27
8/8/77
SAND77-1052
Unlimited Release
25871T45

3151

User/Programmer Guide for UCMD 82 : Compaction of Drawing Dictionary

Donald K. Robbins

MASTER

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115
and Livermore, California 94550 for the United States Energy Research
and Development Administration under Contract AT(29-1)-789
Printed July 1977

Sandia Laboratories

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Issued by Sandia Laboratories, operated for the United States Energy Research & Development Administration by Sandia Corporation.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

*
SAND77-1052
Unlimited release
Printed June 1977 ✓

User/Programmer Guide for UCMD 82;
Compaction of Drawings Dictionary

Donald K. Robbins
Computer Aids System Development Division 9624
Sandia Laboratories
Albuquerque, NM 87115

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ABSTRACT

This document describes Program UCMD 82 -- an Applicon AGS/870 User Command for compacting a drawing dictionary. Input is from a drawing on the table. Output is a compacted dictionary on the table and a summary of the before and after sector dictionary map.

Printed in the United States of America
Available from
National Technical Information Services
U. S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161 9.50
Price: Printed Copy ~~\$4.50~~; Microfiche \$3.00

*

CONTENTS

	Page
1.0 Introduction	7
2.0 Execution Instructions	7
2.1 Input	7
2.2 Output	7
2.3 Operating Procedure	8
2.4 Error Messages	8
2.5 Example	8
3.0 UCMD 82 Program Documentation	10
3.1 Documentation Guidelines	10
3.2 Date of Documentation	10
3.3 Name of Program	10
3.4 Name of Author of Program	10
3.5 Name of Program Administrator	10
3.6 Date Program was Completed	10
3.7 Brief Description of Program	10
3.8 Full Narrative Description of Program	11
3.8.1 General	11
3.8.2 Initialization	11
3.8.2.1 Buffers	11
3.8.2.2 I/O Buffers	11
3.8.2.3 Arrays	12
3.8.3 Options	12
3.8.4 TABL	13
3.8.5 REPAT	13
3.8.6 SEARCH	13
3.8.7 NOMATCH	14
3.8.8 ADD	14
3.8.9 END1	14
3.8.10 Subroutines	15
3.8.10.1 Utilities	15
3.8.10.2 XTRCT.CMD	15
3.8.10.3 STR.CMD	16
3.8.10.4 DECODE.SECTOR	16
3.9 Hardware Configuration Required	17
3.10 Language Used	17
3.11 Operation System Used	17
3.12 Type of Program	17
3.13 Special Routines Used	17
3.14 Form of Program	17
3.16 Program Flow Chart	18
3.17 Program Listing	20
3.18 Backup Capability	20
4.0 Record of Changes to the Program	20
5.0 Key Words	20
6.0 References	21
7.0 Distribution	22

*

*

1.0 Introduction

The Applicon Graphics System (AGS) has provision for a dictionary. The complete dictionary is contained in six disc sectors of 512 bytes each. There are 506 bytes available for storage in the first sector and 508 bytes available in each of the following five sectors. There are three types of entry in the dictionary: 1) Command Symbols (CS), 2) Macros, and 3) Menu commands. When an entry is deleted from the dictionary, its space is taken by moving following entries up. Thus the free space is at the bottom of the sector. Entries do not cross sector boundaries. Therefore, it is possible that the pieces of free storage available at the end of each sector are unusable. With the map from UCMD 81 it may be possible to shuffle the commands to make better use of the free space available. In other words, using the map from UCMD 81, dictionary entries can be deleted and re-inserted. For instance an entry (A) uses 10 bytes of storage and there are 6 bytes of free storage available in sector 1. In another sector an entry (B) uses 16 bytes. Deleting A and B and then re-inserting B and then A would use the 16 bytes of sector 1. UCMD 82 does this shuffling automatically.

2.0 Executing Instructions

The Applicon Graphics System (AGS) provides a dictionary for a drawing. UCMD 82 provides a program for compacting this dictionary.

