



**IBM XEDIT Column Editing/VM
(XCOL)
User's Guide**

Release 1.01

First Edition (July 1991)

This Edition applies to Release 1.0 version of IBM XEDIT Column Editing/VM (Product number: 5799-EGZ).

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to

IBM Corporation
Internally Developed Solutions Center
Department 827
1701 North Street
Endicott, NY 13760

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1991. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Preface

IBM XEDIT Column Editing/VM (XCOL) is a column editing product designed to enhance XEDIT.

COL is the XEDIT COLUMn editing macro contained within the XCOL product. The COL macro allows you, an XEDIT user, to manipulate columns in the same way that you can already manipulate lines using XEDIT's prefix commands. The XCOL product and its COL macro are fully compatible with XEDIT commands and support XEDIT's exclude functions as provided with the XEDIT **X** prefix command and the ALL command. Any width file may be edited with XCOL.

The functions available within the COL macro are:

- Basic column operations
 - Delete
 - Move
 - Copy
 - Add
- Column operations unique to COL
 - Insert text
 - Overlay text
 - Convert to upper or lower case
 - Copy text forward or backward within a column
 - Blank redundant text within a column in a forward or backward direction.
 - Place sequence numbers or characters in a column
- Mathematical operations and REXX interface
 - Sum a column of numbers with optional line totals and subtotals
 - Perform mathematical operations on column data
- Sorting
 - Sort columns
 - Sort lines while honoring XEDIT excluded lines
- Text alignment
 - Align text
 - Flow words within a column
 - Justify text including left, center, right and both
 - Automatically place text in columns for editing
 - Compress column data into script tabular data
 - Remove trailing blanks
- Convenience commands
 - Undo the Last COL Operation
 - Identify the location of columns
 - Set tabs
 - Set verify setting
 - Set truncation column
 - Set zones
- Advanced functions
 - Exclude specific columns from XEDIT prefix commands and subcommands
 - Apply XEDIT prefix commands and subcommands to lines containing specified text within a column
 - Block or group text so that only the blocked areas are displayed.
 - Transpose columns and lines

Contents

Preface	iii
Notices	xi
Introduction	xiii
Who should read this document	xiii
What this document covers	xiii
Part 1. The COL Macro	1
Chapter 1. Getting Started	3
COL Screen	3
COL Profile	4
Invoking COL	6
Using the XEDIT Command Line	6
Using XEDIT Prefix Commands	6
Exiting COL	9
COL Command Line Wrapping	9
XEDIT / COL Maximum Line Width	10
HELP Function Keys	10
Chapter 2. Using Basic COL Functions	13
Familiar XEDIT Prefix Functions	13
Delete	13
Add	15
Insert	16
Targets for Copy, Move and Transfer (Xfer)	17
Copy	17
Move	18
Duplicate	19
Column Functions	19
Upon Target	19
Transfer (Xfer)	20
Replace	21
Overlay	22
Pad	23
Replace Characters With Blanks	24
Lower Case	27
Upper Case	28
Emulate or Copy Data Forward	29
De-emulate or Blank Redundant Data	30
Reverse Column Text	31
Number a Column	32
Sorting	34
Sorting Lines of Text	34
Column Sorts	35
Single Column Sorts	35
COL Command Line Options	36
Current Column (/)	36
Column One Operations	37
Using an Asterisk (*) in the COL Command Line	38

Command Options Using the	38
Reverse the Execution of a COL Command	38
COL Prefix Functions	38
Undo the Last COL Command	38
Picture Columns	39
Re-display the Last COL Command	40
Re-execute the Last COL Command	40
Reset the COL Command Line	40
Set the COL Command Line Symbol	41
Set the COL Blank Character Symbol	41
Reverse the Line Order	41
Transpose Columns and Lines	43
Store COL Commands	44
XEDIT Commands in the COL Prefix Area	44
Convenience Commands	44
Chapter 3. Using COL Mathematical and Logic Functions	47
Sum Function	47
Sum Numbers in a Column	47
Mathematical Operations and REXX Interface	54
Three Ways of Entering Your Expression	56
Mathematical Operators (REXX)	62
Chapter 4. Using COL Text Functions	67
Text Alignment	67
Queue or Align Specified Text	67
Using Queue Within Specific Columns	70
Removing Trailing Blanks	71
Removing Trailing Blanks Within Specific Columns	72
Text Justification	73
Spacing Between Words When Using Justify Commands	79
Automatic Text Alignment	80
Justifying Text Using Tab Settings	83
Paragraph Word Flow	84
Left Justify and Flow the Text Within a Column	84
Right Justify and Flow the Text Within a Column	85
Left and Right Justify and Flow the Text Within a Column.	86
Compressing Text Into a Narrower Column	87
Editing Blocks of Text	87
COLALL XEDIT	91
Chapter 5. XEDIT Commands Within COL	95
Applying XEDIT Commands To Columns	95
Column Include (Y)	95
Column Exclude (-)	96
XEDIT Commands Within a Column	97
Applying XEDIT Commands To Lines with Specified Text in a Column	100
XEDIT Prefix Commands	100
XEDIT Commands	108
Offsetting XEDIT Prefix and Subcommands	109
Delayed Execution of XEDIT Prefix Block Commands	110
Arbitrary Character	112
Using the * Operator to Search All Fields.	112

Part 2. Applications Using the COL macro	115
Adding Blank Separator Lines	115
Adding Lines With Specific Text	117
Inserting Lines Containing ".cp"	117
Calculating Percent of Total	119
Create a Macro To Calculate Percentages	122
Comparing Columns	123
Converting To and From GML or BookMaster List Items	125
Counting Items in a List	127
Creating an XEDIT Macro Using COL to Count Items in Lists	128
Alternate Macro that Stacks the COL Commands to Count Items in Lists	128
Creating a GML/BookMaster Definition List From Column Data	129
Creating Column Data from Non-Column Text	133
Filling in Blanks or Removing Redundant Text	134
Flowing Text Within a Column	135
Editing GML or BookMaster Table Columns	136
Marking All Entries With a Value within Ranges.	137
Making One Column From Two	138
Making Two Columns From One	140
Marking Duplicate Text in a File	143
Moving and Inserting Column Text	145
Rearranging Blocks of Text Such as ":H1." Sections	146
Reformatting Tables	148
Removing Blank Lines From a File	151
Removing Duplicate Text From a File	152
Sorting Blocks of Text Such as :H1. Sections	153
Sorting Columns	158
Sorting Dates	159
Sorting Lists	160
Sorting Text With Imbedded Headers	166
Transferring Column Text With PUT and GET	168
Working With Columns Independently	171
COL Macro	173
PF Keys in COL	175
Stacking Commands for COL	175
XEDIT Interactions	176
Command Priority	176
XEDIT Ring	176
XEDIT Multiple Logical Screens	176
Moving the Range Markers	176
COL Messages	177
Quick Reference	179
Glossary	181
Index	183

Figures

1.	COL Screen	3
2.	Using PREFIXH	8
3.	Using PREFIXH2	8
4.	COL Help Screen	11
5.	COLALL	91
6.	COL Command	173
7.	Quick Reference COL Commands	179
8.	Mathematical Operations	179
9.	Quick Reference COL Prefix Commands	180

Tables

1.	COLALL Predefined Blocks of Text	92
2.	COL Prefix Operators When Applying XEDIT Commands	101

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program, product, or service that does not infringe on any of IBM's intellectual property rights or other legally protectible rights may be used instead of an IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

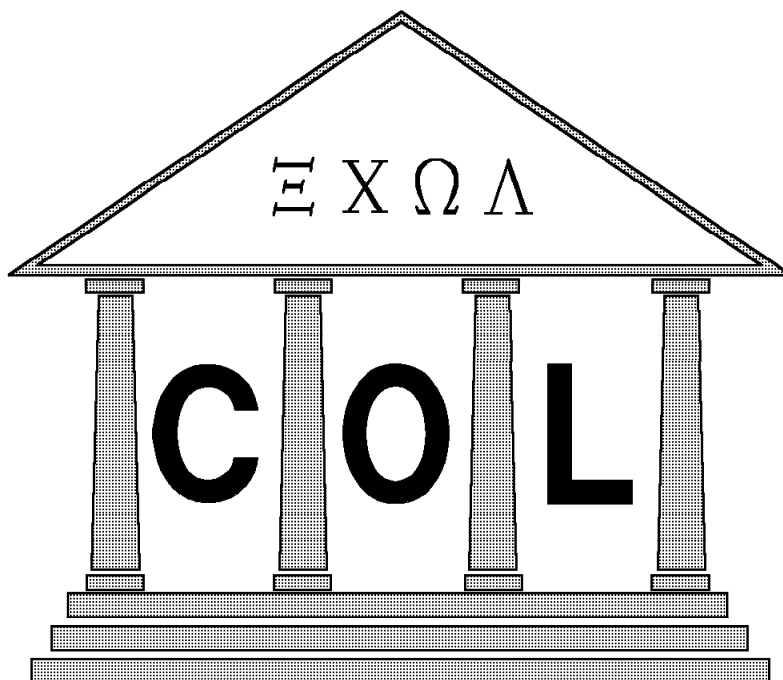
Introduction

Who should read this document

Anyone interested in learning how to use XCOL to simplify column, data and/or document editing using IBM's VM System Product Editor (XEDIT). The user should be familiar with XEDIT and may want to have the XEDIT manual (IBM publication number SC23-0372) available for reference.

What this document covers

This document describes how to use IBM XEDIT Column Editing/VM (XCOL) and its macro COL.



Chapter 1. Getting Started

Before you invoke COL for the first time, an explanation of what you will see is in order. The first sections of this manual will describe the layout of the COL screen, and the methods you can use to customize the screen appearance. This will be followed by the actual procedure for entering and exiting a COL session.

COL Screen

Once you have invoked COL within an XEDIT session, your XEDIT screen will appear as follows:

```
4 Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ov1,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

===== * * * Top of File * * *
===== Subject: Hotel reservations
===== The following employees will be attending the conference.
3 <<-> Please make the necessary hotel reservations for the indicated dates.
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
2 ===== ?????????????????????????????????????????????????????????????
===== Sirr, Lundy 3650 9D5 06/07/89
===== Smith, Ken 6534 D43 IBMVM3 06/08/89
===== Sparks, Karen 3253 D43 IBMVM3 06/09/89
===== Stouffer, Bob 7655 D43 IBMVM3 06/13/89
===== Sweeney, John 7010 D49 IBMVM1 06/14/89
===== Trahern, Cynthia 3866 D41 IBMVM1 06/15/89
3 <<-> Please confirm the reservations directly with the employees.
===== Our next conference will be in 6 months.
=====> 5

<<-> pending...
```

Figure 1. COL Screen

- 1 **COL command line..** Commands entered into this command line are usually not case sensitive.
- 2 **COL prefix area.** The prefix area of the COL command line is used to issue COL commands.
- 3 **COL range markers.** The text within these markers is the text being operated on. While in COL you will be able to scroll within this area, but not outside. XEDIT's TOP or BOTTOM will take you to the top or bottom of the COL range.
- 4 **COL information area.** This area is provided as a key for COL commands. This area will be at the bottom of your screen if you have placed your XEDIT command line at the top of the screen.
Note: In the illustrations that follow, commands being utilized within that example are high-lighted within the information area.
- 5 **XEDIT Command Line**

COL Profile

A number of customizable options for COL are provided. If you want to change any of these options from the default provided, XCOL will look for your options selections in the optional COL profile named PROFCOL XEDIT. A sample profile is provided with the XCOL product and is named PROFCOL SAMPLE.

You may copy this sample profile to your A disk, customize it and rename it to PROFCOL XEDIT. The following options may be altered:

```
SYMBOL      = ?/? /* The COL COMMAND LINE symbol */
BLANK_CHAR  = / /* The blank character symbol */
PREFIXSYMBOL = <-> /* The prefix range marker */
UNDO_OPTION = ON/* Activate the undo option */
TARGET      = /* default target (depth) */
AUTO_EXIT   = OFF/* Exit COL when no screen
/* changes have been made and
/* ENTER is pressed
TAB_SPACE   = 2 /* Column spacing for ATAB, the
/* auto tab command
DIGITS      = 12 /* REXX numeric digits setting
SENTENCE_SPACE = 2 /* spacing after sentences for
/* text flow commands W, < and >
MATH_MSG    = ON/* math error messages
DECIMAL     = PERIOD/* decimal indicator for ++ cmd. */
```

You may alter the numeric format that COL expects to use for summing columns by entering one of the following options for DECIMAL.

```
/* The three options for decimal are:
/* PERIOD for 1,234,567.89 (The Default)
/* COMMA for 1.234.567,89
/* FRENCH for 1 234 567,89 or 1 234 567.89
```

You may alter the color and location of the COL information area as follows:

```
/* Colors */
INFOCOLOR   = RED/* tHelp text
PFCOLOR     = WHITE/* PF key text

SET COLOR MSGLINE RED /* Note that COL uses XEDIT MSG
/* for messages and the current
/* line for the COL command line
/* Set XEDITs
/* COLOR MSGLINE
/* COLOR CURLINE
/* to control these colors
```

You may relocate the position of the mini help information area from the default location by specifying a new value for INFOLOC. Set INFOLOC to OFF to remove all COL messages.

You may also specify OFF whenever you want to set up your own reserved lines.

```
/* Examples:
/*
INFOLOC      = 1 /* One line from top
INFOLOC      = -3 /* Three lines from bottom
INFOLOC      = OFF/* Do not display
```

You can also add XEDIT options to your profile as follows:

```
SET SCALE OFF          /* Option to turn off the scale */
SET NULLS ON           /* Option for nulls on          */
SET FULLREAD ON        /* Option for fullread on       */
```

PROFCOL SAMPLE as supplied with the XCOL product has the SET COLOR and INFOLOC options commented out. You may activate them or you may add any additional options you may require.

Invoking COL

COL may be invoked from within an XEDIT session in two ways.

- From the XEDIT Command line
- From the XEDIT Prefix area

To exit COL you may press either PF3 or PF15.

Using the XEDIT Command Line

From within an XEDIT file, COL can be invoked using the XEDIT command line as shown in the example below:

```
====> COL target
```

```
X E D I T  1 File
```

COL will now operate on the file starting with the XEDIT current line and ending at *target*, where *target* is any standard XEDIT target such as the following:

```
COL 5           start with the current line for a depth of 5 lines
COL :50         down to, but not including line 50
COL /text/      down to, but not including the "text"
COL *          down to the end of the file
```

Using XEDIT Prefix Commands

To use COL from the XEDIT prefix area you have to set up XEDIT Prefix Synonyms in your XEDIT Profile.

You can set up your prefix synonyms to either keep the current line fixed and move the column or keep the column fixed and move the current line.

1. Move the column just below your current line.

You may prefer to set up COL in this manner if your XEDIT current line is located in the middle of the screen. You can bring the first line of the specified COLUMNS depth to the current line location by entering the following commands in your XEDIT Profile:

```
SET PRE SYNONYM H  PREFIXH
SET PRE SYNONYM HH PREFIXH
```

2. Move the COL command line just above your column.

This set up for COL may be preferable if your XEDIT current line is located at the top of your screen. If you want to keep the text on the screen in place and have the COL command line placed at the top of the column then set up prefix synonyms as follows in your XEDIT Profile:

```
SET PRE SYNONYM H  PREFIXH2
SET PRE SYNONYM HH PREFIXH2
```

Whichever synonym you use, you may then call COL from your XEDIT prefix area and specify the depth of the column to be operated on by entering any of the following in the prefix area:

```
H      one line
Hnn    nn lines
nnH     ↑
H*      from here to the end of the file
```

```
or HH   (from here
.       .
.       .
.       .
HH      to here)
```

Now, for example, when you enter **hh** as shown below:

```
COL      SCREEN  A1  V 80  Trunc=80 Size=29 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== Subject: Hotel reservations
===== The following employees will be attending the conference.
===== Please make the necessary hotel reservations for the indicated dates.
hh==== Schultz, Sue                6664 D25 IBMVM3    06/03/89
===== Schurr, Tomoko              5234 D41 IBMVM3    06/04/89
===== Schwind, Sandy              3881 D40 IBMVM3    06/05/89
===== Sekiguchi, Barbara          7650 D46 IBMVM1    06/06/89
===== Sirr, Lundy                 3650 9D5           06/07/89
===== Smith, Ken                  6534 D43 IBMVM3    06/08/89
===== Sparks, Karen               3253 D43 IBMVM3    06/09/89
===== Stana, Sandi                7084 D44 IBMVM1    06/10/89
===== Stevenson, Bob              7346 D49 IBMVM1    06/11/89
===== Stine, Ray                  7895 D43 IBMVM3    06/12/89
===== Stouffer, Bob               7655 D43 IBMVM3    06/13/89
hh==== Sweeney, John               7010 D49 IBMVM1    06/14/89
===== Please confirm the reservations directly with the employees.
===== Our next conference will be in 6 months.
===== * * * End of File * * *

=====>

XEDIT 1 FILE
```

The results for PREFIXH are that the top of the column to be worked on is now located just under your XEDIT current line.

The XEDIT current line is now your COL command line as follows:

```

Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

===== * * * Top of File * * *
===== Subject: Hotel reservations
===== The following employees will be attending the conference.
<<->> Please make the necessary hotel reservations for the indicated dates.
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== Schultz, Sue                6664 D25 IBMVM3      06/03/89
===== Schurr, Tomoko              5234 D41 IBMVM3      06/04/89
===== Schwind, Sandy              3881 D40 IBMVM3      06/05/89
===== Sekiguchi, Barbara          7650 D46 IBMVM1      06/06/89
===== Sirr, Lundy                 3650 9D5             06/07/89
===== Smith, Ken                  6534 D43 IBMVM3      06/08/89
===== Sparks, Karen               3253 D43 IBMVM3      06/09/89
=====
<<->> pending...
```

Figure 2. Using PREFIXH

The results for PREFIXH2 are that the current line moves to the top of the column and becomes the COL command line as follows:

```

Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
===== The following employees will be attending the conference.
<<->> Please make the necessary hotel reservations for the indicated dates.
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== Schultz, Sue                6664 D25 IBMVM3      06/03/89
===== Schurr, Tomoko              5234 D41 IBMVM3      06/04/89
===== Schwind, Sandy              3881 D40 IBMVM3      06/05/89
===== Sekiguchi, Barbara          7650 D46 IBMVM1      06/06/89
===== Sirr, Lundy                 3650 9D5             06/07/89
===== Smith, Ken                  6534 D43 IBMVM3      06/08/89
===== Sparks, Karen               3253 D43 IBMVM3      06/09/89
===== Stana, Sandi                7084 D44 IBMVM1      06/10/89
===== Stevenson, Bob              7346 D49 IBMVM1      06/11/89
===== Stine, Ray                  7895 D43 IBMVM3      06/12/89
===== Stouffer, Bob               7655 D43 IBMVM3      06/13/89
===== Sweeney, John               7010 D49 IBMVM1      06/14/89
<<->> Please confirm the reservations directly with the employees.
=====
<<->> pending...
```

Figure 3. Using PREFIXH2

If you want the option of using the capabilities of both PREFIXH and PREFIXH2, then set up two prefix synonyms in your XEDIT Profile as follows:

```
SET PRE SYNONYM H PREFIXH
SET PRE SYNONYM HH PREFIXH
SET PRE SYNONYM V PREFIXH2
SET PRE SYNONYM VV PREFIXH2
```

Now you may use either **hh** to move the column to the current line or **vv** to move the command line. The rest of this manual will assume you have set your synonym table this way.

Exiting COL

You can exit COL in any of four ways:

1. Press a function key.

Press PF 3 function key or the PF 15 function key

The COL information area displays this as a reminder:

Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y

2. Enter the ECOL command in the XEDIT command line.

====> ECOL	X E D I T 1 File
------------	------------------

3. Enter the ECOL command in the COL prefix area.

ECOL	...+....1....+....2....+....3....+....4....+....5....+....6....+....7... ??
------	--

4. If you have set AUTOEXIT = 'ON' in PROFCOL XEDIT, then you may also exit by pressing ENTER without making any screen alterations, or by pressing ENTER twice after an alteration.

Note: When AUTOEXIT = 'OFF', the default, pressing ENTER will move the cursor to the COL command line, retaining its column position, and will not exit you from COL.

COL Command Line Wrapping

The COL command line is where you will enter your COL commands. This line is normally displayed as a string of question marks, unless you have changed your default setting in PROFCOL XEDIT to a different symbol. (see page 3, the value for symbol)

Depending on your XEDIT setting for **Verify** the COL command line may wrap as follows:

====	...+....1....+....2....+....3....+....4....+....5....+....6....+....7... ?? ??
------	--

This setup is usually bothersome while editing columns. Therefore, the command line wrap may be eliminated by entering the following on XEDIT's command line or by adding it to your XEDIT Profile or to your PROFCOL XEDIT:

```
====> SET VERIFY 1 73
```

X E D I T 1 File

Also, you may use the COL command **V** (not to be confused with the synonym "V" command used to invoke COL) to change XEDIT's Verify setting as follows.

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== v????????????????????????????????????????????????????????v?
????????????????????????????????????????????????????????

```

The result of any of the above will be that your COL command line occupies one line as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????

```

XEDIT / COL Maximum Line Width

You may need to alter XEDIT's record length when using COL to insert columns into a file. XEDIT defaults to a predefined record length based on file type or to a record length equal to the longest record within the file, whichever is greater. If the record length is equal to the longest record you cannot insert new columns without truncation.

To obtain additional column width, you must invoke the **WIDTH** option when first calling XEDIT:

```
XEDIT fn ft fm (WIDTH nnn
```

where **nnn** is the new width of the file including your blank columns for work space.

Once in XEDIT you may need to alter the following XEDIT settings before using COL.

```
SET LRECL nnn
SET ZONE 1 *
SET TRUNC *
```

where **nnn** matches the number specified in the above **WIDTH** option.

HELP Function Keys

The following PF help keys are provided in COL.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

PF 1: You may press PF1 to obtain detailed help.

```
COL TASK      Task Help Information      line 1 of 16
              XCOL COLUMN Edit Macro for XEDIT

Place the cursor under any character of the task you are interested in and
press PF1 or the ENTER key.

General Information
Getting Started
Screen Layout
PF Key Settings
Command Summary
Detailed Command Information
Prefix Command Information
Mathematical Operations
Applying XEDIT Commands
What Is New?
COL Profile (optional)
Macro and Usage Notes

PF1= Help      2= Top      3= Quit      4= Return      5= Clocate      6= ?
PF7= Backward  8= Forward  9= PFkeys  10=          11=          12= Cursor

====>

Macro-read 1 File
```

Figure 4. COL Help Screen

For a command listing by function with examples, select the “Detailed Command Information” line.

PF 2 You may obtain a summary of the COL commands by pressing PF2. See “Quick Reference” on page 179 for these screens.

Chapter 2. Using Basic COL Functions

Familiar XEDIT Prefix Functions

Several of the COL commands provide the same function for column editing as normal XEDIT Prefix commands provide for line editing. The following commands belong in this category: Delete, Add, Insert, Following, Preceding, Copy, Move and Duplicate.

Delete

The delete command is used as shown in the example below:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral           7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin          6150 9D5             06/02/89
===== Schultz, Sue              6664 D25 IBMVM3       06/01/89
===== Schurr, Tomoko            5234 D41 IBMVM3       06/01/89
===== Schwind, Sandy            3881 D40 IBMVM3       06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral           7249 06/01/89
===== Schuessler, Marvin          6150 06/02/89
===== Schultz, Sue              6664 06/01/89
===== Schurr, Tomoko            5234 06/01/89
===== Schwind, Sandy            3881 06/02/89
<<->> * * * End of File * * *
```

Possible combinations for the delete command are:

D Deletes the column where D is placed.

DD DD Deletes a block of columns starting at the position of the
or DDDDDD first D up to the last D.

or Dcolumn_target
where column_target may be any XEDIT CLOCATE target.

Valid Examples: D* Delete to end of line
 D-* Delete to beginning of line
 D9 Delete 9 columns
 D:20 Delete to column :20
 D/the/ Delete to the word "the"
 D-/the/ Delete back to the word "the"

Add

This is the same as the XEDIT Prefix commands Add and Insert. Blank columns are added to the right of the **A**. When you place a number after the **A**, then that number of blank columns will be added.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
PF1=Hlp/2=CMDlist/3=Exit <=W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *  
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...  
===== ?????????????????????????????????a5????????????????????????????  
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89  
===== Schuessler, Marvin         6150 9D5             06/02/89  
===== Schultz, Sue             6664 D25 IBMVM3      06/01/89  
===== Schurr, Tomoko           5234 D41 IBMVM3      06/01/89  
===== Schwind, Sandy           3881 D40 IBMVM3      06/02/89  
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *  
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...  
===== ?????????????????????????????????????????????????????????????  
===== Savickas, Derral          7249      D49 IBMVM1      06/01/89  
===== Schuessler, Marvin         6150      9D5             06/02/89  
===== Schultz, Sue             6664      D25 IBMVM3      06/01/89  
===== Schurr, Tomoko           5234      D41 IBMVM3      06/01/89  
===== Schwind, Sandy           3881      D40 IBMVM3      06/02/89  
<<->> * * * End of File * * *
```

A Adds 1 blank column after the column where the A is positioned.

A5 Adds 5 blank columns.

AAA Adds a blank column after each of the 3 columns.

If you need to add blanks in column 1, please refer to “Column One Operations” on page 37 for information on this procedure.

Should you encounter truncation problems see “XEDIT / COL Maximum Line Width” on page 10.

Insert

The insert command in COL differs from the XEDIT Prefix insert command, but functions like the XEDIT command line insert command. When you place text behind the **I**, that text is then inserted into a column exactly where the text is located in the COL command line.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral           7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin          6150 9D5             06/02/89
===== Schultz, Sue              6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko            5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy            3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral           Ext. 7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin          Ext. 6150 9D5             06/02/89
===== Schultz, Sue              Ext. 6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko            Ext. 5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy            Ext. 3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

If you need to insert text into column 1, please refer to “Column One Operations” on page 37 for information on this procedure.

Should you encounter truncation problems see “XEDIT / COL Maximum Line Width” on page 10.

Targets for Copy, Move and Transfer (Xfer)

Also see “Upon Target” on page 19 for an additional target that places the text upon a column rather than making room for the text as do the Following and Preceding commands.

Following The column following the **F** will be the position into which the column will be inserted.

Preceding The column preceding the **P** will be the position into which the column will be inserted.

Copy

You may copy an existing column into another column as follows.

Note: The **f** indicates the column will be copied to the column following or to the right of the **f** command.

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????f????????????????????cccccccc????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5           06/02/89
===== Schultz, Sue             6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy           3881 D40 IBMVM3      06/02/89
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral          06/01/89      7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         06/02/89      6150 9D5           06/02/89
===== Schultz, Sue             06/01/89      6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko           06/01/89      5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy           06/02/89      3881 D40 IBMVM3      06/02/89
<<-> * * * End of File * * *
```

C Copies 1 column to the prefix target.

C5 Copies 5 columns to the prefix target.

CC CC Copies the block of columns starting at the position of the
or CCCCCC first C up to the position of the last C to the prefix
 target.

Should you encounter truncation problems see “XEDIT / COL Maximum Line Width” on page 10.

Move

You may move an existing column to another column location using the move command. Just as when moving lines, the space which contained the text will close. Also see "Transfer (Xfer)" on page 20 for the command which does not close the vacated column, but rather leaves blanks.

Note: The **p** indicates the column will be copied to the column preceding or to the left of the **p** command.

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== p????????????????????????????????????????????????????????????mmmmmmmm????????????????
===== Savickas, Derral                7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin                6150 9D5           06/02/89
===== Schultz, Sue                   6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko                 5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy                 3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????????????????????????????
===== 06/01/89 Savickas, Derral        7249 D49 IBMVM1
===== 06/02/89 Schuessler, Marvin       6150 9D5
===== 06/01/89 Schultz, Sue           6664 D25 IBMVM3
===== 06/01/89 Schurr, Tomoko         5234 D41 IBMVM3
===== 06/02/89 Schwind, Sandy         3881 D40 IBMVM3
<<->> * * * End of File * * *
```

M Moves 1 column to the prefix target.

M5 Moves 5 columns to the prefix target.

MM MM Moves the block of columns starting at the position of the
or M M M M first M up to the position of the last M to the prefix
 target.

Duplicate

You may duplicate a column using the `"` command. The duplicated text is inserted into the column immediately following the command. A number may be placed behind the command to indicate how many copies of the duplicated data is required.

