

P Fill the target with the contents of the scratchpad (see the manual for a description of the scratchpad.)

D Delete any record where the target is found.

See the manual for details.

r. Now you have decided that the above replacement operations were all a BIG mistake - you didn't mean any of this. Press Ctrl-U to cause all records within current boundary (still set to 10 - 100) to be Undone and restored to their original contents. When asked if you want to restore the entire Record or only the current Field, press R to restore the entire record (across all 91 records, that is.) The town field now says "SOME TOWN" again.

If you were certain that ALL changes you had made to the current file in the current session were unnecessary and you wanted to start over, the easiest way would be to reload the file:

1. Press Escape to return to Browse mode.
2. Type L (for Load), followed by "\*". (The asterisk is a special symbol that stands for the current file name. This is just to prevent you from having to retype it.)  
When prompted with:

Lose changes and continue (Y/N)?

## EditRand Tutorial

---

type Y. This will throw away all changes and reload the file with its original contents intact. To accomplish this, EditRand deletes the current file 1099.DAT and renames the backup file 1099.ERB to 1099.DAT.

### 3. NUMBERS

There are two categories of numeric quantities:

- a. ASCII, or text numbers. This is any number that is entered literally with the number keys, such as

9283, or 938267.23

For an example of this type number, look between byte positions 51 and 140. You can tell instantly that these are numbers because they look like numbers. ASCII numbers are easy to deal with because their representation is easily identifiable. However, the drawback is that they are more difficult for the computer to deal with because they slow calculation speed. The use of binary numbers sacrifices outward readability for performance gains.

- b. Binary numbers. Binary numbers are a representation that is closer to the way the computer actually looks at things, and therefore allow faster calculations and require fewer bytes of storage. For example, an ASCII number that is represented as 9204677821109228 requires 16 bytes, or characters. The same number represented as a binary number only requires 8 bytes of storage, but if viewed using EditRand, will appear as nonsense, such as:

↓<sup>2</sup>-Y♥▲+-

If the computer is asked to perform a calculation upon an ASCII number, it must first convert it to a binary number, then perform the calculation. If the number is already in binary form, then the computer can do its business much faster.

EditRand allows the use of 5 different types of binary numbers, each with its own meaning and length:

## EditRand Tutorial

---

Type	Length	Maximum Value
Integer	2 bytes	32,768
Long Integer	4 bytes	2,147,483,648
Single Precision	4 bytes	2.1 X 10 <sup>9</sup> (in theory)
Double Precision	8 bytes	9.2 X 10 <sup>18</sup> (in theory)
Currency	8 bytes	9.2 X 10 <sup>18</sup> (in theory)

In theory for Single Precision, Double Precision, and Currency is indicative of the fact that anytime you are dealing with a floating point number, it is possible for inaccuracies to eventually find their way into calculations on such numbers (such as add, subtract, multiply, or divide). The theoretical limit means that the number should be able to accurately perform calculations up to that limit, but is not guaranteed. This is not a problem for EditRand, as EditRand does not perform such calculations.

How a numeric field in a data file is represented is up to the designer of the data file. If speed is not an issue, then it might be preferable to use pure ASCII representation. If speed and compactness are issues, then binary representation should be used. On the other hand, some database packages don't give you a choice, and may force one or the other, or even use a proprietary format that is neither ASCII or binary.

- c. EditRand will allow you to enter any binary number anywhere in any record. In Edit mode, press F2. You will see the following prompt:

Integer, Long, Single, Double, or Currency ?

Press the indicated highlighted key to enter the desired numeric type. EditRand will then allow you to enter the number in ASCII (decimal) format. When you press Enter, the ASCII number is converted to binary and inserted in the current record at the current cursor position.

## EditRand Tutorial

---

- d. Verify the number you just inserted. If you entered an integer, press F3 with the cursor at the same position. You should see, on the status line, the same number in ASCII format. Other binary numeric display keys are:

F4 Long Integer  
F5 Single Precision  
F6 Double Precision  
F7 Currency

The use of these keys to show the ASCII representation of the binary number at the current cursor position depends upon you, the user, to know where any particular binary number begins. It is possible to use these keys on ANY portion of a record, even a pure text field such as Name or Address. EditRand will not make the distinction and will attempt to show you what the numeric value at the current cursor position would be if it was a binary number.

- e. Note that the use of the 5 binary numeric types can be brought into play when performing other EditRand operations, such as Search, Replace, and Wordcount. Each of these functions has an option (F2-Bin) to use binary numbers instead of ASCII text.

4. The FILE SELECTION SCREEN

If you are in Edit mode, press the Escape key to exit to Browse mode. (You are in Edit Mode if you can see a blinking cursor anywhere in the record area of the screen.) Now press L and you will be presented with the following prompt:

File name [length]:

You could enter a filename and press Enter to load it, or you could invoke the File Selection Screen. There are two ways to do this:

- a. Enter a file specification that uses the wildcard characters ? and \*, and press Enter. The File Selection Screen will appear with all files that match the specification showing in a window. For example, if you typed \*.DAT, this would cause all files with the extension "DAT" to be collected or found by EditRand, and displayed in the File Selection Screen.
- b. Press the F8 function key. The first time you do this, EditRand will supply the file specification \*.\* which stands for ALL files in the current subdirectory. After the first time, EditRand will use whatever path and file specification you previously used. This prevents you from having to retype the same specification again. To override this, just type a new specification.