2.1 Input

UCMD 82 is executed from an AGS activity with a graphics display (GD) and keyboard (KB) or from an activity with a DECwriter. Three options are available from the keyboard. The options are

- a. 'D'escribe This lists the options available.
- b. 'T'able This runs the program. The drawing on the table will be compacted.
- c. 'A'bort This aborts UCMD 82 and returns to the AGS system

2.2 Output

The execution messages produced by UCMD 82 start with the initial prompting message to allow the user to select the appropriate option. All input is echoed back to verify what the computer sees. If an illegal option is selected then this is noted along with the opportunity to retry. At the end of the command, a summary table is produced of the sector map before and after.

*

*

2.3 Operation Procedure

UCMD 82 should be loaded and executed from the CTRL mode.
If the 'T' option is selected, UCMD 82 proceeds at once to compact the dictionary of the drawing on the table.

2.4 Error Messages

There are a few error messages which are produced if the program runs into a pattern it does not recognize.

2.5 Example

```
UCMD 82; UCMD 82;  
USER COMMAND 82 - VERSION 1.1  
COMPACTION OF DICTIONARY
```

NEXT ENTRY IS A ONE CHARACTER OPTION. CHOICES ARE
D,T,A

'D' GIVES A DESCRIPTION, 'T' RUNS THE PROGRAM,
'A' ABORTS PROG.

D;

OPTION SELECTED WAS 'D'

DESCRIPTION OF HEADINGS AND PROGRAM

DICTIONARY IS CONTAINED IN 6 SECTORS
EACH SECTOR CONTAINS 512 BYTES
COMMANDS DO NOT CROSS SECTORS
A SECTOR IS CLOSED-UP WHEN A COMMAND IS DELETED
A SECTOR SUMMARY IS PRINTED WITH A
BEFORE AND AFTER MAP LISTED

NEXT ENTRY IS A ONE CHARACTER OPTION. CHOICES ARE
D,T,A

'D' GIVES A DESCRIPTION, 'T' RUNS THE PROGRAM,
'A' ABORTS PROG.

A;

OPTION SELECTED WAS 'A'

END OF UCMD 82

*

*

UCMD 82; UCMD 82;
USER COMMAND 82 - VERSION 1.1
COMPACTION OF DICTIONARY

NEXT ENTRY IS A ONE CHARACTER OPTION. CHOICES ARE
D,T,A

'D' GIVES A DESCRIPTION, 'T' RUNS THE PROGRAM,
'A' ABORTS PROG.

T;

OPTION SELECTED WAS 'T'

BEFORE

#COMMANDS #FREE BYTES

25	10
33	36
33	5
18	19
15	44
4	226

AFTER

#COMMANDS #FREE BYTES

26	0
34	0
32	0
19	0
14	45
3	295

END OF UCMD 82

*

*

3.0 UCMD 82 Program Documentation

3.1 Documentation Guidelines

This manual documents Program UCMD 82 per "Guidelines for 9624 System Program Documentation," dated January 15, 1975.

3.2 Date of Documentation

June 1977

3.3 Name of Program

UCMD 82

3.4 Name of Author of Program

D. K. Robbins - 9624

3.5 Name of Program Administrator

Nancy Nelson - 9621

3.6 Date Program was Completed

April 1977

3.7 Brief Description of Program

UCMD 82 compacts the drawing dictionary on the table. The output is a sector-by-sector summary of the number of commands and the free space before and after.

*

*

3.8 Full Narrative Description of Program

3.8.1 General

UCMD 82 is written in the BCPL Programming Language

The source code was developed on the HP 2644A terminal. This video interactive terminal permits easy editing of existing code. Accordingly the program is highly commented.

Because of the number of messages in the program (which causes a STATIC table overflow in the BCPL compiler), the program is separated into 3 segments. These are indicated by a single period on a line by itself. The global entities used by each segment must be declared.