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone  
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ??????????????????????†††3?????????????????????????????????????  
===== Savickas, Derral      |      7249 D49 IBMVM1      06/01/89  
===== Schuessler, Marvin    |      6150 9D5              06/02/89  
===== Schultz, Sue        |      6664 D25 IBMVM3      06/01/89  
===== Schurr, Tomoko       |      5234 D41 IBMVM3      06/01/89  
===== Schwind, Sandy      |      3881 D40 IBMVM3      06/02/89  
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ??????????????????????†††3?????????????????????????????????????  
===== Savickas, Derral      |      |      |      |      7249 D49 IBMVM1      06/01/89  
===== Schuessler, Marvin    |      |      |      |      6150 9D5              06/02/89  
===== Schultz, Sue        |      |      |      |      6664 D25 IBMVM3      06/01/89  
===== Schurr, Tomoko       |      |      |      |      5234 D41 IBMVM3      06/01/89  
===== Schwind, Sandy      |      |      |      |      3881 D40 IBMVM3      06/02/89  
<<->> * * * End of File * * *
```

Should you encounter truncation problems see “XEDIT / COL Maximum Line Width” on page 10.

Column Functions

In addition to the previous commands which mimic XEDIT prefix commands, COL provides many functions which are unique to column editing.

Upon Target

The **Upon** target is in addition to the normal XEDIT prefix **F** and **P** targets. The column marked by the **U** will be the position on top of or upon which the column will be placed. Unlike **F** and **P** which make room for the column to be inserted, the **U** target will overlay whatever is already in that column. Normally a blank column is picked for this location.

Transfer (Xfer)

The **X** command is in addition to the commands Move and Copy. Transfer (X) will lift the column from the lines leaving blanks in its place. Move closes the vacated spot. Consider the following example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <=>W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????????u?????????????????????xxxxxx????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5             06/02/89
===== Schultz, Sue             6664 D25 IBMVM3       06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3       06/01/89
===== Schwind, Sandy           3881 D40 IBMVM3       06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????????u?????????????????????xxxxxx????????????????????
===== Savickas, Derral          IBMVM1    7249 D49      06/01/89
===== Schuessler, Marvin         6150 9D5             06/02/89
===== Schultz, Sue             IBMVM3    6664 D25      06/01/89
===== Schurr, Tomoko           IBMVM3    5234 D41      06/01/89
===== Schwind, Sandy           IBMVM3    3881 D40      06/02/89
<<->> * * * End of File * * *
```

X Transfers 1 column to the COL target.

X5 Transfers 5 columns to the COL target.

XX XX Transfers the block of columns starting at the position of
or XXXXXX the first X up to the position of the last X to the
COL target.

Replace

Replace is very similar to Insert because COL will place text in a column. However, unlike Insert, Replace will not “make room” by inserting the designated text, but rather replaces whatever text is already in that column. Note in the following example, the text is overlaid including the blank space following the period in “Dept.”

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5           06/02/89
===== Schultz, Sue             6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy           3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????
===== Savickas, Derral          Dept. D49 IBMVM1      06/01/89
===== Schuessler, Marvin         Dept. 9D5           06/02/89
===== Schultz, Sue             Dept. D25 IBMVM3      06/01/89
===== Schurr, Tomoko           Dept. D41 IBMVM3      06/01/89
===== Schwind, Sandy           Dept. D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

If you need to replace text in column 1, please refer to “Column One Operations” on page 37 for information on this procedure.

Overlay

Overlay is similar to Replace, except that any blanks in the text placed in the COL command line will not overlay text in the column. Overlay will perform as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????oA 4????????????????????????????
===== Savickas, Derral                7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin                6150 9D5           06/02/89
===== Schultz, Sue                    6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko                  5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy                  3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== Savickas, Derral                7249 A44 IBMVM1      06/01/89
===== Schuessler, Marvin                6150 AD4           06/02/89
===== Schultz, Sue                    6664 A24 IBMVM3      06/01/89
===== Schurr, Tomoko                  5234 A44 IBMVM3      06/01/89
<<->> * * * End of File * * *
```

Note: COL uses the XEDIT command CREPLACE for Replace and COVERLAY for Overlay. Please refer to CREPLACE and COVERLAY in the XEDIT/VM manual for more information on the differences between these two commands.

If you need to overlay text in column 1, please refer to "Column One Operations" on page 37 for information on this procedure.

Pad

The apostrophe (') command replaces blanks within a column with the specified text.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *  
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...  
===== ??????ç0????????????????????????????????????????????????????  
=====      8  
=====      9  
=====     10  
=====     11  
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *  
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...  
===== ??????0????????????????????????????????????????????????????  
=====      08  
=====      09  
=====     10  
=====     11  
<<->> * * * End of File * * *
```

If you need to pad text in column 1, please refer to “Column One Operations” on page 37 for information on this procedure.

Replace Characters With Blanks

The minus (-) command removes specified characters in columns and replaces them with blanks. The command normally searches on a column by column basis. This may be considered as the inverse operation of the Pad command. For example, to remove commas from specified columns you would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??-,,,,,,,,,,,,,????????????????????????????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5           06/02/89
===== Schultz, Sue             6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy           3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== Savickas Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler Marvin         6150 9D5           06/02/89
===== Schultz Sue             6664 D25 IBMVM3      06/01/89
===== Schurr Tomoko           5234 D41 IBMVM3      06/01/89
===== Schwind Sandy           3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

If you need to remove text in column 1, please refer to “Column One Operations” on page 37 for information on this procedure.

Blank Characters Searching from Left to Right: The minus command can also scan text from left to right blanking all matching characters. The blanking process is terminated when a non-match is found. This is accomplished by entering a leading blank in the text string that you enter behind the -. The text match test includes the blank character. XEDIT’s arbitrary character (see SET ARBCHAR section of XEDIT manual) is supported during this left to right text match.

For example, to remove zeros but not numbers that end in zero from specified columns you would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????- 0????????????????????????????????????????????
=====          0
=====          1
=====          10
=====          100
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
=====
=====          1
=====          10
=====          100
<<-> * * * End of File * * *
```

You could also remove leading zeros from a column as follows:

Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
 PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????- 0000????????????????????????????????????????????????
=====          00000
=====          00001
=====          00010
=====          00100
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
=====          0
=====          1
=====          10
=====          100
<<-> * * * End of File * * *
```

Lower Case

To convert characters in a column to lowercase you may place an **L** above each column to be converted. A block command of the form **LL????????LL** is supported to convert a block of columns to lower case. You may also place a number behind the **L** to convert the specified number of columns to lowercase. For example enter **L5** to convert 5 columns to lower case and **L*** to convert from the L to the end of the line to lowercase. For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?llllllllll????????????????????????????????????????????????????
===== SAVICKAS              7249 D49 IBMVM1      06/01/89
===== SCHUESSLER            6150 9D5           06/02/89
===== SCHULTZ               6664 D25 IBMVM3      06/01/89
===== SCHURR                5234 D41 IBMVM3      06/01/89
===== SCHWIND               3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== Savickas              7249 D49 IBMVM1      06/01/89
===== Schuessler            6150 9D5           06/02/89
===== Schultz               6664 D25 IBMVM3      06/01/89
===== Schurr                5234 D41 IBMVM3      06/01/89
===== Schwind               3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

Upper Case

The upper case command is similar to the lower case command except you enter **K** to convert to upper Kase or King size (Sorry, U has already been used for the "Upon" command). For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <=>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=>Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== Savickas          7249 D49 ibmvm1      06/01/89
===== Schuessler        6150 9D5           06/02/89
===== Schultz          6664 D25 ibmvm3      06/01/89
===== Schurr           5234 D41 ibmvm3      06/01/89
===== Schwind          3881 D40 ibmvm3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== Savickas          7249 D49 IBMVM1     06/01/89
===== Schuessler        6150 9D5           06/02/89
===== Schultz          6664 D25 IBMVM3     06/01/89
===== Schurr           5234 D41 IBMVM3     06/01/89
===== Schwind          3881 D40 IBMVM3     06/02/89
<<->> * * * End of File * * *
```

Emulate or Copy Data Forward

The **E** command is used to copy data forward within a column. This process creates groups which may be used for sorts among other things. Blank fields in the column specified by the **E**'s are replaced with the non-blank text string located above the blank field as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== eeeeeee????????????????????????????????????????????????????
===== 10/20/85    549
=====          456
=====          864
===== 10/21/85    549
=====          982
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????
===== 10/20/85    549
===== 10/20/85    456
===== 10/20/85    864
===== 10/21/85    549
===== 10/21/85    982
<<->> * * * End of File * * *
```

Emulate or Copy Data Up a Column The emulate command copies data down the column. To copy data UP the column, use the \ reverse operator preceding the command as follows:

```
| ===== ?????????????????????????????\eeeeeeeeee???????????????????????????? |
```

or the reverse operator may be entered following the command as shown below:

```
| ===== ?????????????????????????????eeeeeeeeee\???????????????????????????? |
```

Note that the \ is not part of the column width being operated on. Also, the emulate command can be designated as blocked data **ee????ee** or a specified length **e5**, as in other COL block commands.

De-emulate or Blank Redundant Data

This function defined by the underscore character `_` is the inverse operation of the **E** command. Any text string in the column identified by the string of `_`'s which is the same as the text string above it will be blanked as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== 10/20/85   549
===== 10/20/85   456
===== 10/20/85   864
===== 10/21/85   549
===== 10/21/85   982
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== 10/20/85   549
=====           456
=====           864
===== 10/21/85   549
=====           982
<<->> * * * End of File * * *
```

De-emulate or Blank Data Up a Column The de-emulate command operates from the top of the column down. This operation may be performed UP the column by using the `\` reverse operator preceding the command as follows: All text which is the same as the text below it will be blanked.

```
| ===== ??????????????????????????\_____ ?????????????????????????????????????? |
```

or the reverse operator may be entered following the command as shown below:

```
| ===== ??????????????????????????_____ \ ?????????????????????????????????????? |
```

Note that the `\` is not part of the column width being operated on. Also, the de-emulate command can be designated as blocked data `____?????` or a specified length `_5`, as in other COL block commands.

Reverse Column Text

You may enter a string of reverse slashes \ to reverse (swap end for end) text within a column.

Note: Using a single \ as a prefix or suffix operator with commands such as **E**, **_**, **N** or **+** causes that command to execute from the bottom up rather than from the top down.

Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y

```
<<->> * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== \\\\\\\\\\\????????????????????????????????????????????????????????
===== .          *.
=====  *.          *.
=====   *.          *.
=====    *.          *.
=====     *.          *.
<<->> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
=====
          .*.
=====
         .*  *.
=====
        .*    *.
=====
       .*      *.
=====
      .*        *.
=====
     .*          *.
<<-> * * * End of File * * *
```

Number a Column

This command may be used to place sequential values in a column. You may select any of the following to sequence your text:

- Arabic numbers (the default)
- Alpha characters
 - Uppercase
 - Lowercase
- Hex numbers

Whenever a number or character is found in the selected column then that number becomes the starting value for the remaining numbers. The number will not be allowed to “grow” outside the column width as specified by the number of **n**’s entered. Truncation occurs in the high order digits, causing the numbers to repeat.

You may add options behind the last **N** by placing a ‘(’ and then selecting one of the following:

Note: The **A** option is case sensitive.

Option

- | | |
|----------|---|
| A | Alpha Sequence (upper case) |
| a | Alpha Sequence (lower case, if XEDIT CASE setting is MIXED) |
| X | Hexadecimal |

You may alter the sequencing by entering a increment value and then entering a repeat factor if required. For example:

Option

- | | |
|-------------|--|
| (2 | Increment each number or alpha value by two. |
| (.1 | Increment each number by .1. |
| (-1 | Decrement each number by one. |
| (1,3 | Increment each number by one, but repeat once 3 is reached. |
| (,3 | Same as above. |
| (a2 | Increment each character by two. |
| (a,3 | Increment each character by one, but repeat once the third character (c) is reached. |

Note: Numbers must be separated by a comma.

For example, consider the following commands operating on different columns.

```

Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y

<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== n?nmn??n??nmn(.1??n(a??n(A?nmn(A?n(A-1??n(A2????????????????????
=====   9   9   0.0               X               E
=====
=====
=====   5
=====
=====
<<-> * * * End of File * * *

```

The results for each column are:

```

<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== 1   9   9   0.0   a   A   X   Z   A   E           1
===== 2  10   0   0.1   b   B   Y   Y   C   F           2
===== 3  11   1   0.2   c   C   Z   X   E  10           3
===== 4   5   2   0.3   d   D  AA   W   G  11           1
===== 5   6   3   0.4   e   E  AB   V   I  12           2
===== 6   7   4   0.5   f   F  AC   U   K  13           3
<<-> * * * End of File * * *

```

To pad numbers with leading zeros, use the COL pad command ('). See “Pad” on page 23.

Number Up a Column The number command operates from the top of the column down. This operation may be performed UP the column by entering the \ reverse operator preceding the command as follows:

```

===== ???????????????????\nmn????????????????????????????????????

```

or the reverse operator may be entered following the command. It may precede or follow any options specified.

```

===== ???????????????????nmn\????????????????????????????????????

```

Or:

```

===== ???????????????????nmn\ (a????????????????????????????????

```

Or:

```
===== ??????????????????mn(a\???????????????????????????????????????? |
```

The \ is not part of the column width being operated on.

Sorting

Note: The COL sort command honors excluded lines. The normal XEDIT SORT does not.

Sorting Lines of Text

To sort lines of text you place a string of **SSSSSSSSSSSS** in every column of the sort field. Use **G** instead of **S** for descending sorts. If you need to specify the sort order of the columns then place numbers within the command string or following the command string. For example to sort based on the column marked 1 first, the column marked 2 second, and then in descending order the column marked 3 you would enter the following:

```
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???s2sssssss?????s1sssssss????g3gggggg???????????????????????? |
```

Suppose you want to sort columns for year, month and then date as follows. Enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=-Col,Que,Y

<<->> * * * Top of File * * *
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????s2?s3?s1????????????????????????????????????????????
===== Bob      12/07/58
===== Betty    11/25/58
===== Gary     12/06/40
===== Mary     11/18/58
===== Sue      03/03/51
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== Gary     12/06/40
===== Sue      03/03/51
===== Mary     11/18/58
===== Betty    11/25/58
===== Bob      12/07/58
<<->> * * * End of File * * *
```

When multiple column sorts are requested without entering sequence numbers, then the sort fields will be sorted from left to right. Sequence numbers from 0 to 9 may be used so that a total of 10 fields may be specified.

Text within a column may be sorted, leaving all text outside the column in place without alteration. This is accomplished by using the **Zone** command with the sort **S** or **G** commands. For example you may want to sort the first column of names in a file by last name first, then by first initial as follows:

```
<<-> * * * Top of File * * *  
      |...+....1...+....2...+....3...+....4...+....5...+....6...+....7...  
===== ???z???s2??s1sssssssssss???z?????????????????????????????????  
===== 1       B. Vaser                          B. Smith  
===== 2       R. Adams                         J. Jones  
===== 3       H. Bare                           K. Thomas  
===== 4       A. Adams                        S. Man  
  
<<-> * * * End of File * * *
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????????
===== 1      A.  Adams      B. Smith
===== 2      R.  Adams      J. Jones
===== 3      H.  Bare       K. Thomas
===== 4      B.  Vaser      S. Man
<<->> * * * End of File * * *
```

```

|           | Column limits marked with z
|           | Primary sort field s1
| |         | Secondary sort field s2

```

Imbedded and trailing sequence numbers are considered to be part of the column width. Leading numbers are not considered to be part of the column width. For example the following columns 14 to 21 marked with the string **s1ssssss** are sorted first and then the single column number 6 with the **s** is sorted second. The **2** is not part of the column width since it is a leading number.

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
         ????2s???????s1ssssss????????????????????????????????????????

```

COL Command Line Options

Current Column (/)

Similar to the XEDIT Prefix command / which makes the marked line the current line, the COL command "/" makes the marked column the leftmost column on the screen.

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5           06/02/89
===== Schultz, Sue             6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy           3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      ...+....4....+....5....+....6....+....7....+....>....+....9....+...10....
===== ???????????????????????????????????????????????????????????????
===== 7249 D49 IBMVM1      06/01/89
===== 6150 9D5           06/02/89
===== 6664 D25 IBMVM3      06/01/89
===== 5234 D41 IBMVM3      06/01/89
===== 3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

Features include:

```
/          Shift screen to the marked column.
/*         Shift as far right as possible and keep a full screen.
/-*        Shift left to column 1.
/0         Shift left to column 1.
/mn        Shift right nn columns.  (nn = number)
/-mn       Shift left nn columns.
/          Entered in the first position, shifts back to column 1.
```

Column One Operations

All of the COL commands that require text to the right of the command, such as Insert, will insert the text exactly below the text as entered in the COL command line. This makes it impossible to insert into column 1 since the COL command line only goes to column 1. For example the following command would insert .* starting in column 2.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== i.* ?????????????????????????????????????????????????????????
```

In order to have text inserted into column 1, use the / command as a prefix operator to these text commands to mean “Insert Here” rather than following the command as normally happens. For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

<<->> * * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== /i.* ?????????????????????????????????????????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin          6150 9D5           06/02/89
===== Schultz, Sue              6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko            5234 D41 IBMVM3      06/01/89
===== Schwind, Sandy            3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????
===== .* Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== .* Schuessler, Marvin          6150 9D5           06/02/89
===== .* Schultz, Sue              6664 D25 IBMVM3      06/01/89
===== .* Schurr, Tomoko            5234 D41 IBMVM3      06/01/89
===== .* Schwind, Sandy            3881 D40 IBMVM3      06/02/89
<<->> * * * End of File * * *
```

/cmd Do the command “here” rather than following the command character. For example:

```
/A(rn)        | Same as the base commands except that the text is placed
/Itxt        | in the column where the / is positioned instead of
/Rtxt        | after the command. This is useful for column 1.
/Otxt        | To insert a blank in column 1, the command would be:
/ctxt        | /i ?????????????????????????????????????????????
/-txt
```

Using an Asterisk (*) in the COL Command Line

COL commands support using the * for all block or string commands. The * will continue the command through the last column. For example **D*** will delete all columns to the end of the line (or to XEDIT's zone setting if not the end of the line) starting at the **D**. The following will delete all text to the end of the line starting with column 40.

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
=====	????????????????????????????????????d*????????????????????????????	

Command Options Using the

(" Commands such as **Sort**, **Number** and **sum (+)** have options that may be entered behind the command. These options are entered using a (. When used, the (and the trailing option are not considered to be part of the column width being defined.

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
=====	nmn(a??	

The above number command covers columns 1 to 3.

Reverse the Execution of a COL Command

When you want a COL command to execute from the bottom to the top of a column, you may precede or follow the command with a \. Commands which benefit from this operation include **Emulate**, **de-emulate** (_), **sum (+)** and **Number**. For example, to reverse the order of the number command you would enter:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
=====	nmn\??	

COL Prefix Functions

The XEDIT prefix area immediately adjacent to the COL command line may be used for special purposes within COL. These functions are listed in this section. Please refer to sections "Automatic Text Alignment" on page 80 and "Sum Numbers in a Column" on page 47 for additional COL prefix commands.

Undo the Last COL Command

The effect of the last executed COL command may be removed by entering **Undo** in the COL prefix area. One level back may be restored.

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
u=====	??	

NOUnd will turn off the undo capability. This is desirable when you wish to reduce system time. Undo requires time to save the text to disk before each execution of a COL command.

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
NOU== ?????????????????????????????????????????????????????????????

```

Picture Columns

Picture Blank Columns When working with columns containing many lines it may be difficult to determine the width of the text within a column. For example, you might want to remove blank spaces between columns but are unsure of what the longest line within each column might be. In other words, which columns contain blanks. You may use the **PIC**ture prefix command to obtain a view of which columns are blank as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
PIC== ?????????????????????????????????????????????????????????????

```

The results are:

```

<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????      ??? ? ? ? ? ? ?      ???????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5              06/02/89
===== Schultz, Sue              6664 D25 IBMVM3       06/01/89
===== Schurr, Tomoko            5234 D41 IBMVM3       06/01/89
===== Schwind, Sandy            3881 D40 IBMVM3       06/02/89
<<->>

```

The columns which contain blanks will show blanks in the COL command line, while the columns which contain text will show the command line symbol.

Picture Columns With Specified Text You may also obtain a picture of which columns contain specific text. You use the **PIC**tu (only 5 characters long!) prefix command as outlined above, but you also place your specified text behind the PIC command or into the COL command line. Any column that contains the text string that you have specified will show the command line symbol. The text match honors XEDIT's CASE IGNORE/RESPECT setting. When the text string is entered within the COL command line it may be placed anywhere within the line. The text string then ends with the first occurrence of the command line symbol (normally a question mark). For example, suppose we want to picture commas in the preceding example so that we may find the length of the longest last name as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
PIC, ?????????????????????????????????????????????????????????????

```

The results are:

<<->			
	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...		
====	???		
====	Savickas, Derral	7249 D49 IBMVM1	06/01/89
====	Schuessler, Marvin	6150 9D5	06/02/89
====	Schultz, Sue	6664 D25 IBMVM3	06/01/89
====	Schurr, Tomoko	5234 D41 IBMVM3	06/01/89
====	Schwind, Sandy	3881 D40 IBMVM3	06/02/89
<<->			

We can now determine that the last comma is in position 11.

We could also use this technique to make sure that all the dates are aligned by testing for columns containing a slash as follows:

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
PIC/	??	

The results should show two distinct locations for this character, columns 53 and 56. If more than two positions are shown, then the dates are not aligned.

Note: The search symbol may also be entered in the COL command line as follows:

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
PIC	,??	

This allows text strings of more than two (2) characters to be entered.

Re-display the Last COL Command

To re-display the last COL command enter a ? in the COL prefix area:

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
?====	??	

Re-execute the Last COL Command

To re-execute the last COL command enter an = in the COL prefix area:

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
=	??	

Reset the COL Command Line

You may reset the COL command line by placing an R in the prefix area. This is very useful after using the include Y or exclude - commands.

Set the COL Command Line Symbol

The results are:

Set the COL Blank Character Symbol

Reverse the Line Order

Chapter 2. Using Basic COL Functions **41**

Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
 PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y

```
<<->
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
\===== ?????????????????????????????????????????????????????????????????
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin         6150 9D5             06/02/89
===== Schultz, Sue             6664 D25 IBMVM3       06/01/89
===== Schurr, Tomoko           5234 D41 IBMVM3       06/01/89
<<->
```

The results are:

```
<<->
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Schurr, Tomoko           5234 D41 IBMVM3       06/01/89
===== Schultz, Sue             6664 D25 IBMVM3       06/01/89
===== Schuessler, Marvin         6150 9D5             06/02/89
===== Savickas, Derral          7249 D49 IBMVM1      06/01/89
<<->
```

Transpose Columns and Lines

To transpose lines into columns or columns into lines enter **TRAN** in the COL prefix area. Transposition will occur at the XEDIT tab positions. The XEDIT tabs will be redefined to the new column positions. You may use the COL semicolon (;) command to set the tabs.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TRAN ?????????;????????????????????????????????????????????
===== 1234      1234.2
===== aaaaaaaa  aaaaa
===== bbbbbbbb  bbbbb
<<->>
```

The results are:

```
<<->>
|...+...1T...+...2T...+...3T...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????
===== 1234      aaaaaaaa  bbbbbbbb
===== 1234.2    aaaaa      bbbbb
<<->>
```

Optionally you may enter a control line containing semicolons (&semi) at the top of the column indicating the start of each column. XEDIT limits the number of tabs to 28. Using the optional control line removes this limit. When using this option, tabs are not reset, but a new control line is generated indicating the new column positions.

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TRAN ?????????????????????????????????????????????????????
===== ;
===== 1234      1234.2
===== aaaaaaaa  aaaaa
===== bbbbbbbb  bbbbb
<<->>
```

The results are:

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????
===== ;      ;      ;
===== 1234      aaaaaaaa  bbbbbbbb
===== 1234.2    aaaaa      bbbbb
<<->>
```

Store COL Commands

To create a log and place COL commands and prefix commands into a file named COL CMDLOG enter:

STORE	$\dots + \dots 1 \dots + \dots 2 \dots + \dots 3 \dots + \dots 4 \dots + \dots 5 \dots + \dots 6 \dots + \dots 7 \dots$
-------	---

All commands will now be saved. These may be useful when saving a command sequence.

To stop saving commands enter **NOSTO**.

[illegible]

XEDIT Commands in the COL Prefix Area

The COL prefix area passes commands not used by COL to XEDIT as follows:

[illegible]

Here the command **ALL** is passed to XEDIT. This allows entering commands such as TOP, BOT, ALL, FILE, QUIT or QQ without the requirement of moving the cursor to the XEDIT command line.

Note: COL contains a special **ALL** command which takes affect whenever text is also entered on the COL command line. see “Editing Blocks of Text” on page 87 for information.

Convenience Commands

The following commands allow you to set up XEDIT parameters while using COL.

Set Tabs: To set the tab settings you may place a semicolon (;) in each column you want to contain a tab.

```
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
          ??,???,?????,?????????,????????,????????????????????????????????
```

Verify Setting: You may place a **V** at the start and end of each column you want to view.

| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
 ===== v??v?
 ??

Or enter strings of **V's**.

==== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
 ????????V????????????????V?????????

Verify in HEX This command is the same as the verify command, but uses **H** instead of **V**'s. The text will be displayed in hex.

Truncation Setting: To change the truncation column, you may place a **T** where you would like your new truncation column.

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
         |????????????????????????????????????????????????????????
=====

```

The results are:

```

===== |...+...1...+...2...+...3...+...4...+...>...+...6...+...7...
         |????????????????????????????????????????????????????????
=====

```

Zone Setting: You may change your zone setting by placing a **Z** at the starting and ending columns where you want to set your XEDIT zones.

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
         |??????????Z??????????????????????????????????????????Z????????????
=====

```

The results are:

```

===== |...+...1|...+...2...+...3...+...4...+...5...+...>...+...7...
         |????????????????????????????????????????????????????????
=====

```

Restoring Verify Setting, Truncation Setting or Zone Setting. Place a **V**, **T** or **Z** as appropriate in the prefix area immediately adjacent to the COL command line. Your setting will be returned to the value it had on entering COL or to the value last set by commands entered in the XEDIT command line or the COL prefix area.

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
v      |????????????????????????????????????????????????????????
=====

```

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
t      |????????????????????????????????????????????????????????
=====

```

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
z      |????????????????????????????????????????????????????????
=====

```

Chapter 3. Using COL Mathematical and Logic Functions

Sum Function

Sum Numbers in a Column

Whenever you have a column of numbers to add you may place a series of + 's above the column to be totaled. The sum will be at the end of the column if a blank field is found. The message line and the XEDIT LASTMSG variable will contain the following:

- A count of the numbers found and added in the column
- Total
- Minimum number found
- Maximum number found
- Average

Make sure the + + + + width is large enough to contain all digits and minus signs since only those columns specified are tested for numbers.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *  
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
===== ??????+?????????????????????????????????????????????????  
=====      2  
=====      3  
=====      4  
=====      5  
=====      6  
=====        
<<->> * * * End of File * * *
```

The results are:

```
5 nbrs Total=20 Min=2 Max=6 Avg=4.00  
The total was inserted at the end of the column
```

```
<<->> * * * Top of File * * *  
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
===== ??????+?????????????????????????????????????????????????  
=====      2  
=====      3  
=====      4  
=====      5  
=====      6  
=====     20  
<<->> * * * End of File * * *
```

Notes

1. Any non-numeric text found in the column is ignored. The task is performed on a word basis. "A5" is not a number.
2. All numbers found in the column are summed.
3. Negative numbers are supported. The minus sign and the number must not be separated by blanks.
4. Trailing negative signs are supported. Make sure that the string of + 's cover the column containing the negative sign.
5. Whenever the numbers being summed contain a dollar sign or commas, then the answer will contain the same.
6. When totaling columns wider than the screen, enter a single + at the start of the column, move the screen right or left and then enter a single + at the end of the column. COL will delay execution until a pair of + 's or a string of + 's have been entered.

Placement of the Total COL will place the total in the first blank field large enough to hold the results using the following rules to select the first available location.

1. At the end of the COL range, right justified with the right most + .
2. One line past the COL range, right justified with the right most + .

Since all numbers in the column are summed, it is possible to sum numbers across one line if the COL depth is 1. For example consider the following string of numbers:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????+++++????????????????????????????????????????????????
=====      1  2  3  4  5  6  7  8
<<->> * * * End of File * * *
```

The results are:

```
8 nbtrs Total=36 Min=1 Max=8 Avg=4.50
The total was inserted at the end of the column
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ??????
=====      1  2  3  4  5  6  7  8      36
<<->> * * * End of File * * *
```

Summing Up a Column The sum command operates from the top of the column down. To perform this operation UP the column placing the total at the top of the column use the \ reverse operator preceding the command as follows:

```
===== ??????????????\+++++???????????????????????????????????????? |
```

Or you may enter the reverse operator following the command as follows:

```
===== ??????????????+++++???????????????????????????????????????? |
```

You may also alter the operation of any of the options listed on the following pages so that the command is performed up the column rather than down the column. The \ is not part of the column width being operated on.