For this exercise, just press the F8 key so all files in the current subdirectory will be displayed. If you have not changed subdirectories, the files 1099.DAT and 1099.DBF should be among the displayed files. Now to try out some of the functions of the File Selection Screen:



c. The main window of the File Selection Screen contains the names of the files, in alphabetical order, that have been found for display. Subdirectories will also appear in this window, including the current (shown as a single dot) and the subdirectory above the current, if there is one (shown as two dots.) If there are too many to show on the screen, you can cause them to scroll onto the screen with the up and down cursor keys, or the Page Up and Page Down keys. The file that is highlighted is the one that will be loaded if you press Enter.

d. You can abort the File Selection Screen without loading a new file, by pressing the Escape key. Also, you can type a new file specification (File mask) without leaving the File Selection Screen. Press the Tab key once and the cursor will appear on the field marked "File Mask:". Using the backspace key, change \*.\* to 1099.\* and press enter three times. Now the only files you should see are:

```
1099.DAT
1099.DBF
1099.ERB
```

1099.ERB is the backup file created by EditRand. EditRand will not allow you to load this file, as it is in use.

e. Highlight the file 1099.DBF using the cursor keys. Now press the Tab key twice to put the cursor on the "Switches:" field. Type /DT and press Enter three times. This will cause the file 1099.DBF to be loaded as a dBASE III file. Answer Yes when asked:

```
Lose changes and continue ?
```

## EditRand Tutorial

---

This means don't save the changes you just made to 1099.DAT and go ahead and load 1099.DBF. The switch /DT was necessary so EditRand would load the file as a dBASE III file rather than an ordinary flat data file. This special handling includes:

1. Use the dBASE III file header length as the EditRand header offset (see the /Onnn switch in the manual.) Notice that line two now has an additional item, H: nnn. This is indicative of the fact that a Header offset is in effect, and the nnn is the actual length, in bytes, of the offset. This means that you will not see the first nnn bytes of the file. The offset in this case (since the /DT switch was used), is the dBASE III header length. EditRand found this length by looking at the specific location at the beginning of the file that contains the header length.
  2. Use the dBASE III record length as the EditRand record length. This also means that any record length shown in the "Record length:" field on the File Selection Screen will be ignored in this case.
  3. The T part of the /DT switch is optional. When used, EditRand will read the dBASE III field names and reset tabs to these fields. This is particularly useful if you are using Dynamic Field Boundaries. Another benefit is that the name of the current field and its type definition is always displayed on the status line.
- g. You can now experiment with some of the same exercises you went through previously with the file 1099.DAT. Take note that 1099.DBF has the same basic information as 1099.DAT, but the fields are not in the same order.

5. SAVE AS and Record Deletions

Load the file 1099.DAT, press E for Edit mode, and do the following.

- a. Press Alt-V. This will allow you save the current file to a different file name - that is, to make a copy of it. You will see the prompt:

Save As:

Enter the name TEMP.DAT or any other name you wish, and press Enter. Press N when asked if you wish to add carriage return/line feeds. The current file, 1099.DAT, is written to the file TEMP.DAT. TEMP.DAT now has the exact same contents as 1099.DAT. You will use TEMP.DAT in a later operation.

- b. Press Alt-D to delete the current record (should be record 1.) The current record will become all blanks except byte 1 will be the Greek letter theta ( $\phi$ ). This is how any deleted record will look.

- c. EditRand will not allow any changes to a deleted record. Try entering text into the record. Notice that you can still move the cursor through the deleted record, but no changes can be made. It is still possible to make changes to any other, non-deleted record.

- d. Press F10 and reset record bounds to "21 40". It is not necessary to change Field Boundaries at this time. Now press Ctrl-D to delete records 21 through 40. When you see the prompt:

Delete records 21 through 40 ?

press Yes and all 20 records will be deleted.

- e. Press Escape to exit to Browse mode. Press L and specify \* (to reload the current file). When asked

## EditRand Tutorial

---

Save changes and continue (Y/N) ?

type Y. When asked

Purge deleted records (Y/N/Escape) ?

type Y. You will now see EditRand read each record in the current file, check it for deleted status, and then write the record to the new version of 1099.DAT if it is not deleted. On the bottom line of the display you will see two counts as EditRand goes through this process: one for total records checked in the current file, and one for the total number of non-deleted records being written out to the new version of 1099.DAT. When this process is complete (it shouldn't take more than a few seconds on a file this small), 1099.DAT will be reloaded. Notice that there are now only 59 records - since you just deleted 21 records.

- f. Now we will restore 1099.DAT to its original state using the copy you made, TEMP.DAT. Press L and specify TEMP.DAT to be loaded. Press E for Edit mode and press Alt-V again and specify 1099.DAT as the Save As file. When you see the prompt:

File already exists, erase it and write over (Y/N) ?

type Y and the smaller version of 1099.DAT will be overwritten with the full version (in the form of TEMP.DAT).

- h. Load 1099.DAT and verify that it contains 80 records.

You may experiment with 1099.DAT and 1099.DBF in any way you wish. The operations you just performed were intended to give you a general feel for EditRand procedures and to give specific experience with some of the more useful functions.