3.8.2 Initialization

The program starts with a comment block as to what is afoot. This is followed by declaring the globals and variables used. The files AGSLIB.HDR and FILDEF.HDR are obtained with the GET command. These 2 files contain global declarations for the BCPL support library. Cells 1 thru 139 are accounted for by these globals. The subroutines used by UCMD 82 are setup in cells 140 thru 160.

The STATIC statements reserve locations for the various variables used in the program.

Vectors are set up for the variables which need vectors.

These vectors fall into 3 categories:

- a) Buffer (disc)
- b) I/O buffers (keyboard)
- c) Arrays

3.8.2.1 Buffers

There are a 256 word buffer, BUFF, where a disc sector is stored and a 512 word buffer, UN, for the byte unpack of BUFF.

3.8.2.2 I/O Buffer

There are 2 KB buffers, ANS and ANS1. Two command buffers are used for 1) storing commands after extraction from a sector and 2) for commands waiting to be stored back to a sector.

*

*

3.8.2.3 Arrays

There are 5 arrays.

- a) **STRT.** A one dimensional array, containing six 64-word segments in which the starting location of each dictionary entry is stored.
- b) **LNG.** A one dimensional array containing six 64-word segments in which the length of each dictionary entry of a sector is stored.
- c) **FREE.** A 6-word array containing the number of free bytes per sector.
- d) **NCOM.** A 6-word array containing the number of entries per sector.
- e) **DIST.** A one dimensional array with six 110-word segments. Each LNG entry is directly indexed into the DIST array for keeping track of what length commands are in each sector.

3.8.3 Options

Next, the program writes out a message to the user identifying itself (UCMD 82) and also lists the different options which are available.

The options are: 'D'escribe, 'T'able, 'A'bort.

The first character of one of the above is entered from the keyboard followed by a <CR>. This is accepted by the routine GETSTRING, packed by PACKSTRING, and listed out to the user. The option selected then goes into a multi-branch switch construct (SWITCHON) of BCPL. If any other character other than one of the above is typed, then the DEFAULT CASE rejects it and returns to 3.8.3 for another try.

*

*

- a) 'D'escribe
This CASE calls out the subroutine DESCRIP and returns to 3.8.3
- b) 'B'ug
This CASE turns SW to 1. This results in much intermediate output which was used for debugging. Control returns to 3.8.3
- c) 'T'able
This case transfers to 3.8.4
- d) 'A'bort
This case transfers control to 3.8.9
- e) Default
Any other character ends up here. A message is written to the user and control returns to 3.8.3

3.8.4 TABL

Control is transferred to this label if the 'T'able option is used. The 6 disc sectors are read, and each of the 256 word buffers is transferred to DECODE.SECTOR where the tables are filled from the entries in the dictionary. The tables are

- 1) STRT contains the starting bytes of the entries,
- 2) LNG contains the length (in bytes) of the entries,
- 3) FREE contains the number of free bytes per sector,
- 4) NCOM contains the number of entries per sector.

DISTR is next called where the entries in LNG are tallied for each sector into the DIST array. PRINTSTAT is called to put out the sector summaries before compaction starts. Control then drops to REPAT in 3.8.5.

3.8.5 REPAT

At this label a search is made of the sectors which have free space. If none is found then the program terminates from this point by going to ENDI at 3.8.9

If there is free space, the number of bytes is put into variable N1 and control passes to 3.8.6 (SEARCH). The current sector is I.

3.8.6 SEARCH

Here a search is made of the other sectors (J) for N1. If an exact match is found, control is transferred to ADD at 3.8.8. This indicates that the unused space in sector I can be exactly filled by an entry in the J'th sector. If there is no exact match, control transfers to 3.8.7 (NOMATCH).

*

*

3.8.7 NOMATCH

Here the I'th sector is searched for the smallest length entry in that sector. This value is put in variable L. Now L and N1 are added together and control is transferred to 3.8.6 to check the other sectors for this number of bytes.