Line Totals You may sum across several lines at a time within the COL range by using the L option following the sum command. For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ++++++ (1????????????????????????????????????????
===== 1    0    2    0
===== 0    1    0    0
===== 1    1    0   55
===== 107   0    0   18
===== 0    27   0  123
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== 1    0    2    0          3
===== 0    1    0    0          1
===== 1    1    0   55          57
===== 107   0    0   18         125
===== 0    27   0  123         150
<<->> * * * End of File * * *
```

Cumulative Line Totals To obtain cumulative line totals, use the **C** option following the sum command as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ++++++ (C????????????????????????????????????????????????????????
===== 1      0      2      0
===== 0      1      0      0
===== 1      1      0      55
===== 107     0      0      18
===== 0      27     0      123
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== 1      0      2      0      3
===== 0      1      0      0      4
===== 1      1      0      55     61
===== 107     0      0      18     186
===== 0      27     0      123     336
<<-> * * * End of File * * *
```

Subtotals Subtotals may be obtained by using the **S** option behind the sum command. Subtotals are placed on blank lines at the end of the **+** string. For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????+++++ (s?????????????????????????????)
===== 10/12/88 844 $3,983
=====                3,094
=====                2,105
=====
=====                739 $1,216
=====                3,327
=====                2,438
=====
=====                                =====
=====
<<-> * * * End of File * * *
```

Using the results of the subtotal option, you may now want to create a total. Enter **+**'s above the values obtained in the subtotal operation as follows:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????+++++?????????????????????????????????
===== 10/12/88 844 $3,983
=====                3,094
=====                2,105
=====                                $9,182
=====                739 $1,216
=====                3,327
=====                2,438
=====                                $6,981
=====                                =====
=====
<<-> * * * End of File * * *
```

The results are:

```

<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
===== 10/12/88   844 $3,983
=====                               3,094
=====                               2,105
=====                               $9,182
=====       739 $1,216
=====                               3,327
=====                               2,438
=====                               $6,981
=====                               =====
=====                               $16,163
<<->> * * * End of File * * *

```

To move the answers up one position within the column, use the column include command **Y** and enter the XEDIT prefix command **D** (Delete) in the prefix area as follows.

```

<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????????????????yyyyyyyy????????????????????
d===== 10/12/88   844 $3,983
=====                               3,094
=====                               2,105
=====                               $9,182
=====       739 $1,216
=====                               3,327
=====                               2,438
=====                               $6,981
=====                               =====
=====                               $16,163
<<->> * * * End of File * * *

```

The results are that the text in the column marked with the string of **Y**'s has been moved up one position, since the first field in that column has been deleted.

```

<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????????????????yyyyyyyy????????????????????
===== 10/12/88   844 $3,983
=====                               3,094
=====                               2,105   $9,182
=====
=====       739 $1,216
=====                               3,327
=====       2,438   $6,981
=====                               =====
=====                               $16,163
=====
<<->> * * * End of File * * *

```

See "Applying XEDIT Commands To Columns" on page 95 for more information on column exclude.

Summing Columns Containing More Than One Number in Each Line All numbers within the area defined by the COL depth and the +’s will be summed:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????+++++++????????????????????????????????????????????
=====      1      4
=====      2      5
=====      3      6
=====
<<-> * * * End of File * * *
```

results in:

```
6 nbrs Total=21 Min=1 Max=6 Avg=3.50
The total was inserted at the end of the column
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????+++++++????????????????????????????????????????????
=====      1      4
=====      2      5
=====      3      6
=====      21
<<-> * * * End of File * * *
```

Obtaining SUM Results When Calling COL as a Macro. Results of the addition operation (++) when calling COL as a macro are available in LASTMSG.1 from the XEDIT EXTRACT /LASTMSG/ command. The form of the values in the returned message is:

```
count total minimum maximum average results
```

This matches the message displayed on the screen in interactive mode. The numeric results are in LASTMSG.1. The extraneous text has been removed. Also a **results** parameter has been added which contains a **Y** or **N** depending on whether the answer has been placed in the file or not. Results are not provided when using the **L** or **C** options. When using the **S** option, the message string will include each subtotal result up to the allowable length of XEDIT’s last message support.

Significant Digits The number of significant digits for the SUM command may be changed from the default setting of 12. Change the digit setting in your PROFCOL XEDIT file or use the COL prefix command area. For example, to use 15 significant digits you would enter:

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
DIG15 ??????+++++++????????????????????????????????????????????
```

The minimum value for digits that may be entered is 9.

National Language Support for the SUM Command Three options are available to alter the form of the decimal point and thousands separator in the numbers being added. These are:

PERIOD for 1,234,567.89 (The Default)
 COMMA for 1.234.567,89
 FRENCH for 1 234 567,89 or 1 234 567.89

These options may be entered in your PROFCOL XEDIT, ("COL Profile" on page 4), as an option in the COL command, ("COL Macro" on page 173), or in the COL prefix area. To use a period (the default) to show decimals in numbers (1,234,567.89) enter:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
PERIO	??	

To use a comma to show decimals in numbers (1.234.567,89) enter:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
COMMA	??	

To use a period or comma to show decimals and blanks for thousand separators in numbers (1 234 567.89 or 1 234 567,89) enter:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
FRENC	??	

Mathematical Operations and REXX Interface

The COL mathematical and logical operations use the **Y** command to mark the columns to be operated upon and the **=** command to locate the placement of the answer. Any valid REXX expression, function, arithmetic, comparison, logical or string operation may be used to generate the answer. The steps for the mathematical operations are:

1. Enter your mathematical operators, function or expression in one of the following three ways:
 - a. In the COL command line between the columns, or
 - b. Within the data, or
 - c. In the XEDIT command line using named columns.
2. Place **Y**'s (as in column include) above the columns containing your source information.
3. Place a string of equal signs above the column that is to receive the results.
4. Press Enter.

Error messages are displayed whenever the indicated operation results in an invalid REXX expression. These error messages may be turned off by setting MATH_MSG = "OFF" in PROFCOL XEDIT.

For example, two columns can be multiplied by entering the * operator in the COL command line between the source columns marked by Y's as shown below:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????YYYYYY * YYYYYY=====????????????????????????
=====
=====          5          2
=====          100         50
```

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????YYYYYY / YYYYYY * 10 =====????????????????????
=====
=====          5          2          10
=====          100         50         5000
```

You may also use constants within your expression. To specify the resulting decimal point location you may place a decimal point within the string of equal signs. For example, to divide two columns, multiply the results by 10 and specify the decimal point location you would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????YYYYYY / YYYYYY * 10 =====.=====????????????????
=====
=====          123.5        22
=====          100          3
```

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????YYYYYY / YYYYYY * 10 =====.=====????????????????
=====
=====          123.5        22          56.13636
=====          100          3          333.33333
```

Notes:

1. The columns are operated upon left to right.
2. The equal signs for the answer placement may be located anywhere.
3. The decimal in the = string is optional and locates the decimal point.
4. Existing text within the = answer column is overlaid.
5. The columns themselves may include the operators.
6. You may use REXX functions.

The text columns marked with **Y** are extracted, the operators entered in the command line are interleaved into the text and the resulting text string passed to REXX for interpretation. The result is then overlaid into the = location.

Decimal Point Location You may specify the decimal point location in the answer field by imbedding a decimal point within the string of equal signs as in the above examples.

Significant Digits If the length of the = string exceeds 9, REXX's Numeric Digits setting is increased to match the length during this operation.

COMMA and FRENCH options. The Comma and French options supported for COL's sum operation are not supported during these mathematical operations. REXX directly evaluates all expressions.

Three Ways of Entering Your Expression

In The COL Command Line The previous examples illustrate the process of entering your mathematical expressions within the COL command line itself. You "build" your expression entirely on the command line.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
[Math] ??????????YYYYYY * YYYYYY=====????????????????????????????
=====
          5          2
=====
          100         50
```

MATh in the COL prefix area is optional and not required when the first COL command line character is one of the following:

1. =
2. Y, and the COL command line also contains =
3. ', and the COL command line also contains =
4. ", and the COL command line also contains =
5. (, and the COL command line also contains =

MATh would be required in the following example placing random numbers in the specified column:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
Math  ???RANDOM() ??????????????=====????????????????????????????????
```

The REXX function name **RANDOM** does not meet any of the above rules. Without the **MATh** prefix entry, COL would decode this entry to be a **R**eplace command.

Note:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
Math  ???DATA TYPE (YYYYYYY) ??=====????????????????????????????
```

is NOT valid since the function name includes a **Y** in it's name and that column would be considered to be part of the field specifications.

In this case you must use one of the following which are explained in subsequent sections.

1. Named columns.
2. Enter DATATYPE into the text itself.

Within the Data The mathematical operators may optionally be within the data itself. For example, let us revisit the first example in this section of multiplying two columns together. The operator could be within the text and the operation would be as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
|?????????YYYYYYYYYYYYYYYYY=====????????????????????????????
=====
|          5      *      2
|          100     *     50
=====
```

The results are:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== |????????????????????????????????????????????????????????????
=====
|          5      *      2      10
|          100     *     50      5000
=====
```

You also might use this method when dealing with function names which include the math symbol **Y**.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== |?????YYYYYYYYYYYYYYYYYYY?=====????????????????????????????????
=====
|          Datatype(  1.5      )
|          Datatype(   A       )
=====
```

The results are:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== |????????????????????????????????????????????????????????????
=====
|          Datatype(  1.5      )  NUM
|          Datatype(   A       )  CHAR
=====
```

In the XEDIT Command Line Using Named Columns You may name columns and then enter your expressions on the XEDIT command line. Default columns names are as follows:

1. The first column is named **y.1**
2. The second column is named **y.2**
3. And so forth.
4. The answer column prior to evaluation is named **y.0**
5. The “previous answer” is named **y.00**

To divide the second column by the first we would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ov1,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????YYYYYYYY??YYYYYYYYY?=====?????????????????????????????

====> [MATH] y.2 / y.1
```

The keyword *MATH* is optional and not required when the expression contains **Y.** within the text string. Therefore **y.2 / y.1** would be adequate in the above example. The following would also be valid since **Y.** is in the expression.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????YYYYYYYY?????????????=====????????????????????????????????

====> Datatype(y.1)
```

The following Random function would require the keyword *MATH* since **Y.** is not specified in the expression:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????=====????????????????????????????????????

===== MATH RANDOM()
```

User Named Columns: You may also assign individual column names as follows:

COL Entry	Column Name
YYYYYYYYYYYYYY	y.1 (default for the first column)
YYYYYYYYYYYYY2	y.2
YYYYYYYYYYYYYa	y.a
YYYYYYaYYYYYY	y.a
YYYYYYYYYYYYY.a	y.a
y.aYYYYYYYYYYY	y.a
YYYYY.....a	y.a
y.....a	y.a
yayYYYYYYYYYYY	y.a
YYYYYYYYYYYtot	y.tot

Notes:

1. If one column is explicitly named, then they must all be explicitly named.
2. The name characters are part of the column width.
3. The answer column name cannot be changed from the default of **y.0**.

To divide the second column by the first using user named columns you would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,t=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????YYYYYYa??YYYYYYb?=====????????????????????????????

```

```
=====> y.b / y.a
```

Update in place The variable **y.0** contains the contents of the answer column prior to evaluation of the expression. You may enter expressions as follows to update a column in place.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????=====????????????????????????????????????????
=====
=====          5
=====          100

=====> y.0 - 3
```

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????=====????????????????????????????????????????
=====
=====          2
=====          97
```

Previous Column Answer The previous answer is contained within a variable named **y.00**. This variable may be used for various purposes such as constructing running totals. The first answer column may contain the initial value to be used for the evaluation. The default initial value is 0 unless y.1 is used and it does not contain a numeric value. In this case the default initial value is blank. Consider the following request for running totals with the initial value set to 5.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????yy?=====????????????????????????????????????????
=====          55      5
=====          45
=====          33

=====> y.1 + y.00
```

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????=====????????????????????????????????????????
=====          55      60
=====          45     105
=====          33     138
```

Mathematical Operators (REXX)

Any valid REXX operator may be used. These include:

Arithmetic:

()	Parentheses
+	Add
-	Subtract
*	Multiply
/	Divide
%	Divide and place the integer part
//	Divide and place the remainder
**	Raise a number to a whole-number power

Comparative:

= =	Terms are exactly equal
=	Terms are equal
≠ =	Not exactly equal
≠	Not equal
>	Greater than
<	Less than
> <	Greater than or less than
< >	Greater than or less than
> =	Greater than or equal to
< =	Less than or equal to
≠ <	Not Less than
≠ >	Not Greater than
< <	Strictly Less than
< < =	Strictly Less than or equal to
≠ < <	Strictly NOT Less than
> >	Strictly Greater than
> > =	Strictly Greater than or equal to
≠ > >	Strictly NOT Greater than

Logical:

¬	Not
	Or
&	And
&&	Exclusive OR

String concatenation:

(blank)	Concatenate
	Concatenate without intervening blank

User Functions and Exits You may use any internal or external user REXX function. For example consider the following hypothetical user written REXX function named **LESSTHAN EXEC A**.

```

/**/
  Parse Arg x
  If x < 25000 Then a = ¢Yes¢
  Else a = ¢No¢
Return a

```

You could use the above function as follows to indicate which entries within a column have values less than 25000 and which do not.

```

Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,t=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????yyyyyy??==????????????????????????????????????
=====          10000
=====          30000
=====          25000

=====> lessthan(y.1)

```

The results are:

```

      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
=====          10000   Yes
=====          30000   No
=====          25000   No

```

Note: COL uses many internal functions with names which are one or two characters long. In order to avoid possible conflicts between internal COL and user functions, it is suggested that you name your functions with three or more characters. REXX does allow forcing use of your external function by placing the function name in quotes:

```

=====> ¢lessthan¢(y.1)

```

Built-in COL Function

NUMBER

NUMBER(column_variable)

column_variable

The column variable name such as **y.1**.

This function returns a zero when the column contains a blank or non-numeric text. If the column contains a valid number, then the number is returned.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????yyy =====????????????????????????????????????
=====          1
=====
=====          2
=====          X
```

```
====> number(y.1)
```

The results are:

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????11 =====????????????????????????????????????
=====          1  1
=====          0
=====          2  2
=====          X  0
```

Chapter 4. Using COL Text Functions

Text Alignment

How do you place data that is not already in a column into a column?

Queue or Align Specified Text

Specified text may be “Queued up” or aligned as shown in the following example where the left parenthesis is used to line up the initials. We’ll use the left parenthesis as the key:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone  
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *  
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
===== ?????????????????q(????????????????????????????????????????  
===== Smith (BH)  
===== Ho (DC)  
===== Von Lintel (AM)  
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *  
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
===== ?????????????????????????????????????????????????????????  
===== Smith                (BH)  
===== Ho                   (DC)  
===== Von Lintel           (AM)  
<<-> * * * End of File * * *
```

The setting of XEDIT’s Case Ignore|Respect setting is honored when searching for the text string.

- | XEDIT’s ZONE setting is honored during this operation. Text outside of the zone are not moved. The companion commands for **Q** are the justify command using a **JL** (Justify Left) and the remove trailing blanks command **B**. See “Left Justify” on page 73 and “Removing Trailing Blanks” on page 71 for more information.

The text being **Queued** is not limited to a single character but can be any character string as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????gary????????????????????????????????????
===== Richard B. Ho
===== Gary G. Vair
===== Brenda A. Person
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????gary????????????????????????????????????
===== Richard B. Ho
===== Gary G. Vair
===== Brenda A. Person
<<-> * * * End of File * * *
```

When a **Q** without any following text is entered or a blank is entered as the text string, then the search is for the preceding word. For example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????q????????????????????????????????????
===== Richard B. Ho
===== Gary G. Vair
===== Brenda A. Person
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????q????????????????????????????????????
===== Richard B. Ho
===== Gary G. Vair
===== Brenda A. Person
<<-> * * * End of File * * *
```

There are times that you may want to line up text located behind a specified character leaving that character in place. You may do this by using **QQ** command. For example, let us line up the text following the comma, but leave the comma in place as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <=W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????qq,????????????????????????????????????????
===== Smith, B. H.
===== Ho, D. C.
===== Von Lintel, A. M.
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????qq,????????????????????????????????????????
===== Smith,           B. H.
===== Ho,             D. C.
===== Von Lintel,     A. M.
<<-> * * * End of File * * *
```

| Using Queue Within Specific Columns

| There are times that you may want to perform the **Q** operation on specific columns while excluding
| other columns. The Queue command honors the XEDIT Zone setting. For example let us consider an
| expanded example of the **Q** command to align text.

```
| _____  
| Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
| PF1=Hlp/2=CMDlist/3=Exit <=>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y  
|  
| <<->> * * * Top of File * * *  
| |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
| ===== z????????????q(???z????????????????????????????????????  
| ===== Smith (BH) 739 844  
| ===== Ho (DC) 525  
| ===== Von Lintel (AM) 523 321  
| <<->> * * * End of File * * *
```

| The results are:

```
| <<->> * * * Top of File * * *  
| |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
| ===== ?????????????????????????????????????????????????????  
| ===== Smith (BH) 739 844  
| ===== Ho (DC) 525  
| ===== Von Lintel (AM) 523 321  
| <<->> * * * End of File * * *
```

| The text outside of the zone setting is not moved. If the **Zs** are entered at the same time as the **Q**,
| then the zone request is considered to be temporary and XEDIT zones are not changed.

Removing Trailing Blanks

You may remove all trailing blanks up to and including a specified column by using the **B** command. This is in many ways the inverse operation of the **Queue** command. Consider the following example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????b????????????????????????????????????????
===== Smith                , B. H.
===== Ho                   , D. C.
===== Von Lintel           , A. M.
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????
===== Smith, B. H.
===== Ho, D. C.
===== Von Lintel, A. M.
<<->> * * * End of File * * *
```

| Removing Trailing Blanks Within Specific Columns

| The **B** command also honors the XEDIT Zone setting. For example let us consider the expanded example of the **B** command to compress text.

```
| _____  
| Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
| PF1=Hlp/2=CMDlist/3=Exit <=>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=>Col,Que,Y  
  
| <<->> * * * Top of File * * *  
| |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
| ===== z?????????????b?????z?????????????????????????????????????  
| ===== Smith (BH) 739 844  
| ===== Ho (DC) 525  
| ===== Von Lintel (AM) 523 321  
| <<->> * * * End of File * * *
```

| The results are:

```
| <<->> * * * Top of File * * *  
| |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
| ===== ??????????????????????????????????????????????????????????  
| ===== Smith (BH) 739 844  
| ===== Ho (DC) 525  
| ===== Von Lintel (AM) 523 321  
| <<->> * * * End of File * * *
```

| The text outside of the zone setting is not moved. If the **Zs** are entered at the same time as the **B**, then the zone request is considered to be temporary and XEDIT zones are not changed.

Text Justification

Left Justify You may left justify data using the COL justify commands.: Consider the following column data with department numbers.

```
Smith (BH)          539 001-2
Ho (DC)             834 025-1
Von Lintel (AM)     927 010-1
```

After a **q** command, the department numbers will no longer be in columns.

```
Smith      (BH)          539 001-2
Ho         (DC)          834 025-1
Von Lintel (AM)         927 010-1
```

However a "Left Justify" command will put the data back into columns as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????????j1????????????????????????????????????????
===== Smith      (BH)          539 001-2
===== Ho         (DC)          834 025-1
===== Von Lintel (AM)         927 010-1
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????????j1????????????????????????????????????????
===== Smith      (BH)          539 001-2
===== Ho         (DC)          834 025-1
===== Von Lintel (AM)         927 010-1
<<-> * * * End of File * * *
```

entire line

To left justify text within a specific column, enter a pair of **J**'s. The first **J** marks the start of the column and the second **J** marks the end of the column. Either **J** may receive the **L** operator. The following example left justifies text within the marked columns:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????jl?????????j????????????????????????????????????????
=====  abc          jfk          xyz
=====  abc          def          xyz
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????jl?????????j????????????????????????????????????????
=====  abc          jfk          xyz
=====  abc          def          xyz
<<->> * * * End of File * * *
```

| | Left justify the marked column

Right Justify Text may be right justified within your file by using the justify right command. To right justify the text enter the **JR** command at the start of the column(s) to be justified as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????jr????????????????????????????????????????
===== abc          jfk          xyz
===== abc          def          xyz
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????jr????????????????????????????????????????
=====      abc          jfk          xyz
=====      abc          def          xyz
<<-> * * * End of File * * *
```

```
start of line      |
```

When a pair of **J**'s are entered, then the data in the column marked by the two **J**'s is right justified (either **J** may receive **R** operator) as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????j?????????jr????????????????????????????????????????
===== abc          jfk          xyz
===== abc          def          xyz
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????jr????????????????????????????????????????
===== abc          jfk          xyz
===== abc          def          xyz
<<-> * * * End of File * * *
```

```
      |              | Right Justify the column between the J's
```

Justify Both Left and Right You may justify text right and left to the column boundaries by using the **B** operator. The text within the column is justified in both directions. The following example shows text which is to be justified left and right:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????jb????????????????????????????????????j????????????????????
=====      The brown cow jumped over the moon
=====      while the dog ran away with the spoon
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????jb????????????????????????????????????j????????????????????
=====      The brown cow jumped over the moon
=====      while the dog ran away with the spoon
<<-> * * * End of File * * *
```

| |

Now let us consider justifying data rather than text. In the following example there are 3 items in each line, but these items are not aligned. You may align them as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???jb????????????????????????????????????????j????????????????????
=====      abc      jfk      xyz      there is
=====      abc      jfk      xyz      lots and lots
=====      abc      jfk      xyz      of text over here
<<-> * * * End of File * * *
```

The result are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
=====      abc      jfk      xyz      there is
=====      abc      jfk      xyz      lots and lots
=====      abc      jfk      xyz      of text over here
<<-> * * * End of File * * *
```

```
      |
      |
Text is justified both left and right in the column.
```

Centering Text You may center each line within the COL range at the position marked by the **J** or you may center text within a column between a pair of **J**'s.: If one **JC** is entered, then each individual line is centered at the **J** as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????jc????????????????????????????????
=====          abc          jfk          xyz
=====                                     President
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????jc????????????????????????????????
=====          abc          jfk          xyz
=====                                     President
<<-> * * * End of File * * *
```

start of line

end of line

When a pair of **J**'s are entered, then the text between the **J**'s is centered within that column (either **J** may receive the **C** operator) as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????jc?????????????j????????????????????????????????????
=====  abc   jfk          xyz
=====  abc   define      xyz
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????jc????????????????????????????????
=====  abc   jfk          xyz
=====  abc   define      xyz
<<-> * * * End of File * * *
```

Text is centered within the column

Spacing Between Words When Using Justify Commands

The justify commands allow you to specify the number of spaces between each word. Place a number next to one of the **J** operators. The number may be any number including 0. For example, let us left justify the text within columns 1 and 35 and place one blank between each word as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== jll????????????????????????????????????????????????????????
=====      abc   jfk           xyz   more stuff
=====      abc      def       xyz   and even more
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????
===== abc jfk xyz           more stuff
===== abc def xyz          and even more
<<->> * * * End of File * * *
```

| with one space

Automatic Text Alignment

Automatic Tabbing When working with text that contains Document Composition Facility (SCRIPT) tab controls or for that matter any other separator character (including blanks) such as:

```
.ti 05
.tp 5 15 25 33 50
811 1211 Williams 544.40 5386
811 Carson 8356
```

it is sometimes necessary to align the text for editing. You may also want to place the text back into compressed form for SCRIPT when you have finished. You can automatically tab this type of text using the ATAB (Auto TAB) COL prefix command. The **ATAB** is placed in the prefix area adjacent to the COL command line. Place your separator character behind the **ATAB** command as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
atab  ?????????????????????????????????????????????????????????????
===== 811 1211 Williams 544.40 5386
===== 811 Carson 8356
<<->> * * * End of File * * *
```

The results are:

```
Tab character specified was † †
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |.T+.T.1...T+...2...T+...3.T...+...4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== 811 1211 Williams 544.40 5386
===== 811 Carson 8356
<<->> * * * End of File * * *
```

The text is tabbed based on the longest entry in each column. Two (2) spaces are placed between each resulting column. The **T** in the scale line shows the XEDIT TAB setting that has been established as a result of this command.

You may change the default spacing (2) between columns when using the auto tab command **ATAB** during your edit session as follows:

```
TS5      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
TS5      ?????????????????????????????????????????????????????????????
```

The default of 2 spaces may also be changed in your PROFCOL XEDIT file using the **tab_space = n** option, where n is the required spacing.

Using Manual Tab Positions If you want to set your own tab positions, rather than have COL automatically set up the tabs based on the text length, then use the TAB command. You may define the XEDIT tabs by using the XEDIT SET TABS command or by entering a semicolon in the COL command line to designate a tab position. See “Set Tabs” on page 44. Consider the following example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
tab   ???;????????;????????????;????????;????????;????????????????
===== 811 1211 Williams 544.40 5386
===== 811 Carson 8356
<<-> * * * End of File * * *
```

The results are:

```
Tab character specified was † †
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...T...1...T...2...+...T...+...T...+...T...+...6...+...7...
===== ?????????????????????????????????????????????????????????
=====      811      1211      Williams 544.40      5386
=====      811      Carson      8356
<<-> * * * End of File * * *
```

Compressing Text Use the **COMP** command to compress existing text or text expanded as a result of using **ATAB** or **TAB** commands. Specify the tab settings to be used if the columns have changed positions since the tab expansion operation. Text will be compressed as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |.T+..T.1...T+....2...T+....3.T..+....4....+....5....+....6....+....7...
comp  ?????????????????????????????????????????????????????????????????
===== 811 1211 Williams 544.40 5386
===== 811      Carson      8356
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???????????????????????????????????????????????????????????????
===== 811 1211 Williams 544.40 5386
===== 811 Carson 8356
<<->> * * * End of File * * *
```

The SCRIPT tab character (↵ here) is optional. If not entered, then the character used with **ATAB** or **TAB** will be reinserted.

Note: The XEDIT tab positions must be set at the start of each column. **ATAB** will set these positions, but if you move column positions, then you must redefine the tab positions before COMPRESSing the text. This may be done with the XEDIT SET TABS command or by using COL's tab set command as follows:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
comp  ??;????;????;????????;????????;????????????????????????????????
===== 811 1211 Williams 544.40 5386
===== 811      Carson      8356
<<->> * * * End of File * * *
```

where ; is COL's tab set command.

Justifying Text Using Tab Settings

After you have expanded the text using ATAB or TAB you may want to justify the text in all columns. For example sorting may be required using several columns. The following COL prefix commands are similar to the COL justify **J** commands which operate on one column. see “Paragraph Word Flow” on page 84 gives more information on justify commands. However, these commands simultaneously justify all columns based on the current tab settings. This includes the column starting at column 1 and ending with the first tab position. Before compressing the text using the **COMP** prefix command, insure that the text is left justified to the tab positions.

Right Justify Use **TR** to right justify text between tab settings.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TR== ?????????????????????????????????????????????????????
```

Left Justify Use **TL** to left justify text between tab settings.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TL== ?????????????????????????????????????????????????????
```

Center Use **TC** to center text between tab settings.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TC== ?????????????????????????????????????????????????????
```

Justify Both Left and Right **TB** justifies the text within the tab positions, aligning the text in both directions.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
TB== ?????????????????????????????????????????????????????
```

Paragraph Word Flow

Word flow is similar to the COL justify commands, except that the justification process takes place using entire paragraphs rather than one line at a time. A paragraph ends when a blank line within the column is found. By default double space occurs for word flow operations after periods, question marks and exclamation points if the next word is capitalized or it is the last character on the line in the original text. The spacing between sentences may be changed by altering the value for SENTENCE_SPACE in your PROFCOL XEDIT file.

Left Justify and Flow the Text Within a Column

To flow your text with the text left justified in the column use the < command as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????<????????????????????????????????????????????????????????
=====  *   A big fat cow was eating all the      Text
=====  *   green grass, while the sun           outside
=====  *   was shining. So too were the         the column
=====  *   birds.                               will not
=====  *                                       be
=====  *   But a lion was watching              altered
=====  *   this cow.                            at all.
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????<????????????????????????????????????????????????????????
=====  *   A big fat cow was eating all the      Text
=====  *   green grass, while the sun was         outside
=====  *   shining. So too were the birds.         the column
=====  *                                       will not
=====  *                                       be
=====  *   But a lion was watching this cow.       altered
=====  *                                       at all.
<<->> * * * End of File * * *
```

Right Justify and Flow the Text Within a Column

To flow your text with the text right justified in the column use the `>` command as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone  
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ??????>????????????????????????????????????????????????????????  
===== *   A big fat cow was eating all the   Text  
===== *   green grass, while the sun        outside  
===== *   was shining. So too were the      the column  
===== *   birds.                            will not  
===== *                                       be  
===== *   But a lion was watching           altered  
===== *   this cow.                         at all.  
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ??????>????????????????????????????????????????????????????????  
===== *   A big fat cow was eating all the   Text  
===== *   green grass, while the sun was    outside  
===== *   shining. So too were the birds.    the column  
===== *                                       will not  
===== *                                       be  
===== *   But a lion was watching this cow. altered  
===== *                                       at all.  
<<->> * * * End of File * * *
```

Left and Right Justify and Flow the Text Within a Column.

To flow your text and both right justify and left justify the text in the column use the **w** command as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone  
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ?????w????????????????????????????????w????????????????????????????  
===== * A big fat cow was eating all the Text  
===== * green grass, while the sun outside  
===== * was shining. So too were the the column  
===== * birds. will not  
===== * be  
===== * But a lion was watching altered  
===== * this cow. at all.  
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *  
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
===== ?????????????????????????????????????????????????????????????????  
===== * A big fat cow was eating all the Text  
===== * green grass, while the sun was outside  
===== * shining. So too were the birds. the column  
===== * will not  
===== * be  
===== * But a lion was watching this cow. altered  
===== * at all.  
<<->> * * * End of File * * *
```

Compressing Text Into a Narrower Column

You may squeeze text into a column while using one of the three word flow <, > or W commands by specifying the Zone for the results. The text will be placed in the zone as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????<??z????????????????????z??z<????????????????????????????
===== *   A big fat cow was eating all the      Text
===== *   green grass, while   the sun         outside
===== *   was shining. So too were the         the column
===== *   birds.                                     will not
===== *                                     be
===== *   But a lion was watching              altered
===== *   this cow.                             at all.
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????<??z????????????????????z??z<????????????????????????????
===== *   A big fat cow was eating              Text
===== *   all the green grass, while            outside
===== *   the sun was shining. So               the column
===== *   too were the birds.                     will not
===== *                                     be
===== *   But a lion was watching                altered
===== *   this cow.                              at all.
<<->> * * * End of File * * *
```

Editing Blocks of Text

Enter the **ALL** command in the COL prefix area to display predefined groups or blocks of text. The command is a modified XEDIT **ALL** command which not only displays the requested tag, but also displays all text up to a predefined ending tag. The ending tag may also be displayed depending on the tag in question. The groups are predefined in COLALL XEDIT. See Table 1 on page 92 for a listing of the predefined groups. The user may add additional block definitions as necessary to the COLALL XEDIT file.

See “Editing GML or BookMaster Table Columns” on page 136 for information on editing GML or BookMaster tables.

The column searched is defined as follows. The position of the text as entered in the COL command line defines the starting column. The column ends with the XEDIT zone setting.

For example in BookMaster/GML files you may display ordered list blocks within the range covered by COL by entering **ALL** in the COL Prefix area and **:ol** in the COL command line as follows:

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow, +=Sum, Low, K=Up, Just, Blk, Nbr, /=Col, Que, Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
ALL== :ol????????????????????????????????????????????????????????????
===== Wouldn't it be nice to know definitions for the following?
===== :ol.
===== :li.lawn
===== :li.gopher
===== :li.agapanthus
===== :eol.
===== We could build a definition list,
===== but in reality, they are just:
===== :ol.
===== :li.weeds
===== :li.rodent
===== :li.flower
===== :eol.
===== I thought so.
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== :ol.
===== :li.lawn
===== :li.gopher
===== :li.agapanthus
===== :eol.
===== ----- 2 line(s) not displayed -----
===== :ol.
===== :li.weeds
===== :li.rodent
===== :li.flower
===== :eol.
===== ----- 1 line(s) not displayed -----
<<->> * * * End of File * * *
```

And to display DO/END groups in a REXX exec that start in column 3 or greater you would enter:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
ALL== ??????????????????????????????????????????????????????????????
===== Do x=1 To cm.0
=====   dd1=Pos(cm.x,cmdltype)
=====   Do While dd1>0
=====     dd2=Pos(cm.x,cmdltype,dd1+2)
=====     If dd2>0 Then Do
=====       cmdline=O(¢ ¢,cmdline,dd2,2)
=====       cmdline=Overlay(dd2-dd1+2|rexblnk,cmdline,dd1+1)
=====       Parse Upper Var cmdline cmdltype
=====       dd1=Pos(cm.x,cmdltype)
=====     End
=====     Else dd1=0
=====   End
===== End
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????????????????????????????????????????????????????????
===== ----- 2 line(s) not displayed -----
=====   Do While dd1>0
=====     dd2=Pos(cm.x,cmdltype,dd1+2)
=====     If dd2>0 Then Do
=====       cmdline=O(¢ ¢,cmdline,dd2,2)
=====       cmdline=Overlay(dd2-dd1+2|rexblnk,cmdline,dd1+1)
=====       Parse Upper Var cmdline cmdltype
=====       dd1=Pos(cm.x,cmdltype)
=====     End
=====     Else dd1=0
=====   End
===== ----- 1 line(s) not displayed -----
<<->> * * * End of File * * *
```

SCRIPT running headings may also be displayed. The running heading tag is not in the predefined tag table. Any text string which is not in the table uses the same text string for its ending tag. In other words, the start and end tag are the same. This is the case for SCRIPT running headings, an example of which is shown below.

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
ALL== .rh????????????????????????????????????????????????????????????
===== This is another example.
===== With lots of stuff here.
===== .rh on
===== .ce IBM
===== .rh off
===== And then there is more text from here on.
<<->> * * * End of File * * *
```

The results are:

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== ----- 2 line(s) not displayed -----
===== .rh on
===== .ce IBM
===== .rh off
===== ----- 1 line(s) not displayed -----
<<->> * * * End of File * * *
```

Specifying the Start and End Label You may specify your own start and end label in the command line for those labels whose end tag has not been predefined. For example, assume that the start text for the group is **aa** and that the end text is **bb**. Enter the the start tag and end tag with a / between the two as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
all== aa/bb????????????????????????????????????????????????????????????
===== Some text before the group
===== aa
===== more text
===== and more text
===== bb
===== after the group
<<->> * * * End of File * * *
```

This will yield the following:

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== aa
===== more text
===== and more text
===== bb
===== ----- 1 line(s) not displayed -----
<<-> * * * End of File * * *
```

COLALL XEDIT

The ALL command as used above in COL may also be called directly as follows.

```
COLALL /rtarget/[etarget/]
```

Figure 5. COLALL

All Text in the block between *rtarget* and *etarget* are displayed. The *rtarget* is the start of the block of text. The *etarget* is an optional identifier for the ending of the group. If *etarget* is not specified, then the predefined endtag in COLALL XEDIT is used. If no predefined endtag is found, then *rtarget* is also used as the *etarget*.

For example you could enter the following to set PF1 to display BookMaster Cell 1's.

```
SET PF1 MACRO COLALL /:C 1/
```

The predefined groups are listed in the following table. The predefined blocks in COLALL XEDIT also include a parameter to show if these groups are to display the end tag when the group is displayed. For example REXX **END** is displayed for **DO** groups. However the end tag for GML/BookMaster cells is not displayed. This endtag parameter is named *xxx_endtag* where *xxx* is the name of the tag and is set to **Y** or **N**.

Group ID	End of group	Display End Tag
GML/BookMaster groups		
address	eaddress	Y
annot	eannot	Y
apl	eapl	Y
bin	ebin	Y
c	c row etable ethd ec tcap	N
caution	ecaution	Y
cgraphic	ecgraphic	Y
char	echar	Y
cit	ecit	Y
cletter	eletter	Y
codel	ecodel	Y
compl	ecompl	Y
danger	edanger	Y
dd	dt edl	N
dialog	edialog	Y
direct	edirect	Y
dl	edl	Y
dt	dd	N
f	ef	N
fig	efig	Y
figlist	efiglist	Y
fle	efle	Y
fn	efn	Y
gd	gt egl	N
gl	egl	Y
gt	gd	N
grid	egrid	Y
h	h	N
hex	ehex	Y
hp	ehp	Y
li	li lp eol esl eul edl egl	N
lblbox	elblbox	Y
len	elen	Y
line	eline	Y
lq	elq	Y
map	emap	Y
mapqfix	emapqfix	Y
md	emd	Y
msgl	emsgl	Y
mv	emv	Y

Group ID	End of group	Display End Tag
notel	enotel	Y
notices	enotices	Y
nt	ent	Y
ol	eol	Y
parml	eparml	Y
partsasm	epartsasm	Y
pk	epk	Y
prolog	eprolog	Y
psc	epsc	Y
ptoc	eptoc	Y
pv	epv	Y
q	eq	Y
qualif	equalif	Y
question	equestion	Y
rev	erev	Y
revcw	erevcw	Y
revpf	erevpf	Y
revpp	erevpp	Y
revpt	erevpt	Y
revslug	erevslug	Y
rk	erk	Y
row	erow row etable	N
schedule	eschedule	Y
screen	escreen	Y
snotice	esnotice	Y
sl	esl	Y
step	estep	Y
sub	esub	Y
sup	esup	Y
synopsis	esynopsis	Y
table	etable	Y
tce	etce	Y
tdesc	etdesc	Y
thd	ethd	Y
tft	etft	Y
title	etitle	Y
titlep	etitlep	Y
tle	etle	Y
tlist	etlist	Y
tnote	etnote	N
toc	etoc	Y

Group ID	End of group	Display End Tag
tp	etp	Y
ul	eul	Y
vnotice	evnotied	Y
warning	ewarning	Y
xmp	exmp	Y
xph	exph	Y
REXX groups		
do	end	Y
Groups not identified		
other	same as entered in Group ID	Y

Table 1. COLALL Predefined Blocks of Text

You may add to or alter the predefined tags as necessary in COLALL XEDIT.

Chapter 5. XEDIT Commands Within COL

Applying XEDIT Commands To Columns

The following column include and column exclude functions provided by COL do for columns what the XEDIT exclude does for lines. The result is that XEDIT

prefix commands,
command line commands and
PF key functions

are excluded from operating on columns other than the ones selected. FILE and SAVE still operate on the entire file regardless of XEDIT excluded lines or COL excluded columns.

Column Include (Y)

When you want to apply an XEDIT prefix command, XEDIT command entered in the XEDIT command line or XEDIT command entered via a PF key to specific columns leaving the remaining columns unaltered, you may mark those columns to be operated on within COL with a **Y** (as in YES). Once you have included the column, then XEDIT commands will only operate on that column. **Y*** shows that the marked column through the end of the line or the current XEDIT zone setting is to be included in XEDIT commands. For example, let us use the XEDIT prefix commands insert, move, following and delete in the column of names and phone numbers below:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone  
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *  
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
===== ??????????????????????y*?????????????????????????????????????  
=====      1   a   v           Joe      1234  
i=====      2   b   w           Gary     4321  
m=====      3   c   x           Jim      5432  
=====      4   d   y           Tom      5555  
f=====      5   e   z           Bill     4679  
d=====      6   f   0           Mary     9876  
=====      7   g   1           Sue      5683  
<<->> * * * End of File * * *
```

The results are that the first column of numbers and letters has been unchanged, but the operations have been applied to the column as follows:

```

Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
*** COLUMN COMMANDS PENDING ***      PF1=Help, PF2=List Commands, PF3=Exit

<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
=====   1   a   v           Joe      1234
=====   2   b   w           Gary     4321
=====   3   c   x
=====   4   d   y           Tom      5555
=====   5   e   z           Bill     4679
=====   6   f   0           Jim      5432
=====   7   g   1           Sue      5683
<<->> * * * End of File * * *

```

Column Exclude (—)

Instead of using the **Y** to mark those columns to be included, you may enter those columns which you want to exclude from XEDIT operations by using the **NOT** symbol — as follows:

```

      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
=====                                     ?????????????????????????????????????????

```

Use either **Y** or — for ease of entry, but not both at the same time.

XEDIT Commands Within a Column

The column include command **Y** or the column exclude command **—** also control the application of XEDIT commands and macros entered on the XEDIT command line and PF key operations in addition to controlling the application of XEDIT prefix commands. Once columns have been included or excluded then all XEDIT commands will now operate only on the selected columns. Any XEDIT macro that honors XEDIT zone settings and truncation settings may be used to manipulate the indicated column. The XEDIT commands **FILE** and **SAVE** will save the entire file regardless of COL's excluded columns or XEDIT's excluded lines. Consider the XEDIT **PUT** * command. The text located in the columns marked with **Y** and within the column range will be placed into the file named "fn ft fm."

Note: The columns must first be included or excluded by pressing **ENTER** prior to entering the XEDIT command. If they are not, then you are asking COL to apply the XEDIT command to lines that match the text as entered into the COL command line as explained in "XEDIT Commands" on page 108. In this case the text match would be "Y."

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
*** COLUMN COMMANDS PENDING ***      PF1=Help, PF2=List Commands, PF3=Exit

<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????????YYYYY????????????????????????????????
===== 1   a   v           Joe   1234
===== 2   b   w           Gary  4321
===== 4   d   y           Tom   5555
===== 5   e   z           Bill  4679
===== 6   f   0           Jim   5432
===== 7   g   1           Sue   5683
<<-> * * * End of File * * *

====> PUT * fn ft fm

                                         X E D I T   1 File
```

Also very convenient is the XEDIT command **PUTD** which places the text into a buffer or file exactly the same as the PUT command, but also deletes the text from the current file. When using PUTD with column include or exclude, the text within the included column will be deleted.

The XEDIT command **GET** will retrieve data from a PUT operation and place the results in the file and column marked with the **Y**. When used with column exclude, commands such as **GET**, will retrieve text so that the first column will be placed at the first **Y**. Normally you would specify a blank column to receive this text. If the column already contains text then the existing text will be pushed down the column just as lines are pushed down in normal XEDIT GET operations.

```
====> GET

                                         X E D I T   1 File
```

DELeTe * will delete the text in the marked column to the end of the column.

```
====> DELeTe *

                                         X E D I T   1 File
```

POWerinput may be used to place text into the marked column.

```
====> P OWerinp
```

```
X E D I T  1 File
```

XEDIT Targets When in COL column exclude operations, the XEDIT command executes from the top of the column. The equivalent current line within the column would correspond to line 0 when in XEDIT. You may use XEDIT targets to alter the line numbers within the column that your command is to operate on as follows:

Relative Displacement Target. An * means the bottom of the column range, -* means the top of the column range. If an XEDIT command or macro requires that the first line in the file be the current line, then precede the command or macro with the number one (1).: To delete the the entire column within the COL range you would enter the following:

```
====> DEL *
```

```
X E D I T  1 File
```

To delete the first line enter the following:

```
====> 1 DEL
```

```
X E D I T  1 File
```

To repeat any command, you must use the XEDIT linend character and include the repeat request in the command as follows:

```
====> 1 # CAppend text # REPEat *
```

```
X E D I T  1 File
```

Where # is the XEDIT linend character.

Absolute Line Number Target. Targets such as :10 may be used as targets in the XEDIT command or macro. Use the line number as displayed on the screen while in COL. COL will adjust the numbers for you since the numbers change in the work buffer when the COL command line is removed.

Named Line Target. Named line targets are supported without modification.

String Expression Target. The string target is searched only within the text located in the marked column.

Column Absolute Target. Column absolute targets are not supported.

PF Keys XEDIT commands assigned to PF keys such as SPLTJOIN will operate within the marked column.

XEDIT eXcluded Lines Excluded XEDIT lines are supported. Should the prefix or subcommand cause the text within a column to move up or down, the XEDIT excluded line will keep in step with its position in the included column.

Work Buffer

1. The column data marked with **Y** or not marked with **—** is placed in a work buffer. The XEDIT command or macro operates on the buffer. The buffer from the previous example using PUT * would be as follows.

```
*** Top of Range (Line=x) ***
|...>....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== Joe
===== Gary
===== Tom
===== Bill
===== Jim
===== Sue
*** End of Range (Line=x) ***
```

2. The work buffer sets zones and truncation column starting at column 1 and ending after the number of **Y**'s in the COL command line string.
3. When the XEDIT command or macro is applied to the work buffer, the current line in the buffer is the equivalent to the **Top Of File** or line 0. In fact it is Top Of Range.
4. If the cursor is located in the marked column, it is placed in the same relative position in the work buffer.

Multiple Columns Only included columns marked with **Y** (or not marked with **—**) are placed into the work buffer. The columns between included columns are not saved. When retrieving text from the buffer back into multiple columns, the text is spread over the included columns marked **Y** (or not marked with **—**). For example let us save the following two columns:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
*** COLUMN COMMANDS PENDING ***      PF1=Help, PF2=List Commands, PF3=Exit
```

```
<<-> *** Top of File ***
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ????????Y?????????????YYYYY????????????????????????????????
===== 1   a   v           Joe    1234
===== 2   b   w           Gary   4321
===== 4   d   y           Tom    5555
===== 5   e   z           Bill   4679
===== 6   f   0           Jim    5432
===== 7   g   1           Sue    5683
<<-> *** End of File ***
```

The work buffer contents would be as follows:

```

* * * Top of Range (Line=x) * * *
|...>...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== aJoe
===== bGary
===== dTom
===== eBill
===== fJim
===== gSue
* * * End of Range (Line=x) * * *
```

When using the XEDIT GET command, the text in the buffer or file being retrieved is placed sequentially from left to right under each included column. The following would GET the lines from “fn ft fm” and place the text from the first 3 columns into columns 19 through 21 and place the text from the next 5 columns into column 40 through 44.

```
Copy, Move, Xfr, Fol, Pre, Upon, Add, Ins, Del, †=Dup, Rep, Ovl, Emul_, Sort, G=Desceng, Zone
*** COLUMN COMMANDS PENDING *** PF1=Help, PF2=List Commands, PF3=Exit
```

```
<<-> * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????YYY?????????????????YYYYY????????????????????????????
===== 1 a v Joe 1234
===== 2 b w Gary 4321
===== 4 d y Tom 5555
===== 5 e z Bill 4679
===== 6 f 0 Jim 5432
<<-> * * * End of File * * *
```

```
=====> GET fn ft fm
```

X E D I T 1 File

Applying XEDIT Commands To Lines with Specified Text in a Column

You may apply an XEDIT prefix command or an XEDIT command to all lines that match the text that you enter into the COL command line. The text search is performed in the same column as the text in the COL command line.

XEDIT Prefix Commands

You may use COL to apply an XEDIT prefix command to all lines that either match, do not match, are greater than, are greater and equal to, are less than or are less than and equal to the text specified in the COL command line. The text match depends on the setting of XEDIT’s CASE setting. If IGNORE has been specified, then uppercase and lowercase representations of the same character will match. If RESPECT has been specified, then uppercase letters will not match a lowercase letter (and vice versa).

Let us delete all lines with blanks in columns 41 through 46 in the following example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,t=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
d===== ?????????????????????????????????????????????????????????????
===== Savickas, Derral                7249 D49 IBMVM1      06/01/89
===== Schuessler, Marvin                6150 9D5             06/02/89
===== Schultz, Sue                   6664 D25 IBMVM3      06/01/89
===== Schurr, Tomoko                  5234 D41             06/01/89
===== Schwind, Sandy                   3881 D40             06/02/89
<<->> * * * End of File * * *
```

The results are:

```
Prefix command D has been applied to 3 line(s) .
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== Savickas, Derral                7249 D49 IBMVM1      06/01/89
===== Schultz, Sue                   6664 D25 IBMVM3      06/01/89
<<->> * * * End of File * * *
```

Prefix Operators: Several operators are supported that may be entered in the COL prefix area to modify which lines are to receive the XEDIT prefix command or subcommand. These operators are listed in the following table:

Operator	Function
¬	not
<	less than
< =	less than or equal to
>	greater than
> =	greater than or equal to
*	search between XEDIT's current Zone settings rather than in the column where the text is entered.
@n	offset the application of the command by n lines.
@ + n	
@ - n	
+ n	
- n	

Table 2. COL Prefix Operators When Applying XEDIT Commands

The logical operators apply to numeric and alpha data. If the values are numeric (after commas and \$ signs are removed), then a numeric comparison is made. Otherwise a character by character alpha comparison is made.

Removing Blank Lines. To remove lines which are completely blank, enter a string of blanks. Make sure you enter enough blanks to uniquely identify your blank lines.

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
d====	

Note: In XEDIT, when you press ERASE EOF to erase an entire line, you must also enter at least one blank before you press ENTER or else that line will reappear. If you blank out the COL command line by pressing ERASE EOF, you must also enter at least one blank, or else COL will see the string of question marks that XEDIT has returned and determine that no text has been entered into the COL command line. COL will apply the prefix command “D”elete in the above example to the next line. Your next text line will be deleted.

The next section will show some examples of using prefixes with search commands.

Now let's see how you could insert 2 blank lines after each occurrence of **:dt.** in columns 1-3 in the following example:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
i2=== :dt.????????????????????????????????????????????????????????
===== :dl tsize=12.
===== :dt.lawn
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== :dt.gopher
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== :dt.agapanthus
===== :dd.A lovely flowering plant
===== the roots of which are the
===== preferred food of gophers.
===== :P.If your flourishing agapanthus suddenly keels over,
===== it means a gopher has had a
===== feast.
===== :edl.
<<-> * * * End of File * * *
```

The results are:

```
Prefix command I2 has been applied to 3 line(s) .
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<-> * * * Top of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????
===== :dl tsize=12.
===== :dt.lawn
=====
=====
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== :dt.gopher
=====
=====
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== :dt.agapanthus
=====
=====
===== :dd.A lovely flowering plant
```

You could insert two blank lines wherever there are matches in any column between XEDIT's current zone setting by using the * prefix operator as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
i2*== do????????????????????????????????????????????????????????????????
===== Do x=1 To cm.0
=====   dd1=Pos(cm.x,cmdltype)
=====   Do While dd1>0
=====     dd2=Pos(cm.x,cmdltype,dd1+2)
=====     If dd2>0 Then Do
=====       cmdline=O(¢ ¢,cmdline,dd2,2)
=====       cmdline=Overlay(dd2-dd1+2| |rexxblank,cmdline,dd1+1)
=====       Parse Upper Var cmdline cmdltype
=====       dd1=Pos(cm.x,cmdltype)
=====     End
=====     Else dd1=0
=====   End
===== End
<<->> * * * End of File * * *
```

The results are:

```
Prefix command I2  has been applied to 3 line(s) .
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== Do x=1 To cm.0
=====
=====
=====   dd1=Pos(cm.x,cmdltype)
=====   Do While dd1>0
=====
=====
=====     dd2=Pos(cm.x,cmdltype,dd1+2)
=====     If dd2>0 Then Do
=====
=====
=====       cmdline=O(¢ ¢,cmdline,dd2,2)
=====       cmdline=Overlay(dd2-dd1+2| |rexxblank,cmdline,dd1+1)
<<->> * * * End of File * * *
```

To duplicate two lines whenever a line with **&rbl.** in columns 1 - 5 is located you would enter the following:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
†2=== &rbl.????????????????????????????????????????????????????????????
===== On this date nothing happened.
===== &rbl.
===== But let us not have that happen again.
===== &rbl.
===== Make it so.
<<->> * * * End of File * * *
```

The results are:

```
Prefix command †2 has been applied to 2 line(s).
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== On this date nothing happened.
===== &rbl.
===== &rbl.
===== &rbl.
===== But let us not have that happen again.
===== &rbl.
===== &rbl.
===== &rbl.
===== Make it so.
<<->> * * * End of File * * *
```

If you were to exclude three lines starting with lines containing a 1 in column 1 you would enter the following:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
x3=== 1????????????????????????????????????????????????????????????????
===== 1Serial Numbers not in Phone Book
=====   STUDENT           DEPT
=====
=====   MITCHELL           E25
=====   BUSCH             98Z
=====   SODERSTROM        98Z
=====   CHAO
===== 1Serial Numbers not in Phone Book
=====   STUDENT           DEPT
=====
=====   HAULMAN           631
=====   LARSON            98Z
=====   MENDEZ            E37
=====   GAUTNEY           98Z
```

The results are:

```
Prefix command X3 has been applied to 2 line(s) .
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???????????????????????????????????????????????????????????????
===== ----- 3 line(s) not displayed -----
=====   MITCHELL           E25
=====   BUSCH             98Z
=====   SODERSTROM        98Z
=====   CHAO
===== ----- 3 line(s) not displayed -----
=====   HAULMAN           631
=====   LARSON            98Z
=====   MENDEZ            E37
=====   GAUTNEY           98Z
<<->> * * * End of File * * *
```

You could delete all lines that do **NOT** contain **abc** in columns 5-7 by entering the following:

```
| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7... |
| d  ???abc???????????????????????????????????????????????? |
```

To show or re-display lines that were excluded by the X prefix macro or the ALL command you could enter the following. Those lines that match the COL command line will be re-displayed.

```
| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7... |
| s  ???abc???????????????????????????????????????????????? |
```

You could exclude all lines that contain values less than 5000 in columns 5-11 by entering the following:

```
| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7... |
| x<  ??? 5000???????????????????????????????????????????? |
```

And you could show or re-display all lines that contain values greater than 5000 in columns 5-11 by entering the following:

```
| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7... |
| s>  ??? 5000???????????????????????????????????????????? |
```

You could exclude all lines that contain values less than or equal to 5000 in columns 5-11 by entering the following:

```
| ...+...1...+...2...+...3...+...4...+...5...+...6...+...7... |
| x<= ??? 5000???????????????????????????????????????????? |
```

Note: To use the = operator make sure that either the prefix area does not contain the = pad character by setting **NUM ON** or enter a blank after the = .

XEDIT Commands

Just as you may apply XEDIT prefix commands to lines that match the text entered in the COL command line, you may also apply XEDIT subcommands and user written macros to matching lines. The same operators may be entered in the COL command line prefix area to control the logic of the match. See Table 2 on page 101.

For example, consider the task of inserting a :dd. behind each :dt. within the current COL range:

```
<<->
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== :dt????????????????????????????????????????????????????????????
===== :dt.
===== :dt.<
===== :dt.<=
===== :dt.>
===== :dt.>=
===== :dt.*
===== :dt.@n
<<->

====> I :dd.

X E D I T  1 File
```

The results are:

```
<<->
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== :dt.
===== :dd.
===== :dt.<
===== :dd.
===== :dt.<=
===== :dd.
===== :dt.>
===== :dd.
===== :dt.>=
===== :dd.
===== :dt.*
===== :dd.
===== :dt.@n
===== :dd.
<<->
```

Multiple lines may be inserted by using XEDIT's linend character. For example, to insert a .cp and :h2. using the default linend symbol #, you would enter the following in the XEDIT command line:

```
====> I .cp # I :h2.

X E D I T  1 File
```

Other XEDIT commands that you may find useful in this mode include:

GET
PUT
CHANGE
DELETE

To use the operators listed in Table 2 on page 101 to control the logic of the match, enter the operator in the COL prefix area. For example, to GET a file whenever .cp is NOT found in column 1 - 3 you would enter:

```
|      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
|      |.cp????????????????????????????????????????????????????????  
|      |  
|====> GET fn ft fm  
|                                     X E D I T  1 File
```

Offsetting XEDIT Prefix and Subcommands

You may use XEDIT itself to offset any XEDIT command. You would accomplish this by prefixing the command with a relative target. While in COL you could insert a .cp above the text match by entering:

```
|====> -1 I .cp  
|                                     X E D I T  1 File
```

XEDIT prefix commands and XEDIT subcommands may be offset by using an @ (at) operator and an offset number in the COL Prefix Area. Consider inserting a **.cp** ABOVE specified lines using the XEDIT MASK setting and apply the XEDIT prefix command **I** which will be offset by one line. -1 is one line above the line containing the matching text.