3.8.8 ADD

This is the place where the movement of entries occurs. There are two cases: 1) If an exact match is found, the entry in the J'th sector is inserted into the I'th sector and the J'th sector's free space is increased, so this has the effect of moving the free space to the later sectors; 2) on the other hand, if an exact match is not found, the entry from the J'th sector replaces the entry from the I'th sector and the I'th entry goes to the J'th sector. Again this has the effect of reducing the free space in the earlier sectors; Control is returned to 3.8.5 (REPAT).

3.8.9 END1

When the iteration above is completed, then there is yet a little touch-up between Sector 5 and Sector 6. END.GAME is called which moves as many entries from Sector 6 to Sector 5 as is possible, depending on the free space in Sector 5. PRINTSTAT prints out the sector summaries after compaction. A message is then put out indicating the end of UCMD 82 and control is returned to the AGS system.

*

3.8.10 Subroutines

3.8.10.1 Utilities

- a) GETSTRING. This routine reads a character from the keyboard and places it in an array.
- b) PRINTUSER. This routine prints the user's name on the output device.
- c) DEBUG. This routine tests the error cell. If zero it RETURNS. If not, it prints out the cell in octal, as well as the message.
- d) DESCRIP. This routine consists only of messages which are put out to the output device explaining the options available.
- e) UNPACK. This routine unpacks two bytes per word to an array with one byte per word, right-justified.
- f) PACK. This routine packs the byte array back into the word array.
- g) PRINTBUFF. This routine decodes a word buffer and prints the bytes in octal.
- h) COMPACT. This routine compresses the space in a byte array where a dictionary entry has been removed.
- i) DISTR. This routine takes the LNG array (which is organized into six 64 word segments where the length of the dictionary entries are recorded) and records in the DIST array (which contains six 110-word segments), a tally for each length in each sector.
- j) DIST.ZERO. This routine zeros out all entries in the segment of the DIST array which has a sector that is full.
- k) SRCH.FREE. This routine searches the FREE array, and where there is no free space in a sector, it calls DIST.ZERO for that sector.
- l) SRCH.SECTS. This routine looks for an entry of length NN in the J'th sector.
- m) SRCH.THIS. This routine looks for an entry of length NN in the I'th sector.
- n) FIXUP. This routine adjusts the STRT and LNG tables when an entry is extracted by XTRCT.CMD.
- o) ADD.FREE. This routine adjusts the FREE array when more free storage becomes available.
- p) SUB.FREE. This routine adjusts the FREE array when less free storage results when STR.CMD is used.
- q) PRINTSTAT. This routine prints the summary status of the sectors.
- r) END.GAME. This routine moves as many entries as possible from sector 6 to sector 5 after the other compaction is done.

3.8.10.2 XTRCT.CMD

First this routine reads the J'th sector from the disc. Next this sector is unpacked into 512 bytes. Then an entry is extracted from the unpacked array. The entry extracted has length NN. The command is placed in the array CMD. Next subroutine COMPACT is called to compact the space vacated by the extracted command. Next the unpacked array is packed and written back to disc. FIXUP is called to update STRT and LNG. If the entry of length NN isn't found, an error is noted, and control is returned to the calling program.

*

* 3.8.10.3 STR.CMD

This routine inserts a dictionary entry at the end of the J'th sector. First the J'th sector is read from the disc and the 256 words are unpacked. Then the STRT array is searched for the first open space so that the entry in CMD can be inserted. The entry is then added and the book keeping entries at the beginning of the sector are updated. Next the byte array is packed and written back to disc. If there is not room to insert the entry an error return is taken.

3.8.10.4 DECODE.SECTOR

This routine is the one where the dictionary is read. The arguments supplied are the 256 word array which constitutes a sector, which sector it is, and an indicator as to the output device (GD or DEC-writer).

The first thing this routine does is to unpack the 256 word array into a 512 byte array. Next the control words at the beginning of the array are listed. These contain the number of commands in a sector, as well as the number of free bytes in the sector. In addition, the first sector contains the total number of commands in all six sectors.

There are 3 types of entries permitted in a dictionary sector. These are 1) Command Symbols, 2) Macros, 3) Menus. Each entry has a 2 byte header. This header contains a code which tells which of the 3 types the entry is. The routine branches to 1 of 3 places to decode the bytes which follow according to the proper pattern.