First enter the following:

```
|====> set mask i .cp  
|                                     X E D I T  1 File
```

Then enter:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
i@-1= :h2????????????????????????????????????????????????????????????
===== :h2.
===== Text
===== :h2.
===== More text
===== :h2.
===== And more text
===== :h2.
===== Final Text
<<-> * * * End of File * * *
```

The results are:

```
<<-> * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== .cp
===== :h2.
===== Text
===== .cp
===== :h2.
===== More text
===== .cp
===== :h2.
===== And more text
===== .cp
===== :h2.
===== Final Text
<<-> * * * End of File * * *
```

Now you would enter the following to restore XEDIT's Mask setting:

```
====> set mask i

<<-> pending...
```

Note that negative displacements such as **@-1** may be abbreviated to **-1** to save space in the prefix area for other operators such as **I2** used with **↵**. For example an **I2-1↵** which uses all 5 available spaces in the prefix area. For symmetry **@1** may be entered as **+1**.

Delayed Execution of XEDIT Prefix Block Commands

Delayed Execution of XEDIT Prefix Block Commands is used when creating several sets of block commands, for example, deleting several blocks of text with identical starting and ending lines. A BookMaster :md. and :emd. is such an example. Using the methods of the previous section it can be seen that an XEDIT prefix of **DD** could be applied to specific lines and even offset if required. However you must delay the execution of the prefix command until you have included the ending of the blocks. In other words you need to wait until the second **DD** in each block has been entered.

When you add @ (with or without an offset value being entered) to a prefix block command, then execution of that prefix is delayed. DD@ and @DD are identical. @n may be used to move delayed block commands down or up where n is the number of lines to move down (up if n is negative).

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
|@dd== :md.????????????????????????????????????????????????????????????
```

This applies DD for delayed execution AT lines matching the text :md. The command will appear in the prefix area as @DD. Build as many sets of block commands as required before executing them. You would complete the block by entering the following:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
|@dd== :emd.???????????????????????????????????????????????????????????
```

The following would be displayed at the terminal after the above delayed operations have been entered.

```
=====
@DD== :md.
===== Lines containing text would be within this block.
===== In fact there are probably many lines between the limits.
@DD== :emd.
=====
===== Here is some text outside the block.
=====
@DD== :md.
===== Other lines with different text could be
===== in this block.
===== Even the number of lines may be different.
@DD== :emd.
=====
```

When you are ready to execute the delayed commands, enter a single @ in the COL command line prefix area.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
|@==== ??????????????????????????????????????????????????????????????
```

All delayed prefix commands will be executed.

This technique is not limited to Delete, but may be used for any block command such as eXclude, Move and Copy which require separate entry of the block's starting and ending points.

You may manually enter your own delayed prefix commands on any line by prefixing the command with @.

Should you change your mind you may remove all delayed commands by entering the XEDIT command **RESET** or by exiting COL.

Arbitrary Character

You may use your arbitrary character as defined within XEDIT when searching for a text match as you are applying XEDIT commands. For example if your arbitrary character were the XEDIT default character \$, you could cause an operation such as Insert to occur on every line as follows:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
I==== $????????????????????????????????????????????????????????
```

You may also search for text strings such as the following when applying an XEDIT command.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
I==== a$b??????????????????????????????????????????????????????
```

Here any number of characters will satisfy the condition just as when using XEDIT commands. For example consider the following command:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
I==== a$b$??????????????????????????????????????????????????????
```

Note that XEDIT will accept any number of arbitrary characters. This means that any of the following will match if located anywhere within columns 1 - 4 in the preceding example:

```
ab
ab
ab
axb
axb
axxb
```

This is different from using the COL command:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
I==== a?b??????????????????????????????????????????????????????
```

Here, the match will only occur when an "a" is found in column one and a "b" is found in column 4.

```
axxb
```

Using the * Operator to Search All Fields.

You may use the * prefix operator to request that all fields starting with the column where the text is entered and ending with the XEDIT zone setting (usually the end of the line) are to be searched for a text match. When using the XEDIT arbitrary character, any number of characters may exist between the entered text.

However if you use a COL command of the form **a??**, only text that contains the text with the specified number of intervening characters (as shown by imbedded command line symbol ?) will be located. For example:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...	
X*==	a\$b??	

will exclude text with any of the following forms:

ab
 axb
 axxb
 axxxb

The preceding text may be located anywhere within the line starting with column 1 (where the text was entered) and the end of the line (XEDIT'S zone setting):

On the other hand:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
X*== a?b????????????????????????????????????????????????
```

will only locate text of the form:

axxb

Again this text may occur anywhere between the columns defined by the text as entered into the COL command line and the end of the line (XEDIT's zone setting).

For example the following will search between columns 6 and the end of the line (or XEDIT's zone setting).

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
X*== ????a?b????????????????????????????????????????????
```

Part 2. Applications Using the COL macro

Part 2. This part of the XCOL User's Guide gives examples of how to accomplish specific tasks using COL.

Adding Blank Separator Lines

Add separator lines between the grouping of machine types.

```
3083
3083
3083
3084
3084
3090
```

Resulting in:

```
3083
3083
3083

3084
3084

3090
```

COL can accomplish this task in 4 or 5 commands as follows in macro form:

```
/**/
¢Locate :0 macro col * (____¢
¢Locate :0 macro col * blank @ prefix i@-1  (@@@¢¢¢
¢Locate :0 macro col * blank @ prefix x@-1  (@@@¢¢¢
¢Locate :0 macro col * (eeee¢
¢All¢
¢top¢
```

In command form the steps would be:

Steps:

1. "DeEmulate columns 1-4"

Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???
===== 3083
===== 3083
===== 3083
===== 3084
===== 3084
===== 3090

2. Insert a blank line in front(@-1) of lines that do not(¬) contain blanks in columns 1-4 (shown by 4 blanks in the command line)

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
i@-1  ?????????????????????????????????????????????????????????
===== 3083
=====
=====
===== 3084
=====
===== 3090

```

3. eXclude lines in front(@-1) of lines that do not(¬) contain blanks in columns 1-4 (shown by 4 blanks in the command line)

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
x@-1  ?????????????????????????????????????????????????????????
=====
===== 3083
=====
=====
=====
===== 3084
=====
=====
===== 3090

```

4. Put data back into columns 1-4

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== eeee????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== 3083
=====
=====
===== ----- 1 line(s) not displayed -----
===== 3084
=====
===== ----- 1 line(s) not displayed -----
===== 3090

```

5. Redisplay all lines

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
all== ?????????????????????????????????????????????????????????

```

The results are:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????
=====
===== 3083
===== 3083
===== 3083
=====
===== 3084
===== 3084
=====
===== 3090

```

Adding Lines With Specific Text

For example let us consider the following specific application.

Inserting Lines Containing ".cp"

Suppose we wish to add a conditional page eject before all lines containing "Internal Use Only" in columns 1 to 17. We start with:

```
==== Internal Use Only
==== .
==== .
==== .
==== Internal Use Only
==== .
==== .
==== .
```

We need to end with:

```
==== .cp
==== Internal Use Only
==== .
==== .
==== .
==== .cp
==== Internal Use Only
==== .
==== .
==== .
```

Using the XEDIT Input Command To perform a conditional insert, we would enter the text match into the COL command line and also enter the XEDIT command "I .cp" into the XEDIT command line, then press ENTER. Since we want to insert the ".cp" above the text match, we will precede the input command with the XEDIT relative target "-1" as follows:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
==== Internal Use Only????????????????????????????????????????????????????
```

```
====> -1 I .cp
```

```
X E D I T  1 File
```

Alternate Method Using the XEDIT Mask and Prefix Input Command: Set XEDIT's mask as follows on the XEDIT command line as follows:

```
====> set mask i .cp  
X E D I T  1 File
```

This sets all new input lines to contain a ".cp." Now use COL to insert new lines that match our search requirements using the "@" offset parameter as follows:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
I@-1= Internal Use Only????????????????????????????????????
```

which is a command to apply the prefix command "I" offset by -1, or one line above the text match.

Reset the XEDIT Mask as follows on the XEDIT command line.

```
====> set mask i  
X E D I T  1 File
```

Calculating Percent of Total

Consider the problem of calculating the percentage that each entry is of the total of all entries.

Before

```
1
2
3
4
5
```

After

```
1      6.7%
2     13.3%
3     20.0%
4     26.7%
5     33.3%
```

First obtain the total of the numbers. Use COL's sum command **+** with the string several columns to the right of the numbers so that the total will be to the right of the numbers:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????+++++++????????????????????????????????????????
      1
      2
      3
      4
      5
```

The results are:

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????
      1
      2
      3
      4
      5    15
```

Now copy the total backwards up the column using the emulate and reverse operator command:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????????eeeeee\????????????????????????????????????????????????
      1
      2
      3
      4
      5      15

```

The results are:

```

      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
      1      15
      2      15
      3      15
      4      15
      5      15

```

We are now ready to calculate the percentages using COL's math functions. Let us use one decimal point of accuracy:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```

      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??????? yy / yyy * 100 ==.= ?????????????????????????????????????
      1      15
      2      15
      3      15
      4      15
      5      15

```

The results are:

```

      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
      1      15      6.7
      2      15      13.3
      3      15      20.0
      4      15      26.7
      5      15      33.3

```

Using named columns the above command would have been:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ??????yyyy?yyyy?????==.=????????????????????????????????
      1      15

====> y.1 / y.2 * 100
```

The results again are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????
      1      15      6.7
```

Now we can remove the total column and add a percent sign as required:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????
      1      15      6.7
      2      15      13.3
      3      15      20.0
      4      15      26.7
      5      15      33.3

====> y.1 / y.2 * 100
```

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????
      1      6.7%
      2      13.3%
      3      20.0%
      4      26.7%
      5      33.3%
```

Create a Macro To Calculate Percentages

If the previous method of calculating percentages is too repetitive, you may want to build an XEDIT macro to do the task for you. You may create macros that call COL in two different ways. One is to sequentially call COL and the other is to stack commands for COL. Stacking commands is much faster.

Sequential Calls

```
/**/  
  
¢SET ZONE 7 *¢                               /* start the commands in column 7 */  
¢COL * (+++++¢  
¢COL * (???¢\¢  
¢COL * MATH {y.1 / y.2 * 100} (yyyy yyyyyy ==.=¢  
¢COL * (???¢ddddd¢??¢i%¢  
¢SET ZONE 1 *¢
```

Stacked Commands

```
/**/  
  
¢SET ZONE 7 *¢  
Queue ¢COLCMD +++++¢                          /* start the commands in column 7 */  
Queue ¢COLCMD ???¢\¢  
Queue ¢COLMAT yyyy yyyyyy ==.= {y.1 / y.2 * 100}¢  
Queue ¢COLCMD ???¢ddddd¢??¢i%¢  
Queue ¢ECOL¢  
¢MACRO COL * NOEXIT NOPROF¢  
¢SET ZONE 1 *¢
```

Note: For more information on creating Macros please refer to the IBM System Product Editor (XEDIT) manual.

Comparing Columns

You may use the REXX mathematical capability of COL to compare columns. Consider comparing the following two columns of data:

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
=====      a          a
=====      b          c
=====      c
=====      d          d
=====      e          e

=====>

X E D I T  1 File
```

You could enter the following:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???????yy????=????yy????=????????????????????????????????????
=====      a          a
=====      b          c
=====      c
=====      d          d
=====      e          e
```

The results are:

```
Invalid mathematical or logical operation ...
tc =†

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
=====      a          a          1
=====      b          c          0
=====      c          *
=====      d          d          1
=====      e          e          1
```

In the above example, a REXX syntax error has occurred on the third line which contains a blank field. To avoid this type of error when comparing character data which may have blank fields, enter quotes around the strings of y's. REXX will now make a test on the columns as literal data.

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====	?????cyyc???=???cyyc???=????????????????????????????????????
=====	a a
=====	b c
=====	c
=====	d d
=====	e e

The results are:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====	??
=====	a a 1
=====	b c 0
=====	c 0
=====	d d 1
=====	e e 1

Note that in the above examples “ d” and “d ” with leading and trailing blanks respectively were considered to be identical by REXX. To make an exact comparison use the REXX operator = = as follows:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====	?????cyyc???=???cyyc???=????????????????????????????????????
=====	a a
=====	b c
=====	c
=====	d d
=====	e e

The results are:

	...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====	??
=====	a a 1
=====	b c 0
=====	c 0
=====	d d 0
=====	e e 1

Converting To and From GML or BookMaster List Items

List

1. A gopher highway
can be identified by dinner-plate-size
mounds of dirt where grass used to be.
2. A gopher is a burrowing rodent that feeds on roots
of plants.
3. A lovely flowering plant
the roots of which are the
preferred food of gophers.
If your flourishing agapanthus suddenly keels over,
it means a gopher has had a
feast.

BookMaster or GML List

```
:li.A gopher highway  
can be identified by dinner-plate-size  
mounds of dirt where grass used to be.  
:li.A gopher is a burrowing rodent that feeds on roots  
of plants.  
:li.A lovely flowering plant  
the roots of which are the  
preferred food of gophers.  
If your flourishing agapanthus suddenly keels over,  
it means a gopher has had a  
feast.
```

Converting To GML Tags.

x====	??	1
=====	/r:li.??	2
=====	???d??? ???	3
all==	??	4
x====	:li??	5
=====	dddddd??	6
d====	??	7
all==	??	8

Steps:

1. Exclude lines with leading blanks
2. Replace the existing numbers with list tags
3. Delete the extra blank columns
4. Show all lines
5. Exclude lines with list tags
6. Delete the leading blanks
7. Delete blank lines
8. Show all lines

Converting From GML Tags.

x====	:li??	1
=====	/a7??	2
all==	??	3
i@-1=	:li??	4
x====	??	5
=====	ddd? ???	6
=====	/a7??	7
=====	rn??	8
=====	?r.??	9
all==	??	10

Steps:

1. Exclude lines that contain the list tags
2. Add 7 blank columns starting in column 1
3. Show all lines
4. Insert a blank line above list items.
5. Exclude lines with leading blanks
6. Delete 4 columns containing the list tag
7. Add 7 blank columns
8. Place numbers in column 1
9. Add a period behind the numbers
10. Show all lines

Counting Items in a List

How many of each part number are in the following list?

Before

12345678
99999999
12345678
87654321
12345678
67584930
87654321
12345678
90123456
87654321
99999999

After

12345678	4
67584930	1
87654321	3
90123456	1
99999999	2

COL Command Sequence

=====	ssssssss	??	1
=====	?????????r	1??	2
=====		??	3
=====	eeeeeeee?	nmn??	4
=====	s1ssssss?	g2g??	5
=====		??	6
d=====		??	7

Steps:

1. Group part numbers together by sorting
2. Insert seed number for counting purposes
3. Blank duplicate entries (including the above seed number)
4. Restore the part number and number (count) the sequence entries

At this point you have the following data. This might be the end if your requirements if you were trying to order or rank within each group of part numbers.

12345678	1
12345678	2
12345678	3
12345678	4
67584930	1
87654321	1
87654321	2
87654321	3
90123456	1
99999999	1
99999999	2

However you may continue with:

5. Sort the part numbers in ascending order and the count in descending order
6. Blank duplicate numbers in the list
7. Delete all lines with blanks in the first 8 columns

Creating an XEDIT Macro Using COL to Count Items in Lists

```
/* Example of how to use COL to count occurrences of */
/* duplicate information. */
/*
/* In this case report the number of part numbers which */
/* use a 8 digit part number in columns 1 to 8. */
/*
/* START WITH THE FOLLOWING WHICH BEGINS IN COLUMN 1: */
/* 12345678 */
/* AB999999 */
/* 12345678 */
/* 87654321 */
/* 12345678 */
/* 87654321 */
/* 12345678 */
/* 90123456 */
/* 87654321 */
/* AB999999 */
/*
/* and end up with: */
/* 12345678 4 */
/* 87654321 3 */
/* 90123456 1 */
/* AB999999 2 */
/*
/* where the number indicates the occurrence */
/* this may be accomplished with the following commands */
/* that are presented below in macro call format */

¢TOP¢
¢COL * NOPROF (ssssssss ¢ /* group part numbers together */
¢COL * NOPROF (???????r 1¢ /* insert seed number */
¢COL * NOPROF ( ¢ /* blank duplicate entries */
¢COL * NOPROF (eeeeeeee nnn¢ /* put back part number, but sequence */
/* each part number occurrence */
¢COL * NOPROF (slssssss g2g¢ /* group count in descending order */
/* within part number. */
¢COL * NOPROF ( ¢ /* blank duplicate part numbers */
¢COL * ? @ PREFIX D (@@@@@@¢ /* Set the blank character to @ and */
/* apply the XEDIT Prefix D to all */
/* lines with blanks in column 1-8 */
```

Alternate Macro that Stacks the COL Commands to Count Items in Lists

```
Queue ¢COLCMD sssssss ¢
Queue ¢COLCMD ????????r 1¢
Queue ¢COLCMD ¢
Queue ¢COLCMD eeeeeeee nnn¢
Queue ¢COLCMD slssssss g2gg¢
Queue ¢COLCMD ¢
Queue ¢COLPCM D @@@@@@@¢
Queue ¢ECOL¢
¢COL * ? @ NOEXIT NOPROF¢
```

Creating a GML/BookMaster Definition List From Column Data

Consider converting the following column data into a GML or BookMaster definition list. The “0” in column 1 is a carriage control character indicating a double space before printing that line. Since a GML definition list defaults to double spacing between definitions, we will have no use for the “0” when we are finished.

Before

```
0140: NO DECIMAL POINT IS GIVEN.
0141: THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES
      NOT KNOW TO WHICH VARIABLE YOU ARE REFERRING.
0142: LENGTH FOR NUMERIC VARIABLES IS: 2-8.
      LENGTH FOR CHARACTER VARIABLES NEEDS A $ SIGN.
      EXAMPLE: LENGTH NAME $ 22;
01421: NUMBER OF DECIMAL PLACES < WIDTH.
```

After

```
140      NO DECIMAL POINT IS GIVEN.
141      THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES NOT KNOW TO WHICH VARI-
      ABLE YOU ARE REFERRING.
142      LENGTH FOR NUMERIC VARIABLES IS: 2-8. LENGTH FOR CHARACTER VARIABLES NEEDS
      A $ SIGN. EXAMPLE: LENGTH NAME $ 22;
1421     NUMBER OF DECIMAL PLACES < WIDTH.
```

Steps:

1. Use the XEDIT SPLIT command to separate the codes in the first column and the text that starts in column 8. Start with the line containing “0140:”

```
====> SPLIT 8
```

```
X E D I T  1 File
```

2. Use the XEDIT REPEAT command to perform the same operation to the end of the list. If the list goes to the end of the file, we may use the XEDIT target “*” as follows:

```
====> REPEAT *
```

```
X E D I T  1 File
```

3. Apply COL to the resulting file.
4. Remove the blank lines within the file. This may be done by applying the XEDIT prefix command “D” to each line that contains blanks in the first several columns as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
d==== ?????????????????????????????????????????????????????
===== 0140:
===== NO DECIMAL POINT IS GIVEN.
===== 0141:
===== THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES
=====
===== NOT KNOW TO WHICH VARIABLE YOU ARE REFERRING.
===== 0142:
===== LENGTH FOR NUMERIC VARIABLES IS: 2-8.
=====
===== LENGTH FOR CHARACTER VARIABLES NEEDS A $ SIGN.
=====
===== EXAMPLE: LENGTH NAME $ 22;
===== 01421:
===== NUMBER OF DECIMAL PLACES < WIDTH.

```

- Now exclude all the lines that will not be converted to GML definition term tags. That is, display all of the codes that were originally in column 1-6. Since we have a "0" in column 1 for each of these code items we can use COL to apply the XEDIT prefix command "X" to eXclude lines that do NOT contain "0" in column 1 as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
x === 0????????????????????????????????????????????????????
===== 0140:
===== NO DECIMAL POINT IS GIVEN.
===== 0141:
===== THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES
===== NOT KNOW TO WHICH VARIABLE YOU ARE REFERRING.
===== 0142:
===== LENGTH FOR NUMERIC VARIABLES IS: 2-8.
===== LENGTH FOR CHARACTER VARIABLES NEEDS A $ SIGN.
===== EXAMPLE: LENGTH NAME $ 22;
===== 01421:
===== NUMBER OF DECIMAL PLACES < WIDTH.

```

The file is now displayed as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== 0????????????????????????????????????????????????????
===== 0140:
===== ----- 1 line(s) not displayed -----
===== 0141:
===== ----- 2 line(s) not displayed -----
===== 0142:
===== ----- 3 line(s) not displayed -----
===== 01421:
===== ----- 1 line(s) not displayed -----

```

- Convert these displayed lines to GML definition terms using the following COL command sequence:

```

===== d???????????????????????????????????????????????????? 1
===== /i:dt.???????????????????????????????????????????????? 2
===== -::::?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?:?: 3

```

Steps:

- a. Delete column 1
- b. Insert “:dt.” starting in column 1
- c. Remove “:”s behind the codes

The results are as follows:

```

===== |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== :dt.140
===== ----- 1 line(s) not displayed -----
===== :dt.141
===== ----- 2 line(s) not displayed -----
===== :dt.142
===== ----- 3 line(s) not displayed -----
===== :dt.1421
===== ----- 1 line(s) not displayed -----

```

7. Use COL to apply the XEDIT prefix command “S” to Show the line that follows the definition term line. We must do this because the next line needs the definition description tag added. We can use the offset “@” feature to accomplish this. We want to place the “S” AT 1 line from the matching text. In other words we will apply “S” to the next (@1) line following the lines with :dt. as follows:

```

===== | s@1== :dt.????????????????????????????????????????????????????????????????
=====

```

The results are:

```

===== |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????????
===== :dt.140
===== NO DECIMAL POINT IS GIVEN.
===== :dt.141
===== THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES
===== ----- 1 line(s) not displayed -----
===== :dt.142
===== LENGTH FOR NUMERIC VARIABLES IS: 2-8.
===== ----- 2 line(s) not displayed -----
===== :dt.1421
===== NUMBER OF DECIMAL PLACES < WIDTH.

```

8. Now eXclude the lines with the definition terms and then insert the definition description tag with the following two COL commands.

```

===== | x==== :dt.???????????????????????????????????????????????????????????????? | 1
===== | /i:dd.???????????????????????????????????????????????????????????????? | 2
=====

```

Steps:

- a. eXclude lines with this text
- b. Insert “:dd.” starting in column 1

The results are:

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????
==== ----- 1 line(s) not displayed -----
==== :dd.NO DECIMAL POINT IS GIVEN.
==== ----- 1 line(s) not displayed -----
==== :dd.THIS ABBREVIATION IS AMBIGUOUS. THE COMPILER DOES
==== ----- 2 line(s) not displayed -----
==== :dd.LENGTH FOR NUMERIC VARIABLES IS: 2-8.
==== ----- 3 line(s) not displayed -----
==== :dd.NUMBER OF DECIMAL PLACES < WIDTH.
```

9. Apply the XEDIT command ALL and exit COL to redisplay all lines.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????????????????????????????????????????????????????
```

Creating Column Data from Non-Column Text

Let us place the initials in the following list of names before the last name and remove the first name.

Before

Barber, (CA) Chuck	841 001-3 x-5584 BARBER	AT BLDVM1
Barr, (DU) Diran	827 001-3 x-4219 BARR	AT BLDVM3
Bemis, (BB) Barry	826 001-3 x-7840 BEMIS	AT BLDVM3
Bending, (LE) Larry	849 001-3 x-5940 BENDING	AT BLDVM1
Bettge, (JR) Joni	826 001-3 x-7664 BETTGE	AT BLDVM3
Bogott-Palinkx, (DC) Donna	844 001-3 x-6509 BOGOTT	AT BLDVM3
Bolander, (D) Denny	846 001-3 x-3220 BOLANDER	AT BLDVM3

After

C.A. Barber	841 001-3 x-5584 BARBER	AT BLDVM1
D.U. Barr	827 001-3 x-4219 BARR	AT BLDVM3
B.B. Bemis	826 001-3 x-7840 BEMIS	AT BLDVM3
L.E. Bending	849 001-3 x-5940 BENDING	AT BLDVM1
J.R. Bettge	826 001-3 x-7664 BETTGE	AT BLDVM3
D.C. Bogott-Palinkx	844 001-3 x-6509 BOGOTT	AT BLDVM3
D. Bolander	846 001-3 x-3220 BOLANDER	AT BLDVM3

COL Command Sequence

```
===== ?????????????????q, ????????????????????????????????????????????? 1
===== ?????????????????????dd ????????????????????????????????????????? 2
===== ?????????????????????q) ????????????????????????????????????????? 3
===== ?????????????????????????dd ????????????????????????????????????? 4
===== ?????????????????????i. ????????????????????????????????????????? 5
x===== ?????????????????????? ????????????????????????????????????????? 6
===== ?????????????????????i. ????????????????????????????????????????? 7
all= ????????????????????????????????????????????????????????????????? 8
===== p ?????????????????????mmmm ????????????????????????????????????? 9
===== ??????????????????????jl ????????????????????????????????????? 10
```

Steps:

1. Queue or line up the “,” thereby placing the initials in a column. The column selected is not important except that it be behind the longest last name in the list. Do not panic because the right columns are no longer aligned. The last step will make everything line up as before.
2. Delete the “,” and the “(.”
3. Queue or line up the “).”
4. Delete the “)” and first names.
5. Insert a “.” after the first initial.
6. Exclude lines with no middle initial. These lines have blanks in that column.
7. Insert a “.” after the middle initial.
8. Show all lines.
9. Move the initials along with one blank preceding the last name.
10. Left Justify the ragged right columns.

Filling in Blanks or Removing Redundant Text

Blank Columns

10/12/88	844	\$3,983
		3,094
		2,105
739		1,216
		3,327
		2,438

Filled Columns

10/12/88	844	\$3,983
10/12/88	844	3,094
10/12/88	844	2,105
10/12/88	739	1,216
10/12/88	739	3,327
10/12/88	739	2,438

Fill In the Blanks.

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
=====	eeeeeeee??e??e??	

Blank the Duplicates.

	...+....1....+....2....+....3....+....4....+....5....+....6....+....7...	
=====	_____??_??	

Flowing Text Within a Column

Before

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???????????<????????????????????????????????????????????????<??????
===== 1.    A gopher highway
=====      can be identified by dinner-plate-size
=====      mounds of dirt where grass used to be.
=====
===== 2.    A gopher is a burrowing rodent that feeds on roots
=====      of plants.
=====
===== 3.    A lovely flowering plant
=====      the roots of which are the
=====      preferred food of gophers.
=====      If your flourishing agapanthus suddenly keels over,
=====      it means a gopher has had a
=====      feast.
<<->>
```

After

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???????????<????????????????????????????????????????????????<??????
===== 1.    A gopher highway can be identified by dinner-plate-size
=====      mounds of dirt where grass used to be.
=====
=====
===== 2.    A gopher is a burrowing rodent that feeds on roots of
=====      plants.
=====
=====
===== 3.    A lovely flowering plant the roots of which are the
=====      preferred food of gophers. If your flourishing
=====      agapanthus suddenly keels over, it means a gopher has
=====      had a feast.
=====
=====
<<->>
```

Editing GML or BookMaster Table Columns

Consider editing GML or BookMaster Tables.

Title	Name	Date
Manager	A. Smith	1947
Employee	G. Vair	1984

The table might look as follows in your text file. However you might want to work with a single column. The following command will allow that operation.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
all== :c 1.????????????????????????????????????????????????????????????
===== :rdef id=row
=====  concat = no
===== cwidths= ç* * *ç
===== concat = no
===== :table refid=row
===== :thd refid=row
===== :c 1.Title
===== :c 2.Name
===== :c 3.Date
===== :ethd.
===== :row.
===== :c 1.Manager
===== :c 2.A. Smith
===== :c 3.1947
===== :row.
===== :c 1.Employee
===== :c 2.G. Vair
===== :c 3.1984
===== :row.
===== :etable.
```

The results are:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== ----- 5 line(s) not displayed -----
===== :c 1.Title
===== ----- 4 line(s) not displayed -----
===== :c 1.Manager
===== ----- 3 line(s) not displayed -----
===== :c 1.Employee
===== ----- 4 line(s) not displayed -----
```

Marking All Entries With a Value within Ranges.

Suppose you want to flag all entries with a salary within a range of \$10,000 and \$100,000. You also wish to count the number of occurrences.

Before

221	2	984,999
223	11	9,678
311	8	768,000
312	10	535,980
433	8	88,888
433	9	324,555
542	6	432,920
572	6	42,033
684	5	123,000
739	7	10,000
831	3	44,254
913	4	429,321

After

221	2	984,999	
223	11	9,678	
311	8	768,000	
312	10	535,980	
433	8	88,888	*
433	9	324,555	
542	6	432,920	
572	6	42,033	*
684	5	123,000	
739	7	10,000	*
831	3	44,254	*
913	4	429,321	

COL Command Sequence

x>	??	100,000????????????????????????????????????	1
x<	??	10,000????????????????????????????????????	2
====	??r*	????????????????????????????????????	3
====	??mn	????????????????????????????????????	4
all==	??	????????????????????????????????????	5

Steps:

1. Exclude all values greater than 100,000. The leading blanks in the COL command line are required to define the width of the column being searched. Make sure the largest number is found. You may use the COL prefix command PICTu to show the maximum width of text in a column.
2. Exclude all values less than 10,000
3. Flag all remaining entries with a "*."
4. Number the entries.
5. Show all lines.

Making One Column From Two

Before

C.A. Barber	841 001-3	D. Bolander	846 001-3
D.U. Barr	827 001-3	D.L. Brighton	846 001-3
B.B. Bemis	826 001-3	D.H. Brookins	844 001-3
L.E. Bending	849 001-3	J.L. Brozovich	827 001-3
J.R. Bettge	826 001-3	J.N. Chartier	A73 001-3
D.C. Bogott-Palinkx	844 001-3	D.L. Clapp	6W6 001-3

After

C.A. Barber	841 001-3
D.U. Barr	827 001-3
B.B. Bemis	826 001-3
L.E. Bending	849 001-3
J.R. Bettge	826 001-3
D.C. Bogott-Palinkx	844 001-3
D. Bolander	846 001-3
D.L. Brighton	846 001-3
D.H. Brookins	844 001-3
J.L. Brozovich	827 001-3
J.N. Chartier	A73 001-3
D.L. Clapp	6W6 001-3

Using XEDIT Put and Get Please see “Transferring Column Text With PUT and GET” on page 168 for using excluded columns and XEDIT’s PUT and GET commands.

Using Column Input and Delete Commands

COL Command Sequence The following sequences picture the XEDIT screen for each step. Include the columns to be moved down by entering

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????y*????????????????????????????
===== C.A. Barber      841 001-3      D. Bolander      846 001-3
===== D.U. Barr        827 001-3      D.L. Brighton    846 001-3
===== B.B. Bemis       826 001-3      D.H. Brookins    844 001-3
===== L.E. Bending     849 001-3      J.L. Brozovich   827 001-3
===== J.R. Bettge      826 001-3      J.N. Chartier    A73 001-3
===== D.C. Bogott-Palinkx 844 001-3      D.L. Clapp       6W6 001-3
<<->>
```

Now insert blank lines in the marked column. Here six lines are inserted because of the six names in the column

```
<<->
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
i6=== ??????????????????????????????????YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
===== C.A. Barber          841 001-3      D. Bolander           846 001-3
===== D.U. Barr            827 001-3      D.L. Brighton             846 001-3
===== B.B. Bemis           826 001-3      D.H. Brookins             844 001-3
===== L.E. Bending         849 001-3      J.L. Brozovich            827 001-3
===== J.R. Bettge          826 001-3      J.N. Chartier              A73 001-3
===== D.C. Bogott-Palinkx   844 001-3      D.L. Clapp                 6W6 001-3
<<->
```

Reinvoke COL on the bottom half of the column to delete the blank column to the left.

```

===== C.A. Barber                841 001-3
===== D.U. Barr                  827 001-3
===== B.B. Bemis                 826 001-3
===== L.E. Bending               849 001-3
===== J.R. Bettge                826 001-3
<<->> D.C. Bogott-Palinkx      844 001-3
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== d35????????????????????????????????????????????????????????????
=====
                                D. Bolander      846 001-3
=====
                                D.L. Brighton    846 001-3
=====
                                D.H. Brookins    844 001-3
=====
                                J.L. Brozovich    827 001-3
=====
                                J.N. Chartier    A73 001-3
=====
                                D.L. Clapp       6W6 001-3
<<->>

```

Results after exiting COL.

=====	C.A. Barber	841	001-3
=====	D.U. Barr	827	001-3
=====	B.B. Bemis	826	001-3
=====	L.E. Bending	849	001-3
=====	J.R. Bettge	826	001-3
=====	D.C. Bogott-Palinkx	844	001-3
=====	D. Bolander	846	001-3
=====	D.L. Brighton	846	001-3
=====	D.H. Brookins	844	001-3
=====	J.L. Brozovich	827	001-3
=====	J.N. Chartier	A73	001-3
=====	D.L. Clapp	6W6	001-3

Making Two Columns From One

Before

C.A. Barber	841 001-3
D.U. Barr	827 001-3
B.B. Bemis	826 001-3
L.E. Bending	849 001-3
J.R. Bettge	826 001-3
D.C. Bogott-Palinkx	844 001-3
D. Bolander	846 001-3
D.L. Brighton	846 001-3
D.H. Brookins	844 001-3
J.L. Brozovich	827 001-3
J.N. Chartier	A73 001-3
D.L. Clapp	6W6 001-3

After

C.A. Barber	841 001-3	D. Bolander	846 001-3
D.U. Barr	827 001-3	D.L. Brighton	846 001-3
B.B. Bemis	826 001-3	D.H. Brookins	844 001-3
L.E. Bending	849 001-3	J.L. Brozovich	827 001-3
J.R. Bettge	826 001-3	J.N. Chartier	A73 001-3
D.C. Bogott-Palinkx	844 001-3	D.L. Clapp	6W6 001-3

Using XEDIT Put and Get Commands Please see “Transferring Column Text With PUT and GET” on page 168 for using excluded columns and XEDIT’s PUT and GET commands.

Using Column Input and Delete Commands

COL Command Sequence The following sequences picture the XEDIT screen for each step. Start with COL range covering the last half of the list of names.

```
===== C.A. Barber      841 001-3
===== D.U. Barr       827 001-3
===== B.B. Bemis      826 001-3
===== L.E. Bending    849 001-3
===== J.R. Bettge     826 001-3
<<->> D.C. Bogott-Palinkx 844 001-3
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????????
===== D. Bolander     846 001-3
===== D.L. Brighton   846 001-3
===== D.H. Brookins   844 001-3
===== J.L. Brozovich   827 001-3
===== J.N. Chartier    A73 001-3
===== D.L. Clapp      6W6 001-3
<<->>
```

Shift this last half right 35 columns with the command “/a35”

```

===== C.A. Barber          841 001-3
===== D.U. Barr            827 001-3
===== B.B. Bemis           826 001-3
===== L.E. Bending         849 001-3
===== J.R. Bettge          826 001-3
<<->> D.C. Bogott-Palinkx  844 001-3
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== /a35????????????????????????????????????????????????????????????
===== D. Bolander          846 001-3
===== D.L. Brighton        846 001-3
===== D.H. Brookins        844 001-3
===== J.L. Brozovich        827 001-3
===== J.N. Chartier         A73 001-3
===== D.L. Clapp           6W6 001-3
<<->>

```

Reinvoke COL with a range covering the entire list of names including the names which were shifted. Include the column with the shifted names by entering “Y*” at column 36.

```

<<->>
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????y*????????????????????????
===== C.A. Barber          841 001-3
===== D.U. Barr            827 001-3
===== B.B. Bemis           826 001-3
===== L.E. Bending         849 001-3
===== J.R. Bettge          826 001-3
===== D.C. Bogott-Palinkx  844 001-3
=====
===== D. Bolander          846 001-3
===== D.L. Brighton        846 001-3
===== D.H. Brookins        844 001-3
===== J.L. Brozovich        827 001-3
===== J.N. Chartier         A73 001-3
===== D.L. Clapp           6W6 001-3
<<->>

```

Now apply the XEDIT block Delete command.

```

<<->>
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????YYYYYYYYYYYYYYYYYYYYYYYY
dd=== C.A. Barber          841 001-3
===== D.U. Barr            827 001-3
===== B.B. Bemis           826 001-3
===== L.E. Bending         849 001-3
===== J.R. Bettge          826 001-3
dd=== D.C. Bogott-Palinkx  844 001-3
=====
===== D. Bolander          846 001-3
===== D.L. Brighton        846 001-3
===== D.H. Brookins        844 001-3
===== J.L. Brozovich        827 001-3
===== J.N. Chartier         A73 001-3
===== D.L. Clapp           6W6 001-3
<<->>

```

Results:

```
<<->>
|...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
===== ?????????????????????????????????????????????????????????????
===== C.A. Barber          841 001-3      D. Bolander           846 001-3
===== D.U. Barr            827 001-3      D.L. Brighton           846 001-3
===== B.B. Bemis           826 001-3      D.H. Brookins           844 001-3
===== L.E. Bending         849 001-3      J.L. Brozovich          827 001-3
===== J.R. Bettge          826 001-3      J.N. Chartier           A73 001-3
===== D.C. Bogott-Palinkx  844 001-3      D.L. Clapp              6W6 001-3
=====
=====
=====
=====
=====
=====
<<->>
```

Marking Duplicate Text in a File

The first step for marking duplicate data is to blank the duplicate data in the file. This may be accomplished by using the de-emulate “_” command once the data has been sorted. For example consider marking lines with duplicate dates in the following:

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ??? ?????????????????????????????????????????????????????????
===== 1 10/20/85 549
===== 2 10/21/85 456
===== 3 10/21/85 864
===== 4 10/22/85 549
===== 5 10/23/85 982
<<->>
```

Now that there is a blank column within the file, these lines that contain the blank column may be displayed and then marked. Exclude all lines that do not have blanks in the 8 columns containing the date here.

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
x  ??? ?????????????????????????????????????????????????????????
===== 1 10/20/85 549
===== 2 10/21/85 456
===== 3          864
===== 4 10/22/85 549
===== 5 10/23/85 982
<<->>
```

The results are:

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????
===== 2 ----- 2 line(s) not displayed -----
===== 3          864
===== 5 ----- 2 line(s) not displayed -----
<<->>
```

Optional Step To Show First Occurrence You may also want to mark the first duplicate line. The following COL prefix command shows lines located “at” 1 line above all lines wherever blanks are found in columns 3 through 11.

```

<<->
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
s@-1= ???              ?????????????????????????????????????????????????????????
===== 2 ----- 2 line(s) not displayed -----
===== 3              864
===== 5 ----- 2 line(s) not displayed -----
<<->

```

The results are:

```

===== ----- 1 line(s) not displayed -----
===== 2 10/21/85    456
===== 3              864
===== ----- 2 line(s) not displayed -----

```

Mark the Lines Being Displayed. These are the lines that originally had the duplicate data. Insert your mark into a column.

```

<<-> * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????????????i*????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== 2 10/21/85    456      (not displayed if the optional step is skipped)
===== 3              864
===== ----- 2 line(s) not displayed -----
<<-> * * * End of File * * *

```

Show all lines.

```

all== ?????????????????????????????????????????????????????????????????????????

```

Replace the text that was removed in step 1 by emulating the text in columns 3 through 11 as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???eeeeeeee????????????????????????????????????????????????????????????

```

The results with duplicates marked is:

```

===== 1 10/20/85    549
===== 2 10/21/85    456 *      (not marked if the optional step is skipped)
===== 3 10/21/85    864 *
===== 4 10/22/85    549
===== 5 10/23/85    982

```

Rearranging Blocks of Text Such as ":H1." Sections

Suppose it becomes necessary to rearrange many large blocks of text. This example uses :H1. tags as an example. Our document may contain several thousand lines of text and perhaps a hundred heading tags. Our chapters need to be rearranged as follows:

Chapters Before Arranging

Schedule

Management Overview

Exposures

Checkpoints

Chapters After Arranging

Management Overview

Schedule

Checkpoints

Exposures

The source code for the above may be:

Before

```
:H1.Schedule
Pages of schedule information.
More pages and pages of schedule information.
Lots of text.
:H1.Management Overview
A few paragraphs for management.
:H1.Exposures
A couple of pages listing the very few exposures expected.
:H1.Checkpoints
A detailed list of checkpoints.
In fact there are quite a few lines here.
```

After

```
:H1.Management Overview
A few paragraphs for management.
:H1.Schedule
Pages schedule information.
More pages and pages of schedule information.
Lots of text.
:H1.Checkpoints
A detailed list of checkpoints.
In fact there are quite a few lines here.
:H1.Exposures
A couple of pages listing the very few exposures expected.
```

COL Command Sequence Invoke COL to cover the entire body of the document. Now issue the following COL command sequence:

```
===== /a5???????????????????????????????????????????????????????????????????????? 1
X == ?????:H1.???????????????????????????????????????????????????????????????????? 2
```

Steps:

1. Add 5 blank spaces before every line.
2. Exclude all lines that do not contain the :H1. Tag.

The terminal display will now appear as follows.

```
===== ?????????????????????????????????????????????????????????????????????????
=====      :H1.Schedule
===== ----- 3 line(s) not displayed -----
=====      :H1.Management Overview
===== ----- 1 line(s) not displayed -----
=====      :H1.Exposures
===== ----- 1 line(s) not displayed -----
=====      :H1.Checkpoints
===== ----- 3 line(s) not displayed -----
```

Now manually place sequence numbers before each :H1. tag. This will show its final position in the file.

```
===== ?????????????????????????????????????????????????????????????????????????
===== 2      :H1.Schedule
===== ----- 3 line(s) not displayed -----
===== 1      :H1.Management Overview
===== ----- 1 line(s) not displayed -----
===== 4      :H1.Exposures
===== ----- 1 line(s) not displayed -----
===== 3      :H1.Checkpoints
===== ----- 3 line(s) not displayed -----
```

Now continue with the following COL command sequence:

```
===== ALL== ????????????????????????????????????????????????????????????????????? 1
===== ee???????????????????????????????????????????????????????????????????????? 2
===== ss???????????????????????????????????????????????????????????????????????? 3
===== d5???????????????????????????????????????????????????????????????????????? 4
```

Steps:

1. Show all lines.
2. Emulate or copy forward the sequence numbers that you have placed before each :H1. tag.
3. Sort the file using your sequence numbers.
4. Delete the first 5 columns which are no longer needed.

Reformatting Tables

Before

California	
AAK	Los Angeles
ABL	San Diego
ACK	San Francisco
ADM	Buena Vista
AEN	Hollywood
AFO	Oakland
AGP	Anaheim
AHQ	Orange Grove
AIR	Death Valley
AJS	Port Humene
Colorado	
BAB	Denver
BBK	Boulder
BCZ	Colorado Springs
BDY	Yuma
BEX	Fort Garland
BFW	Burlington
BGV	Durango
BHU	Steamboat Springs
BIT	Divide
BJS	Cripple Creek

After

"AAK" = "California"
"ABL" = "California"
"ACK" = "California"
"ADM" = "California"
"AEN" = "California"
"AFO" = "California"
"AGP" = "California"
"AHQ" = "California"
"AIR" = "California"
"AJS" = "California"
"BAB" = "Colorado"
"BBK" = "Colorado"
"BCZ" = "Colorado"
"BDY" = "Colorado"
"BEX" = "Colorado"
"BFW" = "Colorado"
"BGV" = "Colorado"
"BHU" = "Colorado"
"BIT" = "Colorado"
"BJS" = "Colorado"

COL Command Sequence

x ==	??	1
====	????????????????d*??	2
all==	??	3
x====	??	4
====	/a10??	5
all==	??	6
====	?????????????????eeeeeeeeeeeeeeee??	7
====	o" " = " "??	8

Steps:

1. Exclude all lines NOT having 2 blanks starting in column 1
2. Delete unnecessary text, here the city name
3. Show all lines
4. Exclude all lines having 2 blanks starting in column 1
5. Add 10 blank columns
6. Show all lines
7. Copy forward state names
8. Overlay required text

Using normal XEDIT commands, delete any extraneous lines.

Another After Conversion

"AAK " = "Los Angeles"
 "ABL " = "San Diego"
 "ACK " = "San Francisco"
 "ADM " = "Buena Vista"
 "AEN " = "Hollywood"
 "AFO " = "Oakland"
 "AGP " = "Anaheim"
 "AHQ " = "Orange Grove"
 "AIR " = "Death Valley"
 "AJS " = "Port Humene"
 "BAB " = "Denver"
 "BBK " = "Boulder"
 "BCZ " = "Colorado Springs"
 "BDY " = "Yuma"
 "BEX " = "Fort Garland"
 "BFW " = "Burlington"
 "BGV " = "Durango"
 "BHU " = "Steamboat Springs"
 "BIT " = "Divide"
 "BJS " = "Cripple Creek"

COL Command Sequence

[illegible]

Steps:

1. Overlay required text
2. Remove trailing blanks to pull in the " to the text
3. Remove extra blank columns

Using normal XEDIT commands, delete any extraneous lines.

Removing Blank Lines From a File

We will use COL to apply the XEDIT prefix command “D” to all lines that contain blanks in specific columns. Enter a string of blanks in the COL command line along with a “D” in the adjacent prefix area.

Before

```
<<->>
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
d==== ???      ?????????????????????????????????????????????
===== 1 10/20/85  549
===== 2          456
===== 3          864
===== 4 10/21/85  549
===== 5          982
<<->>
```

After

```
<<->>
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????
===== 1 10/20/85  549
===== 4 10/21/85  549
<<->>
```

If you want to remove lines which are totally blank, enter a string of blanks. Make sure you enter enough blanks to uniquely identify your blank lines.

```
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
d====
```

Note: In XEDIT, when you press ERASE EOF to erase an entire line, you must also enter at least one blank before you press ENTER otherwise that line will reappear. If you blank out the COL command line by pressing ERASE EOF, you must also enter at least one blank. Otherwise, COL will see the string of question marks that XEDIT has returned and determine that no text has been entered into the COL command line. COL will apply the prefix command “D”elete in the above example to the next line. Your next text line will be deleted.

Removing Duplicate Text From a File

The first step for removing duplicate data is to blank the duplicate data in the file. You may then delete the lines with blank columns. This may be accomplished by using the de-emulate “_” command. For example consider removing lines with duplicate dates:

Before

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ???_????????????????????????????????????????????????????????
===== 1 10/20/85 549
===== 2 10/20/85 456
===== 3 10/20/85 864
===== 4 10/21/85 549
===== 5 10/21/85 982
<<->>
```

After

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????????????????????????
===== 1 10/20/85 549
===== 2
===== 3
===== 4 10/21/85 549
===== 5
===== 5 982
<<->>
```

Sorting Blocks of Text Such as :H1. Sections

Suppose it becomes necessary to sort many large blocks of text. This example uses :H1. tags as an example of this type of sort technique. Our document may contain several thousand lines of text and perhaps a hundred heading tags. Our chapters need to be rearranged alphabetically as follows:

Chapters Before Sorting

Schedule

Management Overview

Exposures

Checkpoints

Chapters After Sorting

Checkpoints

Exposures

Management Overview

Schedule

In fact, some of the tags may even include extra information such as an id. The source code for the above may be as follows:

Before

```
:H1.Schedule
Pages of schedule information.
More pages and pages of schedule information.
Lots of text.
:H1 id=mo.Management Overview
A few paragraphs for management.
:H1.Exposures
A couple of pages listing the very few exposures expected.
:H1.Checkpoints
A detailed list of checkpoints.
In fact there are quite a few lines here.
```

After

```
:H1.Checkpoints
A detailed list of checkpoints.
In fact there are quite a few lines here.
:H1.Exposures
A couple of pages listing the very few exposures expected.
:H1 id=mo.Management Overview
A few paragraphs for management.
:H1.Schedule
Pages of schedule information.
More pages and pages of schedule information.
Lots of text.
```

COL Command Sequence Invoke COL to cover the body of the document. Now issue the following COL command to exclude all lines that contain the :H1. tag.

```
| X==== :H1.????????????????????????????????????????????????????????????????? |
```

Now shift this text to the right 35 spaces as follows:

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== /a35?????????????????????????????????????????????????????????????????
==== ----- 1 line(s) not displayed -----
==== Pages of schedule information.
==== More pages and pages of schedule information.
==== Lots of text.
==== ----- 1 line(s) not displayed -----
==== A few paragraphs for management.
==== ----- 1 line(s) not displayed -----
==== A couple of pages listing the very few exposures expected.
==== ----- 1 line(s) not displayed -----
==== A detailed list of checkpoints.
==== In fact there are quite a few lines here.
==== ----- 12 line(s) not displayed -----
<<->>
```

Show all lines:

```
| ALL== ?????????????????????????????????????????????????????????????????? |
```

In our example some header tags contained additional information, an id attribute. We therefore need to align all of the :H1. titles using the “Q” command. We will use a period as our alignment character. If your header tags do not contain any additional information then this step may be bypassed.

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ?????????q.?????????????????????????????????????????????????????????
==== :H1.Schedule
====
==== Pages of schedule information.
==== More pages and pages of schedule infor
==== Lots of text.
==== :H1 id=mo.Management Overview
====
==== A few paragraphs for management.
==== :H1.Exposures
====
==== A couple of pages listing the very few
==== :H1.Checkpoints
====
==== A detailed list of checkpoints.
==== In fact there are quite a few lines he
<<->>
```

The resulting file now has the text aligned in columns 11 through 35. We can now emulate or copy forward the headings as follows:

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ??????????eeeeeeeeeeeeeeeeeeeeeeeeeeee? ?????????????????????????????????
===== :H1          .Schedule

=====                                     Pages of schedule information.
=====                                     More pages and pages of schedule infor
=====                                     Lots of text.

===== :H1 id=mo.Management Overview

=====                                     A few paragraphs for management.

===== :H1          .Exposures

=====                                     A couple of pages listing the very few

===== :H1          .Checkpoints

=====                                     A detailed list of checkpoints.
=====                                     In fact there are quite a few lines he
<<->>
```

[illegible]

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
X :H1.????????????????????????????????????????????????????????
===== :H1      .Checkpoints
=====      Checkpoints      A detailed list of checkpoints.
=====      Checkpoints      In fact there are quite a few lines he
===== :H1      .Exposures
=====      Exposures      A couple of pages listing the very few
===== :H1 id=mo.Management Overview
=====      Management Overview      A few paragraphs for management.
===== :H1      .Schedule
=====      Schedule      Pages of schedule information.
=====      Schedule      More pages and pages of schedule infor
=====      Schedule      Lots of text.
<<->>
```

Now remove the trailing blanks up to column 9 by entering a “B” in column 9 as follows:

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ????????b????????????????????????????????????????????????????????????
===== :H1          .Checkpoints
===== ----- 2 line(s) not displayed -----
===== :H1          .Exposures
===== ----- 1 line(s) not displayed -----
===== :H1 id=mo.Management Overview
===== ----- 1 line(s) not displayed -----
===== :H1          .Schedule
===== ----- 3 line(s) not displayed -----
<<->>
```

Redisplay all lines as follows:

[illegible]

Exclude the :H1. tags so that we may remove the extra text from the remaining lines as follows:

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
X :H1.????????????????????????????????????????????????????????????
==== :H1.Checkpoints
====      Checkpoints      A detailed list of checkpoints.
====      Checkpoints      In fact there are quite a few lines here
==== :H1.Exposures
====      Exposures      A couple of pages listing the very few
==== :H1 id=mo.Management Overview
====      Management Overview      A few paragraphs for management.
==== :H1.Schedule
====      Schedule      Pages of schedule information.
====      Schedule      More pages and pages of schedule information
====      Schedule      Lots of text.
<<->>
```

Now delete the 35 columns that we have added as follows:

```

<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== d35????????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
=====          Checkpoints          A detailed list of checkpoints.
=====          Checkpoints          In fact there are quite a few lines he
===== ----- 1 line(s) not displayed -----
=====          Exposures            A couple of pages listing the very few
===== ----- 1 line(s) not displayed -----
=====          Management Overview  A few paragraphs for management.
===== ----- 1 line(s) not displayed -----
=====          Schedule             Pages of schedule information.
=====          Schedule             More pages and pages of schedule infor
=====          Schedule             Lots of text.
<<->>

```

Redisplay all lines as follows and exit COL.

```

| ALL== ?????????????????????????????????????????????????????????????? |

```

The blocks are sorted and appear as follows:

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????????????????????????????????
===== :H1.Checkpoints
===== A detailed list of checkpoints.
===== In fact there are quite a few lines here.
===== :H1.Exposures
===== A couple of pages listing the very few exposures expected.
===== :H1 id=no.Management Overview
===== A few paragraphs for management.
===== :H1.Schedule
===== Pages of schedule information.
===== More pages and pages of schedule information.
===== Lots of text.

```

Sorting Columns

Let us sort the middle column leaving the text in the other two columns in place.

Before

	T. Smith	
	A. Smith	B. Good
G. Vair	M. Krieger	D. Day
A. Class	B. Kaser	G. Night

After

	B. Kaser	
	M. Krieger	B. Good
G. Vair	A. Smith	D. Day
A. Class	T. Smith	G. Night

COL Command

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...|
==== ??????????z?s2s1sssssss?z????????????????????????????|
```

The text between the “z”s will be sorted. The column marked by “s1sss” (last names) is sorted first and the column marked by “s2” (initials) is sorted second.

Sorting Dates

Often the dates entered into a file are not in an order that may be easily sorted. The following will show how to sort dates that have been entered in mm/dd/yy format. The text will be sorted first by year, then by month and last by date.

Before

10/12/88	844	\$3,983
09/21/88	844	3,094
11/07/88	844	2,105
08/16/88	739	1,216
10/07/88	739	3,327
07/26/88	739	2,438

After

07/26/88	739	2,438
08/16/88	739	1,216
09/21/88	844	3,094
10/07/88	739	3,327
10/12/88	844	\$3,983
11/07/88	844	2,105

COL Command

```

===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...

```

Sorting Lists

Lists such as VM nicknames or BookMaster/GML

- Simple Lists
- Ordered Lists
- Unordered Lists
- Definition Lists
- Glossary Lists
- Table rows

may be sorted using the technique shown in the following example. The example shows how to sort a definition list. When sorting lists which use list items, use `:li.` rather than `:dt` as shown in the example for the definition list. Use `:gt` for glossary lists and `:name` for VM nickname lists. If you are sorting table row entries based on cell 1, make sure the tag is entered as `:c 1.` and then use `:c 1` in place of the `:dt.`

When using this technique for lists within imbedded lists which use the same tags, for example a definition list within a definition list or a list item within a list which also uses list items, then the imbedded list must first be removed and saved in another file prior to the sort. An `.im` may be used to retrieve the nested list or the list may be recovered using XEDIT's GET command after the sort has been accomplished.

Before

TERM	DESCRIPTION
lawn	Gopher Highway. Can be identified by dinner-plate-size mounds of dirt where grass used to be.
gopher	A burrowing rodent that feeds on roots of plants.
agapanthus	A lovely flowering plant the roots of which are the preferred food of gophers. If your flourishing agapanthus suddenly keels over, it means a gopher has had a feast.

After

TERM	DESCRIPTION
agapanthus	A lovely flowering plant the roots of which are the preferred food of gophers. If your flourishing agapanthus suddenly keels over, it means a gopher has had a feast.
gopher	A burrowing rodent that feeds on roots of plants.
lawn	Gopher Highway. Can be identified by dinner-plate-size mounds of dirt where grass used to be.

COL Command Sequence

1. On the XEDIT command line issue `"COL /:edl./"` with the first `:dt.` tag as the current line. If you have set up prefix commands using `"HH,"` then enter the following:

```

===== :dl tsize=12.
===== :dthd.TERM
===== > :ddhd.DESCRPTION
HH=== :dt.lawn
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== :dt.gopher
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== :dt.agapanthus
===== :dd.A lovely flowering plant
===== the roots of which are the
===== preferred food of gophers.
===== :P.If your flourishing agapanthus suddenly keels over,
===== it means a gopher has had a
HH=== feast.
===== :edl.

```

2. Exclude the :dt. tags so that all lines that do NOT contain the “sort” tag are displayed.

```

===== :dl tsize=12.
===== :dthd.TERM
<<-> :ddhd.DESCRPTION
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
x===== :dt.????????????????????????????????????????????????????????????
===== :dt.lawn
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== :dt.gopher
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== :dt.agapanthus
===== :dd.A lovely flowering plant
===== the roots of which are the
===== preferred food of gophers.
===== :P.If your flourishing agapanthus suddenly keels over,
===== it means a gopher has had a
===== feast.
<<-> :edl.

```

3. Starting in column 1, enter /a20 in the COL command line where “20” is a large enough number to shift the text past the longest :dt. tag text.

```

===== :dl tsize=12.
===== :dthd.TERM
<<->> :ddhd.DESCRPTION
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== /a20????????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== ----- 1 line(s) not displayed -----
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== ----- 1 line(s) not displayed -----
===== :dd.A lovely flowering plant
===== the roots of which are the
===== preferred food of gophers.
===== :P.If your flourishing agapanthus suddenly keels over,
===== it means a gopher has had a
===== feast.
<<->> :edl.

```

4. Show all lines.

```

===== :dl tsize=12.
===== :dthd.TERM
<<->> :ddhd.DESCRPTION
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
all== ?????????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
=====          :dd.Gopher Highway.
=====          :p.Can be identified by dinner-plate-size
=====          mounds of dirt where grass used to be.
===== ----- 1 line(s) not displayed -----
=====          :dd.A burrowing rodent that feeds on roots
=====          of plants.
===== ----- 1 line(s) not displayed -----
=====          :dd.A lovely flowering plant
=====          the roots of which are the
=====          preferred food of gophers.
=====          :P.If your flourishing agapanthus suddenly keels over,
=====          it means a gopher has had a
=====          feast.
<<->> :edl.

```

5. Emulate (copy forward) the text in column 5 - 20 by entering the “E” operator.