When all the commands contained in the sector are decoded, return is made to the calling program.

*

3.9 Hardware Configuration Required

UCMD 82 executes on an Applicon Graphics System using a Digital Equipment Corporation PDP-11 minicomputer. Peripheral devices required for input/output operations are a graphics device and a keyboard or a DECwriter.

3.10 Language Used

UCMD 82 is written in BCPL and assembled with PDP-11 MACRO. The support modules for the FIRST sub program, the user command table and sub program UCMD82.MAC are written and assembled in PDP-11 MACRO.

3.11 Operating System Used

UCMD 82 executes in the user command environment of an Applicon AGS/870 Operating System of version 7.3 or greater.

3.12 Type of Program

UCMD 82 is composed of a main program and sub programs which execute under the Applicon Operating System.

3.13 Special Routines Used

Special routines used within UCMD 82 are linked from the Applicon Library Modules LIB762.OBJ. In addition various globals are obtained from files COMMON, DADEF, and MSSDEF. The BCPL interfaces for the file structure require routines AFILE, IOS, and OFILE. The BCPL file support libraries are AGSFIL and BCPLIB. The basic BCPL functions are contained in the object module RES762, AGSLIB and LLIB. All of the above files are in UIC [1,1].

3.14 Form of Program

UCMD 82 is an absolute load module, formatted for execution on the AGS System by the utility LOVALL. Because of the nature of this user command, it should be resident on the system disc.

*

3.16 Program Flow Chart

*

*

3.17 Program Listing

A current listing should be obtained from the 9624 Software Coordinator.

3.18 Backup Capability

The following DOS PIP directory shows all of the files necessary to rebuild UCMD 82. Included in this list is a file named UCMD82.BAT. This file contains all of the DOS/BATCH commands to assemble, link, and LOVALL the UCMD 82 program. The current backup tape and version should be obtained from the Software Certificate.

DIRECTORY DD0: [300,14]

19-APR-77

AGS182.MAC	4	11-MAR-77	<233>
COMPAC.BPL	58	19-APR-77	<233>
UCMD82.BAT	1	11-MAR-77	<233>
UCMD82.MAC	1	11-MAR-77	<233>
UCMT82.MAC	1	11-MAR-77	<233>

4.0 Record of Changes to the Program

See Software Certificate

5.0 Key Words

Applicon Graphic system, Compaction of Dictionary.

*
6.0 References

- a. Macro User Command Interface Option AGS/872 System Reference Manual, Version 3.000, Applicon Incorporated, Burlington, Massachusetts, February 1976.
- b. DOS/BATCH Handbook, DEC-11-ODBHA-A-D, Digital Equipment Corporation, Maynard, Massachusetts, April 1974.
- c. PDP-11 Processor Handbook, Digital Equipment Corporation, Maynard, Massachusetts, 1975.
- d. T. R. Glauner, MADDS User Guide AGS/700, Sandia Laboratories, Org. 9623, September 1976.
- e. J. M. Hudson, BCPL User's Manual, August 1, 1974.

*

7.0 Distribution

The Bendix Corporation
Kansas City Division
P.O. Box 1159
Kansas City, Missouri 64141
Attn: D. R. Lambeth, FD42
For: J. H. Menatt

0323 A. O. Schuknecht
8430 W. A. Little
9620 R. W. DeVore
9621 R. A. Richards (2)
9621 N. J. Nelson
9624 Gino Carli
9624 W. C. Rurd
9624 D. A. Oliver
9624 D. K. Robbins
9624 G. R. Urish
9624 K. E. Wiesandt
9624 File (5)
9624 Software Coordinator
9636 T. R. Glauner

8266 E. A. Aas (2)
3141 C. A. Permueller (5)
3151 W. L. Garner (3)
For ERDA/TIC (Unlimited Release)

ERDA/TIC (25) (R. P. Campbell, 3171-1)