```
===== :dl tsize=12.
===== :dthd.TERM
<<-> :ddhd.DESCRPTION
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???seeeeeeeeeeeeeee????????????????????????????????????????
===== :dt.lawn
=====                               :dd.Gopher Highway.
=====                               :p.Can be identified by dinner-plate-size
=====                               mounds of dirt where grass used to be.
===== :dt.gopher
=====                               :dd.A burrowing rodent that feeds on roots
=====                               of plants.
===== :dt.agapanthus
=====                               :dd.A lovely flowering plant
=====                               the roots of which are the
=====                               preferred food of gophers.
=====                               :P.If your flourishing agapanthus suddenly keels over,
=====                               it means a gopher has had a
=====                               feast.
<<-> :edl.
```

6. Sort the text by placing the “S” operator in columns 5 through 20.

```
===== :dl tsize=12.
===== :dthd.TERM
<<-> :ddhd.DESCRPTION
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ???ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
===== :dt.lawn
=====   lawn                       :dd.Gopher Highway.
=====   lawn                       :p.Can be identified by dinner-plate-size
=====   lawn                       mounds of dirt where grass used to be.
===== :dt.gopher
=====   gopher                     :dd.A burrowing rodent that feeds on roots
=====   gopher                     of plants.
===== :dt.agapanthus
=====   agapanthus                 :dd.A lovely flowering plant
=====   agapanthus                 the roots of which are the
=====   agapanthus                 preferred food of gophers.
=====   agapanthus                 :P.If your flourishing agapanthus suddenly keels over,
=====   agapanthus                 it means a gopher has had a
=====   agapanthus                 feast.
<<-> :edl.
```

7. Exclude the :dt. tags so that all lines that do NOT contain the “sort” tag are displayed.

```

===== :dl tsize=12.
===== :dthd.TERM
<<-> :ddhd.DESCRPTION
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
x===== :dt.????????????????????????????????????????????????????????????
===== :dt.agapanthus
===== agapanthus :dd.A lovely flowering plant
===== agapanthus the roots of which are the
===== agapanthus preferred food of gophers.
===== agapanthus :P.If your flourishing agapanthus suddenly keels over,
===== agapanthus it means a gopher has had a
===== agapanthus feast.
===== :dt.gopher
===== gopher :dd.A burrowing rodent that feeds on roots
===== gopher of plants.
===== :dt.lawn
===== lawn :dd.Gopher Highway.
===== lawn :p.Can be identified by dinner-plate-size
===== lawn mounds of dirt where grass used to be.
<<-> :edl.

```

8. Delete columns 1 - 20 by placing the “D” operator in those columns.

```

===== :dl tsize=12.
===== :dthd.TERM
<<-> :ddhd.DESCRPTION
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== d20????????????????????????????????????????????????????????????
===== ----- 1 line(s) not displayed -----
===== agapanthus :dd.A lovely flowering plant
===== agapanthus the roots of which are the
===== agapanthus preferred food of gophers.
===== agapanthus :P.If your flourishing agapanthus suddenly keels over,
===== agapanthus it means a gopher has had a
===== agapanthus feast.
===== ----- 1 line(s) not displayed -----
===== gopher :dd.A burrowing rodent that feeds on roots
===== gopher of plants.
===== ----- 1 line(s) not displayed -----
===== lawn :dd.Gopher Highway.
===== lawn :p.Can be identified by dinner-plate-size
===== lawn mounds of dirt where grass used to be.
<<-> :edl.

```

9. Show All lines and press PF3.

```

| ALL== ?????????????????????????????????????????????????????????????????? |

```

Results:

```
===== :dl tsize=12.
===== :dthd.TERM
===== :ddhd.DESCRPTION
===== :dt.agapanthus
===== :dd.A lovely flowering plant
===== the roots of which are the
===== preferred food of gophers.
===== :P.If your flourishing agapanthus suddenly keels over,
===== it means a gopher has had a
===== feast.
===== :dt.gopher
===== :dd.A burrowing rodent that feeds on roots
===== of plants.
===== :dt.lawn
===== :dd.Gopher Highway.
===== :p.Can be identified by dinner-plate-size
===== mounds of dirt where grass used to be.
===== :edl.
```

Sorting Text With Imbedded Headers

Before Sort the salary in descending order leaving the headers and blank lines in place during the sort.

Department Number	Employees in Dept.	Total Yearly Salary
221	2	984,999
223	11	9,678
311	8	768,000
312	10	535,980

Department Number	Employees in Dept.	Total Yearly Salary
433	8	88,888
433	9	324,555
542	6	432,920
572	6	42,033

Department Number	Employees in Dept.	Total Yearly Salary
684	5	123,000
739	7	10,000
831	3	44,254
913	4	429,321

After

Department Number	Employees in Dept.	Total Yearly Salary
221	2	984,999
311	8	768,000
312	10	535,980
542	6	432,920

Department Number	Employees in Dept.	Total Yearly Salary
913	4	429,321
433	9	324,555
684	5	123,000
433	8	88,888

Department Number	Employees in Dept.	Total Yearly Salary
831	3	44,254
572	6	42,033
739	7	10,000
223	11	9,678

COL Command Sequence

x===	??	1
x2===	department??	2
=====	????????????????????????????????gggggggg????????????????????????????????	3
all==	??	4

Steps:

- 1. Exclude blank lines
- 2. Exclude header lines
- 3. Sort Salary in descendinG order
- 4. Show all lines

When using COL to perform the XEDIT Sort commands, excluded lines are honored. Without COL, the XEDIT sort would sort all lines, whether included or excluded.

Transferring Column Text With PUT and GET

Any XEDIT command may be applied to a column when using COL include “Y” or exclude “_”. For example you may use XEDIT’s PUT and GET to transfer column text within a file or from one file to another in the same manner that lines are normally transferred. Consider the problem of making two columns from one. We will then consider the process of making one column from two.

One Column

C.A. Barber	841 001-3
D.U. Barr	827 001-3
B.B. Bemis	826 001-3
L.E. Bending	849 001-3
J.R. Bettge	826 001-3
D.C. Bogott-Palinkx	844 001-3
D. Bolander	846 001-3
D.L. Brighton	846 001-3
D.H. Brookins	844 001-3
J.L. Brozovich	827 001-3
J.N. Chartier	A73 001-3
D.L. Clapp	6W6 001-3

Two Columns

C.A. Barber	841 001-3	D. Bolander	846 001-3
D.U. Barr	827 001-3	D.L. Brighton	846 001-3
B.B. Bemis	826 001-3	D.H. Brookins	844 001-3
L.E. Bending	849 001-3	J.L. Brozovich	827 001-3
J.R. Bettge	826 001-3	J.N. Chartier	A73 001-3
D.C. Bogott-Palinkx	844 001-3	D.L. Clapp	6W6 001-3

Make Two Columns From One Issue the XEDIT command “PUT” to save or “PUTD” to save and remove the text. Start with the middle of the column as your current line as follows:

```
===== C.A. Barber      841 001-3
===== D.U. Barr        827 001-3
===== B.B. Bemis       826 001-3
===== L.E. Bending     849 001-3
===== J.R. Bettge      826 001-3
===== D.C. Bogott-Palinkx 844 001-3
===== D. Bolander      846 001-3
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== D.L. Brighton    846 001-3
===== D.H. Brookins    844 001-3
===== J.L. Brozovich    827 001-3
===== J.N. Chartier     A73 001-3
===== D.L. Clapp        6W6 001-3
```

```
=====> PUTD *
```

```
X E D I T  1 File
```

Note: If the text to be transferred does not start in column 1 or if the entire line is not to be transferred then enter COL and include (or exclude) the columns to be transferred. Press ENTER then enter the PUT(D) command as follows:

```

===== C.A. Barber      841 001-3
===== D.U. Barr       827 001-3
===== B.B. Bemis      826 001-3
===== L.E. Bending    849 001-3
===== J.R. Bettge     826 001-3
<<->> D.C. Bogott-Palinkx 844 001-3
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== YYYYYYYYYYYYYYYYYYYYYYYYYYYY????????????????????????????????????
===== D. Bolander     846 001-3
===== D.L. Brighton   846 001-3
===== D.H. Brookins   844 001-3
===== J.L. Brozovich   827 001-3
===== J.N. Chartier   A73 001-3
===== D.L. Clapp      6W6 001-3
<<->>

=====> PUTD *

X E D I T  1 File

```

Invoke COL with a range covering the names in the first half of the list. Include the column to receive the names by entering “Y” above the columns. We can use the space available to the end of the line by:

```

<<->>
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????y*????????????????????????????
===== C.A. Barber      841 001-3
===== D.U. Barr       827 001-3
===== B.B. Bemis      826 001-3
===== L.E. Bending    849 001-3
===== J.R. Bettge     826 001-3
===== D.C. Bogott-Palinkx 844 001-3
<<->>

```

Press ENTER then issue the XEDIT GET command:

```

=====> GET

X E D I T  1 File

```

The results are:

```

<<->>
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== ?????????????????????????????????????YYYYYYYYYYYYYYYYYYYYYYYYYYYYY
===== C.A. Barber      841 001-3      D. Bolander      846 001-3
===== D.U. Barr       827 001-3      D.L. Brighton    846 001-3
===== B.B. Bemis      826 001-3      D.H. Brookins   844 001-3
===== L.E. Bending    849 001-3      J.L. Brozovich   827 001-3
===== J.R. Bettge     826 001-3      J.N. Chartier   A73 001-3
===== D.C. Bogott-Palinkx 844 001-3    D.L. Clapp      6W6 001-3
<<->>

```

Make One Column From Two Using the same text as above, invoke COL on your column text and mark the columns you are transferring with Y*, which shows that we are transferring the column that continues to the end of the line.

```
<<->>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ?????????????????????????????????Y*????????????????????????????????
===== C.A. Barber      841 001-3      D.  Bolander      846 001-3
===== D.U. Barr        827 001-3      D.L. Brighton     846 001-3
===== B.B. Bemis       826 001-3      D.H. Brookins     844 001-3
===== L.E. Bending     849 001-3      J.L. Brozovich    827 001-3
===== J.R. Bettge      826 001-3      J.N. Chartier     A73 001-3
===== D.C. Bogott-Palinkx 844 001-3      D.L. Clapp        6W6 001-3
<<->>
```

Press ENTER and then issue the XEDIT PUT command. PUTD * could be used to save the data and to delete it from the column. The XEDIT target "*" shows that the entire column is to be saved.

```
====> PUTD *

X E D I T  1 File
```

Exit COL and make the last line in the left column your current line. You may now execute the XEDIT command GET as follows:

```
===== C.A. Barber      841 001-3
===== D.U. Barr        827 001-3
===== B.B. Bemis       826 001-3
===== L.E. Bending     849 001-3
===== J.R. Bettge      826 001-3
===== D.C. Bogott-Palinkx 844 001-3
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...

====> GET

X E D I T  1 File
```

The results are:

```
===== C.A. Barber      841 001-3
===== D.U. Barr        827 001-3
===== B.B. Bemis       826 001-3
===== L.E. Bending     849 001-3
===== J.R. Bettge      826 001-3
===== D.C. Bogott-Palinkx 844 001-3
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== D.  Bolander     846 001-3
===== D.L. Brighton    846 001-3
===== D.H. Brookins    844 001-3
===== J.L. Brozovich   827 001-3
===== J.N. Chartier    A73 001-3
===== D.L. Clapp       6W6 001-3
```

Working With Columns Independently

The following examples illustrate how XCOL can be used to: Insert, Delete, Move, Copy, Duplicate, etc., text in a column.

Before

L. Cobb	A73 001-3	D.R. Galloway	840 001-3
J.N. Chartier	A73 001-3	J. Crone	849 001-3
J.L. Brozovich	827 001-3	A.L. Davidson	846 001-3
D.J. Compton	985 001-3	K.A. Dahle	985 001-3
D. Bolander	846 001-3	J.R. Degner	846 001-3
D.C. Bogott-Palinkx	844 001-3	J.M. Devins	843 001-3
J.R. Bettge	826 001-3	K.T. Gambon	825 001-3
L.E. Bending	849 001-3	G.D. Dillon	826 001-3
D.H. Brookins	844 001-3	W.H. Dillon	849 001-3
D.L. Clapp	6W6 001-3	P.A. Douglas	36H 001-3

After

L. Cobb	A73 001-3	D.R. Galloway	840 001-3
J.N. Chartier	A73 001-3	J. Crone	849 001-3
		A.L. Davidson	846 001-3
J.L. Brozovich	827 001-3	K.A. Dahle	985 001-3
D.J. Compton	985 001-3	J.R. Degner	846 001-3
D. Bolander	846 001-3	J.M. Devins	843 001-3
D.C. Bogott-Palinkx	844 001-3	K.T. Gambon	825 001-3
J.R. Bettge	826 001-3	G.D. Dillon	826 001-3
L.E. Bending	849 001-3	W.H. Dillon	849 001-3
D.H. Brookins	844 001-3	P.A. Douglas	36H 001-3
D.L. Clapp	6W6 001-3		

COL Command Sequence

```
Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
PF1=Hlp/2=CMDlist/3=Exit <>W=Workflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
```

```
<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
===== L. Cobb A73 001-3 D.R. Galloway 840 001-3
I J.N. Chartier A73 001-3 J. Crone 849 001-3
===== J.L. Brozovich 827 001-3 A.L. Davidson 846 001-3
===== D.J. Compton 985 001-3 K.A. Dahle 985 001-3
===== D. Bolander 846 001-3 J.R. Degner 846 001-3
===== D.C. Bogott-Palinkx 844 001-3 J.M. Devins 843 001-3
===== J.R. Bettge 826 001-3 K.T. Gambon 825 001-3
===== L.E. Bending 849 001-3 G.D. Dillon 826 001-3
===== D.H. Brookins 844 001-3 W.H. Dillon 849 001-3
===== D.L. Clapp 6W6 001-3 P.A. Douglas 36H 001-3
<<->>
```

The results are:

```

<<->>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
===== L. Cobb A73 001-3 D.R. Galloway 840 001-3
===== J.N. Chartier A73 001-3 J. Crone 849 001-3
===== A.L. Davidson 846 001-3
===== J.L. Brozovich 827 001-3 K.A. Dahle 985 001-3
===== D.J. Compton 985 001-3 J.R. Degner 846 001-3
===== D. Bolander 846 001-3 J.M. Devins 843 001-3
===== D.C. Bogott-Palinkx 844 001-3 K.T. Gambon 825 001-3
===== J.R. Bettge 826 001-3 G.D. Dillon 826 001-3
===== L.E. Bending 849 001-3 W.H. Dillon 849 001-3
===== D.H. Brookins 844 001-3 P.A. Douglas 36H 001-3
===== D.L. Clapp 6W6 001-3
<<->>

```

Now apply any XEDIT prefix command such as “I” to insert a blank. XEDIT prefix commands and command line commands will take effect only on the included lines.

COL Macro

The format of the COL macro is:

```
COL
  [target|1]
  [ S [ B ]]
  [PROFILE fn]
  [NOPROF]
  [UNDO]
  [NOUNDO]
  [NOEXIT]
  [PREFIX x]
  [ATABx]
  [TABx]
  [COMPx]
  [TSn]
  [PERIOD]
  [COMMA]
  [FRENCH]
  [MATH [{expression}]]
  [NOINFOx]

  [(column_command

ECOL
```

Figure 6. COL Command. and optional ECOL to exit COL

Examples: COL * invokes COL to end of file

COL :5 * \$ invokes down to but not including line 5
with the command line symbol set to *
and the blank symbol set to \$

COL * (???ddd executes COL with the command:
???ddd???...

target is a normal XEDIT line target (e.g: *,:12,/uptohere/...) and marks the range within your file where COL will execute. COL always starts at the current line and ends (not including) on the target line.

S is the symbol used in the COL command line. The default is the character ?. The character used as the symbol may not be placed in the text by any of the commands.

B is the symbol used in the command line to show that a blank character is to be inserted. COL sets the blank character to an actual blank, but this may be altered in PROFCOL XEDIT.

column_command

is the command that COL will execute. This is for macro writers who want to build the command line and have COL execute that command line rather than have a user input the command line. The current setting of XEDIT's zones is used to apply this command. Therefore if the zone setting is 10 *, then the command will apply starting in column 10.

The following keywords may be entered anywhere in the COL argument string:

PROFILE <i>fn</i>	Name of the alternate profile. Default name is PROFCOL.
NOPROF	Do not execute the COL profile PROFCOL XEDIT.
UNDO	Start the undo option which allows undoing the results of the last COL command. Undo is also a COL command line prefix command. Internal setting for the undo option is NOUNDO. PROFCOL XEDIT may override this default. When <i>column_command</i> is used, Undo is always set to NOUNDO regardless of the profile setting unless UNDO is specified. Undo uses XEDIT's PUT and GET commands. System response will be improved when undo is not active.
NOUNDO	Turn off the undo option regardless of the setting in PROFCOL XEDIT.
NOEXIT	When using a macro call to COL, do not exit COL after completing the argument string.
PREFIX <i>x</i>	Apply the XEDIT prefix command as shown by <i>x</i> to all lines that match the text in the <i>column_command</i> . For example: COL * PREFIX D (will delete all blank lines as defined by the current XEDIT zone setting.
ATABx	Where <i>x</i> is the SCRIPT tab control character. For example, your SCRIPT text may have tabs defined as follows: .ti 05 .tp 10 20 30 04 811 Most 111.00 9583 The ATAB function will automatically tab or align the text based on the length of the longest text in each field. Fields are defined by the SCRIPT tab control character which is a <code>␣</code> in the above example. "ATAB <code>␣</code> " will use the character <code>␣</code> to automatically tab the text. The XEDIT tab settings are adjusted to match the text.
TABx	Where <i>x</i> is the SCRIPT tab control character. This is the same as ATABx except that the text will be tabbed to the XEDIT tab settings rather than automatically tabbed based on text length of each field.
COMPx	Where <i>x</i> is the SCRIPT tab control character. The Compress operation is the reverse of ATAB or TAB. The data will be recompressed into its SCRIPT format. The character <i>x</i> will be reinserted as the field divider and control character.
TSn	The space control for ATAB is set by this parameter. The internal default for COL is to place 2 spaces between each field when auto tabbing. The <i>n</i> is the number of spaces for this call and will override the default 2. TS may also be defined in PROFCOL XEDIT as tab_space = n .
PERIOD	Use a period to show decimals in numbers (1,234,567.89) when using the SUM command + + regardless of the setting in PROFCOL XEDIT.
COMMA	Use a comma to show decimals in numbers (1.234.567,89) when using the SUM command + + regardless of the setting in PROFCOL XEDIT.
FRENCH	Use a period or comma to show decimals and blanks for thousand separators in numbers (1 234 567.89 or 1 234 567,89) when using the SUM command + + regardless of the setting in PROFCOL XEDIT.
MATH [{ <i>expression</i> }]	Forces mathematical evaluation of the COL command line. When named columns are used, then { <i>expression</i> } is used to evaluate the columns.

ECOL

Exit COL.

```
Queue ¢ECOL¢                               /* exit COL                */
```

other

Any text stacked without the above prefix operators will be discarded.

An example of a complete command sequence within a REXX macro might be as follows.

```
Queue ¢COLCMD ??YYY¢           /* exclude all but columns 3 - 5    */
Queue ¢COLXED GET¢ fn ft fm /* get text from fn ft fm and place */
                               /* into the non excluded columns */
Queue ¢ECOL¢                   /* exit COL                        */
¢MACRO COL * NOEXIT NOPROF¢ /* execute COL                      */
```

XEDIT Interactions

Command Priority

COL commands are processed left to right. XEDIT treats the COL command line as a line within the file during editing. The COL command line priority is the same as a Screen Change priority as documented in the XEDIT Reference Manual. Other Screen changes and XEDIT Prefix commands are processed before COL commands using the default XEDIT settings. An exception of course would be when using COL to apply XEDIT prefix commands or subcommands to lines that match the column text as entered into the COL command line. This is accomplished by entering text into the COL command line and entering either an XEDIT prefix command in the COL prefix area or an XEDIT subcommand on the command line. see "Applying XEDIT Commands To Lines with Specified Text in a Column" on page 100. In this case the COL command line is processed in its turn. The requested command is applied to matching lines top to bottom. The XEDIT prefix command or the XEDIT subcommand requested is then processed for the matching lines. XEDIT and COL commands entered simultaneously are interpreted in this manner.

XEDIT Ring

Support is provided for XEDIT Ring. When in a COL edit session, if you leave to tour the ring, you will find the prefix commands "HH" or your own synonym pending when you return to your session. Just press ENTER to re-establish your COL session. You may still UNDO, recall (?) or re-execute (=) your last COL command in the COL prefix area. If you do not want to re-establish the COL session, enter RESET on the XEDIT command line. You may use COL in any of the files as you tour the ring.

XEDIT Multiple Logical Screens

Limited support is provided for multiple logical screens. COL may be active in only one logical screen at a time. Unpredictable results may occur if COL and XEDIT commands are entered on multiple logical screens simultaneously.

Moving the Range Markers

The lines containing the range markers "< < - > >" are named lines within XEDIT. Moving these lines will change the COL range to the new location of the markers.

COL Messages

A separate file named COL MESSAGES contains all of the COL error and help messages. You may customize this file to suit your needs.

Several lines within this file contain %'s within the text. These are used by COL to insert variable information or screen control characters. If you decide to modify this file, keep the % in the correct relative position.

The two lines of information that COL provides are defined by the lines labeled COL001 through COL003. If you blank the text behind these labels, you will remove that information from the COL screen. However, do keep the label in the file.

```
COL001 Copy,Move,Xfr,Fol,Pre,Upon,Add,Ins,Del,†=Dup,Rep,Ovl,Emul_,Sort,G=Desceng,Zone
COL002 %PF1=Hlp/2=CMDlist/3=Exit%<>W=Wordflow,+=Sum,Low,K=Up,Just,Blk,Nbr,/=Col,Que,Y
COL003 %*** COLUMN COMMANDS PENDING ***%   PF1 = Help, PF2 = List Commands, PF3 = Exit
```

The quick reference provided when pressing PF2 is defined by the lines labeled COL901 and greater. You may add or remove lines to suit your needs. The quick reference for the COL commands is:

```
COL901 ***** COLUMN Quick Reference *****
COL902 /      Shift screen to this column      | K      convert column to UPPER case
COL903 /cmd Place txt there† not behind cmd    | L      CONVERT COLUMN TO lower CASE
COL904 ‡txt Pad (overlay) blanks with text     | M      Move columns
COL905 -txt Replace text with blanks           | N      Number (insert sequence nbrs)
COL906 ;      Set tabs                         | Otxt   Overlay text
COL907 †      Duplicate                       | P      Preceding target
COL908 +      Sum numbers in the column        | Qtxt   Line up text (Queue-up text)
COL909 \      Reverse the command or column    | QQtxt  Line up text behind character
COL910 A      Add (insert) a number of blanks  | Rtxt   Replace text
COL911 B      Remove trailing Blanks to this column | S      Sort
COL912 C      Copy columns                    | T      Truncation setting
COL913 D      Delete columns                  | U      Upon target
COL914 E      Emulate (copy forward) data      | V      Verify setting
COL915 _      De-emulate (blank duplicates)    | W      Flow words just left and right
COL916 F      Following target                 | <      Flow words justifying left
COL917 G      descending sort                  | >      Flow words justifying right
COL918 H      Verify in HEX                    | X      Xfer columns
COL919 Itxt Insert text                       | Y      Column include(inverse of † †)
COL920 J      Justify (use with JL, JC, JR, or JB) |      Column exclude
COL921      Add nbr (i.e. JL1) to specify spacing | Z      Zone setting
```

The quick reference for the COL prefix commands is:

```
COL922 ***** Prefix Commands adjacent to COL command line *****
COL923 ?      Re-display last command          | TRANs  Transpose columns and lines
COL924 =      Re-execute last command          | Bx      Set Blank character to x
COL925 Reset  COL command line                 | Sx      Set command line Symbol to x
COL926 Undo   last COL command                 | T,Z,V   Reset Truncation,Verify,Zone
COL927 prefix Apply prefix cmd to lines w/match | @       Execute delayed block cmds
COL928 prefix@ Offset the prefix command @ amount | TSn     Set auto tab spacing to tn†
COL929 ATABx Auto Tab function using char ††† | TABx    Manual Tab using char †††
COL930 COMPx  Compress †tabbed† text using ††† | TRUNC   Allow column truncation
COL931 ALL    Block ALL using command line entry | TLeft   Left just. text within tabs
COL932 PICTu  Picture or display blank columns | TCent   Center text within tabs
COL933 \      Reverse the line order           | TBoth   Left and Right justify
```

The quick reference for the mathematical commands is:

```
COL934 ***** Mathematical and Logical Operations *****
COL935 Enter:
COL936 Y      Above the columns to be operated upon,
COL937 =      Above the column to receive the results. (==.= to specify decimal point.)
COL938 +-* /   Symbols or constants between columns to indicate the operation.
```

The remaining information and error messages that are used within COL are:

COL101 Valid addition options: tLline total, tCt cumulative line total and tStubtotal
COL102 WARNING The same symbol has been specified for both the
COL103 Command Line symbol and the Blank character symbols.
COL104 You cannot either insert/replace blanks or do math operations.
COL105 Target Not Found.
COL106 The replace command requires text to use, i.e. Rabc.
COL107 The overlay command requires text to use, i.e. Oabc. Text must be non blank.
COL108 Tab failed to find the request tab character t%t.
COL109 Auto tab failed to find the request tab character t%t.
COL111 The remove command requires text to use, i.e. -abc. Text must be non blank.
COL112 The pad command requires text to use, i.e. ¢abc. Text must be non blank.
COL113 UNDO = ON. You may enter tNoundt in the COL prefix area for faster response.
COL114 Automatic tab spaces set to %.
COL115 Invalid tab space value entered: t%t.
COL116 REXX Numeric Digits set to %.
COL117 COL commands will be saved in COL CMDLOG A.
COL118 COL command saves turned off.
COL119 Decimal is now %.
COL120 t%t The symbol used to indicate a blank character.
COL121 UNDO option has been activated.
COL122 A COL command has not been executed to undo or cannot be performed after HELP.
COL123 UNDO option has been turned off.
COL124 Undo has failed which in reality uses XEDIT GET. GET Return Code = %.
COL127 Zones set to %.
COL128 Truncation set to %. You may also want to reset Zones.
COL129 Verify set to %.
COL130 No unused character available for masking data, % used.
COL131 New command line symbol is t%t.
COL132 New blank character symbol is t%t.
COL133 No match with the COL command line found. Prefix t%t not applied.
COL134 Prefix command t%t has been applied to % line(s).
COL135 Unable to decode prefix request t%t.
COL136 Severe transmission error. XEDIT has set terminal mode to typewriter. COL has been aborted.
COL137 COL command t%t not found.
COL138 Shift RIGHT request exceeds right zone.
COL139 ALL (or NONE) of the sort groups must be numbered.
COL140 Unsupported Function
COL141 % lines have been truncated.
COL142 Use tRt to REPLACE a column with blanks. See XEDIT manual: COVERLAY & CREPLACE
COL143 The total was inserted in the column.
COL144 The width of the column specified is not wide enough to contain the total.
COL145 No blank space was found to insert the total.
COL146 % nhrs Total=% Min=% Max=% Avg=%
COL147 WARNING: One or more paragraphs have been flowed together.
COL148 would be truncated due to lack of space. Add blank lines or make column wider.
COL149 Zones must be between the word flow window.
COL150 Only one J or a single pair of J¢s allowed.
COL151 % line(s) could not be formatted with % spaces.
COL152 Best spacing possible has been performed on those lines.
COL153 Binary option is not supported.
COL155 You cannot use the tYt and the t t operators at the same time.
COL156 Column(s) truncated at % line(s).
COL157 Tab character specified was t%t.
COL158 Your disk does not have enough space to save COL changes.
COL159 You will not be able to issue the UNDO command.
COL160 Enter tLleft, tCtenter, tRtright or tBtoth (null to abort).
COL161 Enter number of spaces between words
COL162 (null to leave data with the current spacing).
COL163 No prefix command found.
COL164 The prefix commands have been offset by %.
COL165 No numbers found.
COL166 % is not supported.
COL167 Transposed columns require a width of %. XEDIT width setting is %.
COL168 XEDIT command t%t has been applied to % line(s).
COL169 Command t%t outside of XEDIT ZONE setting ignored.
COL170 No column specified for the results.
COL171 Invalid mathematical or logical operation on line: %.
COL172 Unable to place answer of t%t within specified column on line: %.
COL175 *****
COL176 % Lines contained errors. The text t%t was inserted.
COL177 If one column is named, then all columns must be named.
COL178 Profile % not found.
COL179 COL range does not contain any lines to edit.

COL180 COL cannot be called from a prefix macro.
 COL181 Invalid XEDIT column target ††.
 COL182 Column delete target †† has been applied to % lines.
 COL183 Conflicting justification specifications entered.
 COL184 Invalid number specifications entered.

Quick Reference

***** COLumn Quick Reference *****			
/cmd	Do the command here not behind	/	Shift screen to this column
†txt	Pad (overlay) blanks with text	K	convert column to UPPER case
-txt	Replace text with blanks	L	CONVERT COLUMN TO lower CASE
;	Set tabs	M	Move columns
†	Duplicate	N	Number (insert sequence nbrs)
+	Sum numbers in the column	Otxt	Overlay text
\	Reverse the command or column	P	Preceding target
A	Add (insert) a number of blanks	Qtxt	Line up text (Queue-up text)
B	Remove trailing Blanks to this column	QQtxt	Line up text behind character
C	Copy columns	Rtxt	Replace text
D	Delete columns	S	Sort
E	Emulate (copy forward) data	T	Truncation setting
-	De-emulate (blank duplicates)	U	Upon target
F	Following target	V	Verify setting
G	descending sort	<	Flow words justifying left
H	Verify in HEX	>	Flow words justifying right
Itxt	Insert text	W	Flow words just left and right
J	Justify (use with JL, JC, JR, or JB)	X	Xfer columns
	for Left, Center, Right or Both	Y	Column include(inverse of)
	directions. Add a number (i.e. JL1)		Column exclude
	to specify spacing between words.	Z	Zone setting

Figure 7. Quick Reference COL Commands

In order to do mathematical operations you enter:

***** Mathematical Operations *****	
Y	Above the columns to be operated upon,
=	above the column to receive the results and
+*/	Mathematical symbols or constants between columns in the COL command line or in the columns themselves. Also use: () & <>=%1234567890.

Figure 8. Mathematical Operations

The prefix commands are summarized in the following screen:

```

***** Prefix Commands adjacent to COL command line *****
?      Re-display last command
@      Execute delayed block commands
=      Re-execute last command
ALL    Block all command based on the COL command line entry
ATABx  Automatically tab SCRIPT data based on the character x
Bx     Set Blank character to x
COMMA  Set the decimal indicator to comma          (1.234.567,89)
COMPx  Compress SCRIPT data leaving x as the Script tab character.
DIGm   Set the REXX Numeric Digits setting for the Sum command to mn
FRENC  Set the decimal indicator to French        (1 234 567.89 or 1 234 567,89)
MATH   Mathematical evaluation of columns
PERIO  Set the decimal indicator for the SUM command to period (1,234,567.89)
PICTu  Picture text widths within the columns in the COL command line
prefix Apply prefix to lines matching the text in the COL command line
@      with prefix offsets the application of the prefix command
       and delays the execution of block commands
Reset  COL command line
\      Reverse the order of the lines within the COL range
Sx     Set command line Symbol to x
T      Reset Truncation Setting
TABx   Tab SCRIPT data to the XEDIT tab columns using x
TBoth  Left and Right justify text between all XEDIT tabs
TCent  Center text between all tabs
TLeft  Left justify text between all XEDIT tabs
TRANS  Transpose columns and lines
TRigh  Right justify text between all XEDIT tabs
TSn    Set the auto tab spacing to n
Undo   last COL command
V      Reset Verify setting
Z      Reset Zone setting

Any other command or text is passed to XEDIT
***** PF11 Column split/join within the current zone setting *****

```

Figure 9. Quick Reference COL Prefix Commands

Glossary

ALL. An XEDIT macro to display selected lines.

BookMaster. An IBM Publishing System Document Formatter

COL. An XEDIT COLumn edit macro.

COL command line. The line where COL commands are entered. This line is highlighted while in COL and usually contains question marks.

COL information area. The two lines which display the most commonly used COL commands and the PF key settings at the top or bottom of the screen.

COL prefix area. The XEDIT prefix spot adjacent to the COL command line.

COL prefix command. A command which is entered into the COL prefix area.

COL range. The top of the column to the bottom of the column. This range may have been identified when using HH as the prefix command. The first HH identifies the top of the range and the second HH identifies the bottom of the range.

Comma. The representation of numbers where a decimal point may be used to separate thousands and a comma is used to show the decimal location such as:

1234567,89
1.234.567,89

Emulate. The function which copies existing data forward within a column.

French. The representation of numbers where blanks may be used to separate thousands and either a period or a comma is used to show the decimal location such as:

1234567,89
1 234 567,89
1 234 567.89

GML. Generalized Markup Language

King size. Uppercase letters.

Period. The representation of numbers where commas may be used to separate thousands and a period is used to show the decimal location such as:

1234567.89
1,234,567.89

Picture. Viewing columns to determine which columns contain blanks and which columns contain text.

POWERINP. XEDIT's power input function.

Queue. The function which lines up text.

REXX. The VM System Product Interpreter

SCRIPT. The IBM Document Composition Facility and its text processing programs.

XEDIT. The VM System Product Edition

Index

' (pad)
 See *also* - (remove text)
 / operator 37
 characters in column one 37
 command 23
 example 130
; (set tabs) 44
- (remove text)
 See *also* ' (pad)
 / operator 37
 characters in column one 37
 command 24
 example 130

Special Characters

_ (de-emulate)
 See *also* emulate
 \ operator to execute bottom to top 30
 command 30
 examples 115, 127, 128, 134
? (recall) 40
/
 column one operations 37
 do it here 37
 move screen 36
 operator 37
† (duplicate) 19
@ (prefix offset)
 command 109
 example 115, 126, 131, 144
\$ in numbers being summed 48
*
 in COL command line 38
 in COL prefix command 101
\ command
 de-emulate up a column 30
 emulate up a column 29, 122
 number up a column 33
 reverse column 31
 reverse command to execute from bottom to
 top 38
 reverse line order 41
 sum up a column 49
+ (sum)
 accuracy
 changing 53, 54
 default 53
 average of numbers 53
 comma 54
 command 47
 count of numbers 53
 cumulative line totals 49
 digit accuracy 53, 54
 dollar sign 48

+ (sum) (*continued*)
 French 54
 line totals 50
 macro call, returning results 53
 maximum of numbers 53
 minimum of numbers 53
 number count 53
 period 54
 subtotals 51
 total
 placement of 48
 returned in message for macro call 53
<
 command (word flow left) 84
 See *also* commands "w" and ">"
 example word flow 135
 prefix command example 137
 prefix command, using in 100, 107
= (reexecute) 40
>
 command (word flow right) 85
 See *also* commands "w" and "<"
 prefix command example 137
 prefix command, using in 100, 107
⌊
 command (exclude columns) 96
 prefix command example 96, 100, 107, 115, 129,
 143, 149
 prefix command, using in 107

A

A command (add)
add
 / operator 37
 column one
 examples 126, 141, 147, 162
 working with 37
 command 15
 lines with specific text 118
 mathematical 62
 to sum 47
 See *also* sum
align text
 justify 73
 queue
 command 67
 example 133
ALL
 blocks of text 91
 COL prefix command 87
 predefined blocks of text 87, 92
 XEDIT iii
alphabetic sequencing 32

- Arabic sequencing 32
- arbitrary character 112
- asterisk
 - in COL command line 38
 - in COL prefix command 101
- ATAB
 - option 173
 - prefix command 80
- auto exit
 - automatic option in profile 4
 - ECOL 9, 173
 - PF3(15) 6, 9
- automatic
 - exit
 - macro call 173
 - profile option 4, 9
 - tab
 - command 80
 - space setting in profile 4
 - spacing 80, 174

B

- B command (blank) 71
- blank
 - duplicate text 134
 - See also* de-emulate
 - fields for math containing 64
 - filling in with data above 29, 134
 - line, inserting 103
 - line, removing 151
 - picture or view columns containing 39
 - redundant text 30
 - See also* de-emulate
 - remove trailing 71, 150
 - replace character with 24
 - symbol
 - changing 41
 - option 173
 - profile setting 4
 - treating as zero containing 64
- block
 - delay execution 110
 - of text
 - editing 87
 - rearrange 146
 - sort 153
 - prefix commands 110
- BookMaster
 - definition list sort 160
 - glossary list sort 160
 - list, converting to or from 125, 129
 - ordered list sort 160
 - simple list sort 160
 - table editing 87, 136
 - table row sort 160
 - unordered list sort 160

C

- C command (copy) 17
- case
 - lower 27
 - upper 28
- center
 - column, within 78
 - line 78
 - text within tab settings 83
- changing
 - blank symbol 41
 - blank symbol, example 115, 128
 - column width when truncation occurs 10
 - command line symbol 41
 - file width 10
 - sum accuracy 53, 54
 - to lower case 27
 - to upper case 28
- COL
 - ALL prefix command 87
 - blank symbol 41, 173
 - comma
 - mathematics 56
 - prefix command 54
 - setting in profile 4, 173
 - summing 54
 - command line
 - location 3
 - option, COL macro 173
 - prefix area 3
 - stacking 175
 - symbol 173
 - wrap 9
 - digit
 - as used in summing 47
 - prefix command 54
 - setting in profile 4
 - entering XEDIT commands 44
 - examples 172
 - French
 - mathematics 56
 - prefix command 4, 54
 - setting in profile 173
 - summing 54
 - information area 3, 4
 - options 173
 - period
 - as used in summing 54
 - prefix command 54
 - setting in profile 4
 - prefix area 3
 - prefix commands
 - ? (recall COL command) 40
 - @ (execute delayed block commands) 111
 - @ (offset) 109
 - @ example 115, 126, 131
 - * (all columns) 101
 - < (less than) 100, 107
 - = (execute last command) 40

COL (*continued*)

- prefix commands (*continued*)
 - > (greater than) 100, 107
 - ¬ (not) 96, 100
 - atab (auto tab) 80
 - b (blank character) 41
 - comma (comma setting) 54
 - comp (compress) 82
 - dig (digit setting) 53
 - frenc (French setting) 54
 - nostore 44
 - other (command passed to Xedit) 44
 - perio (period setting) 54
 - reset (reset COL command line) 40
 - s (command line symbol) 41
 - stacking 175
 - store 44
 - T (reset XEDIT truncation setting) 45
 - tab (manual tab) 81
 - ts (auto tab spacing) 80
 - undo (undo) 38
 - V (reset XEDIT verify setting) 45
 - XEDIT prefix application 44
 - XEDIT similarity 13
 - Z (reset XEDIT zone setting) 45
- prefix option
 - when using COL macro 173
- profile 4
- range marker 3, 4, 176
- re-display 40
- re-execute 40
- save commands 44
- screen 3
- setup 6
- stacking commands 175
- store commands 44
- T prefix commands 45
- target
 - default in profile (PROFCOL) 4
 - specifying in 6, 173
- undo prefix command 38
- V prefix command 45
- Z prefix command 45

COLALL 91, 136

COLCMD 175

COLMAT 122, 175

color of information area 4

COLPCM 175

COLPRF 175

column

- add 15
- blank treated as zero 64
- compare 123
- current 36
- exclude 96
- include
 - command 95, 97
 - examples 141, 171

column (*continued*)

- make 1 from 2 138, 170
- make 2 from 1 140, 168
- match 100
- named 59
- picture 39
- relocate 138, 140, 168
- sort 35, 158
- target
 - / operator 37
 - delete 13
 - following 17
 - preceding 17
 - screen shift 36
 - upon 19
- transposing into lines 43
- viewing columns 39
- width, maximum 10
- width, viewing 39
- column one, working with
 - / operator 37
- example
 - add 126, 141, 147, 162
 - insert 130
 - overlay 37
 - pad 37
 - remove 126
 - replace 37
- columnize text
 - See queue and justify
- COLXED 175
- comma
 - changing setting
 - in numbers being summed 48
 - mathematics 56
 - option 173
 - prefix command 54
 - profile 4
 - summing 54
- command line
 - location 3
 - resetting 40
 - stacking 175
 - symbol 173
 - wrap 9
- command summary 10, 179
- comp
 - compress option 173
 - compress prefix command 82
- compare columns 123
- compare operator 62
- concatenation 62
- converting to and from GML or BookMaster list 125
- copy backward 29, 122
- copy command 17, 17
- copy forward 29
 - See also emulate

count occurrences in a list 127
cumulative line total 50
current column 36

D

D command (delete) 13
dates, sorting of 159
de-emulate text
 See also emulate
 \ operator to execute bottom to top 30
 command 30
 examples 115, 127, 128, 134
decimal point location 56
default 4, 84
definition list sort 160
delay
 execution of prefix block commands 110
delete
 columns, examples 126, 129, 133, 145, 147, 149,
 150, 154, 164
 command 13
 lines 101, 107
digit
 changing setting of 54
 prefix command 54
 profile 4
 setting for sum accuracy 53, 54
divide 62
do it here 37
dollar sign in number being summed 48
duplicate
 command 19
 lines 104
 text
 blank 134
 mark 143
 remove 152

E

E command (emulate) 29
ECOL
 exit COL 9, 173
 used in macro 176
emulate
 See also _ (de-emulate)
 \ operator to execute bottom to top 29
 command 29
 examples 115, 127, 128, 134, 147, 149, 154, 163
error messages 4, 54, 177
examples 172
exclude
 columns 95
 command 96, 97
 examples, applying XEDIT 126, 137, 143, 147, 154,
 161, 164, 167, 168, 169
 lines iii, 106
 sort, lines 34

exclude (*continued*)
 XEDIT lines iii, 106
execute delayed block prefix commands 111
exit COL
 automatic option in profile 4
 ECOL 9, 173
 PF3(15) 6, 9

F

F target (following) 17
file width and truncation 10
fill in blank columns with text from above 29, 134
flow text
 example 135
 left 84
 right 85
 right and left 86
 text compression 87
 zone setting, used with 87
following target 17
French
 changing setting
 mathematics 56
 option 173
 COL macro 173
 prefix command 54
 profile 4
 summing 54
fullread as used in XEDIT 5
function
 built-in 64
 name 63
 user written 63

G

G command (descending sort) 34
get
 columns 97, 168
 multiple columns 99, 100
getting started 3
glossary 181
glossary list sort 160
GML
 definition list sort 160
 glosary list sort 160
 list, converting to or from 125
 ordered list sort 160
 simple list sort 160
 table editing 87, 136
 table row sort 160
 unordered list sort 160
greater than
 command (word flow right) 85
 See also commands "w" and "<"
 prefix command example 137
 prefix command, using in 100, 107

groups of text
 creating 29
 editing 87

H

H command (verify in hex) 45
H prefix to call COL 6
header
 sorting sections of 153
 sorting text with 166
help 10
hex
 numbering 32
 verify 45
HH prefix to call COL 6

I

I command (insert) 16
ignore capitalization 67
imbedded header, sorting text with 166
include
 columns 95
 command 96, 97
 examples, applying XEDIT 126, 137, 143, 147, 154,
 161, 164, 167, 168, 169
 lines iii, 106
 sort, lines 34
 XEDIT lines iii, 106
incrementing, numbers 32
information area 177
 color 4
 defined 3
 location 3, 4
 suppressing 175
information messages 177
input mask 109, 118
insert
 / operator 37
 command 16
 example 116, 133, 144, 145
 lines
 blank, example 103
 offset from existing lines 109, 116
 separator 115
 with specific text 109, 118
 XEDIT mask 118
 text into column one 37, 130
interchange columns and lines 43

J

J command (justify) 73
justify command
 between tab settings 83
 both left and right 76
 center 73, 78
 left 73, 84, 133

justify command (*continued*)
 right 75
 spacing between words 79

K

K command (upper case) 28

L

L command (lower case) 27
leading zeros 23, 25
left
 justify 73, 84, 133
 justify using tab settings 83
 move screen 36
 word flow 84
less than
 command (word flow left) 84
 See also commands "w" and ">"
 example word flow 135
 prefix command example 137
 prefix command, using in 100, 107
line sum 49
lines transposed into columns 43
lines with specific text
 blank 103
 exclude 106
 insert
 lines with 118
 new lines around 103, 104
 offset from existing lines 109
list
 converting to and from GML or BookMaster 125
 count items in 127
 sort 160
 values within a range 137
logic
 operation 54, 62
 operator 101
lower case 27

M

M command (move) 18
macro call
 command line 173
 example 115, 122, 128
 options 173
 stacking commands for 175
 sum results in last message 53
 zone setting, honoring 173
manual tabbing 81
mark duplicate text 143
marker, range 3, 4, 176
mask, input 109, 118
matching the COL command line
 apply XEDIT prefix command to 100
 column match 100

- matching the COL command line (*continued*)
 - text match 101
- math command with named columns 59, 122
- math prefix command 56
- mathematic
 - error messages 4, 54
 - named columns 59, 60
 - operation 54
 - option 122, 173
 - previous answer 61
 - significant digits 56
 - update in place 61
 - y. variables 59
- message line 4, 175
- messages 177
- metric number 54
- move
 - command 18, 17
 - example 133, 145
- multiple 176
- multiple screens 176
- multiply 62

N

- N command (number) 32
- named columns 59, 60
- National Language Support (NLS)
 - COL messages 177
 - summing numeric values 54
- noexit option 173
- noinfo option 173
- noprof option 173
- not
 - command (exclude columns) 96
 - prefix command example 96, 100, 107, 115, 129, 143, 149
 - prefix command, using in 107
- noundo
 - default setting in PROFCOL 4
 - option 173
 - prefix command 38
 - system response 38, 174
- nulls 5
- number
 - \ operator to execute bottom to top 33
 - alphabetic sequence 32
 - Arabic sequence 32
 - examples 126, 137, 145
 - function bottom to top 64
 - hex 32
 - incrementing by 32
 - padding leading zeros 23
 - random bottom to top 56, 59
 - repeating by 32
 - starting at 32
- numeric form 54

O

- O command (overlay)
 - See also ?
 - COL command 22
- offset
 - command 109
 - example 115, 126, 131, 144
 - XEDIT prefix command 109
- one column from two 138, 140, 168
- operator
 - column one operations 36
 - current column 36
 - execute command from bottom to top 38
 - logic 101
 - options using the (38
 - prefix 101
 - using the * 38
- options
 - COL macro 173
 - command line 38
- ordered list sort 160
- overlay
 - / operator 37
 - command 22
 - examples 149, 150
 - text over column one

P

- P target (preceding) 17
- pad
 - See also remove text
 - characters in column one 37
 - command 23
 - leading zeros 23
- pad text
 - See also - (remove text)
 - / operator 37
 - characters in column one 37
 - command 23
 - example 130
- percent of total 119
- period
 - as used in summing 54
 - changing setting
 - option 173
 - prefix command 54
 - profile 4
 - option 173
- PF keys
 - 1 10, 175
 - 2 11, 175, 177
 - 3(15) 9, 175
 - within column exclude 98
- PICture prefix command 39
- POWERINP into a column 98
- preceding target 17

- predefined blocks of text 92
- prefix area 3
- prefix command for COL command line
 - ? (recall command) 40
 - @ (offset) 109
 - * (all columns) 101
 - < (less than) 100, 107
 - = (reexecute command) 40
 - > (greater than) 100, 107
 - ¬ (not) 96, 100
- atab 80
- auto tab 80
- auto tab spacing 80
- blank character 41
- block delay execution 110
- comma setting 54
- command line symbol 41
- command passed to XEDIT 44
- compress 82
- digit setting 53
- execute delayed block commands 110
- French setting 54
- manual tab 81
- math 56
- nostore 44
- offset 109
- period setting 54
- recall COL command 40
- reexecute COL command 40
- reset
 - COL command line 40
 - XEDIT prefix command 111
 - XEDIT truncation setting 45
 - XEDIT verify setting 45
 - XEDIT zone setting 45
- store 44
- tab 81
- undo 38
- XEDIT prefix application example 115, 126, 127, 128, 129, 131, 137, 147, 151, 154, 161, 164, 171
- prefix command to execute COL 6
- prefix operator 101
- prefix option 173
- PREFIXH
 - PROFILE XEDIT entry 6, 9
 - screen results 6
- PREFIXH2
 - PROFILE XEDIT entry 6, 9
 - screen results 8
- previous answer 61
- priority 176
- profile
 - alternate 174
 - PROFCOL 4
 - sample 4
- put
 - columns 97, 168
 - multiple columns 99, 100

Q

- Q command (queue or align text) 67, 154
- QQ command (queue trailing) 69
- queue example 133
- queue or align text 67, 154
- quick reference 177, 179

R

- R command (replace) 21
 - See also O command (overlay)
- random numbers 56, 59
- range marker 3, 4, 176
- range, list values in 137
- ranking 127
- re-display 40
- re-execute 40
- rearrange blocks of headers 146
- recall command 40
- reference
 - PF key 10
 - quick, table 179
- reformat tables 148
- remove
 - blank lines 130, 151
 - characters 24
 - See also – (remove text)
 - characters, example 130
 - duplicate text 152
 - leading zeros 25
 - lines with specific text 103
 - trailing blanks 71, 150
- remove text
 - See also ' (pad)
 - / operator 37
 - characters in column one 37
 - command 24
 - example 130
- repeat command 40
- repeating numbers 32
- replace
 - / operator 37
 - character with blank 24
 - command 21
 - examples 126, 127, 128, 137, 145
 - text over column one 37
- reserved lines 4, 175
 - COL command line 40
 - disconnected service machine 175
 - prefix command 40
 - truncation setting 45
 - verify setting 45
 - zone setting 45
- reset
 - COL command line 40
 - XEDIT prefix command 111
- respect capitalization 67

- response, system 38, 174
- reverse operator
 - de-emulate up a column 30
 - emulate up a column 29, 122
 - number up a column 33
 - reverse column 31
 - reverse command to execute from bottom to top 38
 - reverse line order 41
 - sum up a column 49
- reverse text 31
- REXX
 - code, editing 87
 - expressions 54
 - functions 54
 - operators 62
 - user function or exit 63
- right
 - justify 75
 - justify using tab setting 83
 - move screen 36
 - word flow 85
- ring 176

S

- S command (sort) 34
- sample profile 4
- save COL commands in a file 44
- save commands in a file 44
- scale default in profile (PROFCOL) 5
- screen
 - command line wrap 9
 - layout 3
 - right and left 36
 - up and down 3
- SCRIPT
 - aligning tabbed text into columns 80, 81
 - compressing columns 82
- seed number 32
- sentence spacing
- separator lines 115
- sequencing 32
 - See *also* number
- set tabs (;) 44
- set up 3
- setting in profile
- setup COL 6
- shift
 - screen 36
 - text
 - center 78
 - left 84
 - queue or line up 67
 - right 85
 - right example 140
 - to upper case 28
- show lines previously excluded 107, 144
- significant digits command
 - math operations command 56
 - sum function command 53
- similar XEDIT prefix command 13
- simple list sort 160
- single column sort 35
- sort
 - blocks of text 153
 - columns 34, 35, 158
 - dates 159
 - examples 127, 128, 147, 154, 163, 166, 167
 - excluded lines 34, 167
 - header sections 153
 - lists 160
 - single column 35
 - zone setting, used with 35
- space
 - auto tab, between columns 80, 174
 - between sentences 4, 84
 - justify, between words 73, 79
 - setting in profile for auto tab 4
- stacking COL commands 128, 175
- starting number 32
- store COL commands in a file 44
- subtotal 51
- subtract 62
- sum
 - \ operator to execute bottom to top 49
 - accuracy
 - changing 53, 54
 - default 53
 - across one line 48, 49
 - all numbers within an area 53
 - average of numbers 53
 - C option to cumulative sum lines 50
 - comma 54
 - command 47
 - count of numbers 53
 - cumulative line totals 49
 - digit accuracy 53, 54
 - dollar sign 48
 - examples 51
 - French 54
 - L option to sum lines 49
 - line totals 50
 - macro call, returning results 53
 - maximum of numbers 53
 - minimum of numbers 53
 - number count 53
 - period 54
 - placement of 48
 - returned in message for macro call 53
 - S option for subtotals 51
 - subtotals 51
 - total
 - placement of 48
 - returned in message for macro call 53

- summary, command 10, 179
- swap text end for end 31
- symbol
 - blank 41, 173
 - command line 41, 173
- system response, undo setting 38, 174

T

- T command (truncation) 45
- T prefix command 45
- tab
 - automatic 80
 - justifying text within 83
 - manual 81
 - option, COL macro 173
 - prefix command 80
 - space setting in profile 4, 173
 - spaces for automatic 80, 174
- tab space (TS)
 - option 173
 - prefix command 80
- table
 - BookMaster or GML 87, 136
 - reformatting 148
 - row sort 160
- target
 - COL 6, 173
 - COL ending
 - default in profile (PROFCOL) 4
 - specifying in command 173
 - column 14
 - following 17
 - preceding 17
 - upon 19
 - within column exclude for XEDIT 98
- text
 - blocks of
 - editing 87
 - rearranging 146
 - sorting 153
 - blocks predefined in COLALL 91, 92, 136
 - duplicate
 - blank 134
 - mark 143
 - remove 152
 - fill in the blanks 29, 134
 - location within a column 39
 - match, applying prefix commands to 100
 - remove duplicate 30
 - width of a column, viewing 39
- text alignment
 - < command 84
 - > command 85
 - center 78
 - justify 73
 - queue
 - command 67
 - example 133

- text alignment (*continued*)
 - remove trailing blanks 71, 150
 - Script tabs
 - automatic aligning into columns 80
 - compress existing columns 82
 - manual aligning into columns 81
 - word flow 84
 - word spacing 79, 84
- text flow
 - example 135
 - left 84
 - right 85
 - right and left 86
 - text compression 87
 - zone setting, used with 87
- total
 - \ operator to execute bottom to top 49
 - across one line 48, 49
 - all numbers within an area 53
 - C option to cumulative sum lines 50
 - command 47
 - examples 51
 - L option to sum lines 49
 - percent of 119
 - placement of 48
 - returned in message for macro call 53
 - S option for subtotals 51
 - sum 47
- trailing blanks 71, 150
- transfer 20, 17
- transpose columns and lines 43
- truncation
 - column width 10
 - command 45
 - file width 10
 - option 173
 - restore XEDIT setting 45
 - setting for XEDIT 45
- TS (tab space)
 - option 173
 - prefix command 80
- two columns from one 140, 170

U

- U target (upon) 19
- undo
 - default setting in PROFCOL 4
 - option 173
 - prefix command 38
 - system response 38, 174
- unordered list sort 160
- update in place 61
- upon target 19
- upper case 28

V

V command (verify) 10, 44
V prefix command 45
verify
 hex 45
 setting in XEDIT 9, 44
view column width 39

W

W command (word flow) 84
 See also commands ">" and "<"
width
 line size 10
 viewing text within a column 39
word flow
 example 135
 left 84
 right 85
 right and left 86
 text compression 87
 zone setting, used with 87
word spacing
 auto tab 80
 justify 79
wrap 9

X

X command (transfer) 20, 17
XEDIT
 absolute column target with column exclude 98
 absolute target with column exclude 98
 add 15
 ALL iii
 ALL compared with ALL in COL prefix area 44
 COL prefix area entry 44
 commands, applying to a column 97, 168
 delete 107
 DELETE a column 97
 exclude
 COL compatibility iii
 example 106, 115, 126, 129, 137, 147, 154, 161, 164
 fullread 5
 get 168, 174
 GET into a column 97, 168
 ignore 67
 insert 16
 macro
 example 63, 115, 128
 options 173
 stacking COL command 122, 175
 mask 109, 118
 named target with column exclude 98
 nulls 5
 POWERINP into a column 97
 prefix command
 applying to columns 52, 95

XEDIT (*continued*)

 prefix command (*continued*)
 applying to lines that contain the text in the COL command line 101
 applying to lines that match the COL command line 100
 COL commands that are the same 13
 delaying block command 110
 example 115, 126, 127, 128, 129, 131, 137, 147, 151, 154, 161, 164, 171
 offset 109
 offset examples 115, 126, 131
 option 173
 resting 111
 similarity with COL 13
 priority with column exclude 176
 put 168, 174
 PUT into a column 168
 relative target with column exclude 98
 respect 67
 ring with column exclude 176
 screens, multiple with column exclude 176
 sort 34
 string target with column exclude 98
 target
 for COL 6, 173
 with column exclude 98
 verify setting 9, 44

Xfer 20, 17

Y

Y command (include) 95, 97
 See also exclude, columns
Y. variables 59

Z

Z command (zone) 45
Z prefix command 45
zeros, leading 23, 25
zone
 macro call uses 122, 173
 restoring XEDIT setting 45
 sort, used with 35, 158
 wordflow, used with 87
 XEDIT setting 45



Printed in U.S.A.