1401 GENERAL PROGRAM LIBRARY Core-to-Card Dump 1.4.148

DISCLAIMER

Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the contributor, in connection therewith.

Core-to-Card Dump

Kenneth W. Dritz
1007 Seven Springs Road
Spartanburg, South Carolina
29302

DECK KEY

1. Object Deck - 40 cards

PROGRAM ABSTRACT

Title: Core-to-Card Dump

Subject Classification: General utility program

Author: Kenneth W. Dritz
M.I.T., Class of 1966

Direct Inquiries to: Mr. Kenneth W. Dritz
1007 Seven Springs Road
Spartanburg, South Carolina
29302

Purpose/Description: This forty card program dumps the contents of all used storage outside the read area onto cards in a condensed, reloadable format, preceded by clear storage cards. Its main use is to enable the operator to stop the execution of a program, save the contents of storage at the time, get off the computer, and continue the execution of the program at any later time. It can also be used to condense programs or to generate object decks for typed-in programs. The advantages of chaining and indexing have been exploited.

Method: The dump program considers characters and word marks in separate phases in order to simplify the assembly of output records and minimize the storage used by this program. Character output cards contain up to sixty-six digits, and word mark output cards cause twenty word marks to be set. Output cards are sequence numbered. The dump program occupies no storage outside the read area until a small amount of storage has been dumped in intermediate form, to be converted to final form and integrated with the main output in a subsequent phase.

Restrictions: None.

Storage Requirements: The program has been written for an 8K machine. It can easily be modified for any capacity machine. The highest core occupied by the program is 539.

Equipment Specifications: One IBM 1401 of any storage capacity. One 1402 Card Read-Punch. The following special features--indexing, store A- and B-address, and modify address.

Additional Remarks:
Language: Written in machine language.
Running Time: Less than a minute average.
Programming Hours: About twenty.

PROGRAM WRITEUP

CORE-TO-CARD DUMP

I. Purpose

A. The Core-to-Card Dump program is designed to dump the contents of core storage onto cards without the use of tapes or disks. It is capable of dumping every character outside of the read area. Large areas of blank storage are not dumped. The dumped information is arranged on the output cards in condensed format with reloading instructions. Thus, the output is a condensed object deck, preceded by clear storage cards, containing the contents of storage at the time it was dumped.

B. The primary use of the Core-to-Card Dump program is to enable the operator to stop the execution of a program, save the contents of storage at the time, get off the computer, and continue the execution of the program at any later time. It has found other uses as a condensing program and a program to generate an object deck for a typed-in program.

II. Functioning Method

A. Use of Program

1. Load read feed with dump program.
2. Load punch feed with blank cards.
3. Press start reset.
4. Press load.
5. When reader stops at last card, place the 13 cards (intermediate output) from the normal punch stacker in the read feed.
6. Press start.
7. When reader stops at last card, press start.
8. When computer halts, remove dump from center stacker.
9. Separate and discard intermediate output from behind dump program in normal read stacker.

B. Output

1. Clear storage cards
2. Bootstrap I
3. Character cards
4. Bootstrap II
5. Word mark cards
6. End card (see End Card Option, below)

C. End Card Option

1. In all cases the dump program will produce an end card which, when read during loading, will cause a clear-storage-and-branch operation to be executed. Normally, this instruction will be ∠333080. If a branch to 333 is not desired, this card can be discarded. If no end card is used when a dump is reloaded, the read area will, of dourse, have to be cleared manually and the program started manually.
2. Any other seven-character instruction (such as a clear-storage-and-branch to a different address, or a halt-and-display-address) will replace the instruction ∠333080 if a card, with the seven charac-

ters punched in columns 56-62, is placed in the read feed immediately following the 13 cards of intermediate output (see step 5, above). In the final phase of the dump program, the last card indicator is tested to determine if such a card is present. If one is <u>not</u>, /333080 will appear on the end card. If a card <u>is</u> present, it is read and columns 56-62 are transferred to the end card. When used as a condensing program to condense an SPS object deck, the SPS end card should be used for this purpose.

D. Additional Notes on Use of Program

1. When used to get off the computer:
   a. Use the address-stop mode to stop the computer at a convenient instruction determined by consulting a symbolic listing, and make a note of this address. Information in the punch area will be saved, but information in the read area will, of course, be lost during dumping and will have to be restored manually to the read area when it is desired to resume execution by loading the dump. In order to minimize the information that will have to be restored manually to the read area, choose to stop the computer at a point in the program where the information in the read area is no longer needed, such as one or two instructions before a new read operation. Then, only word marks in the read area will have to be restored manually. Their locations should be determihed ny printing the contents of the read area in the storage print-out mode before dumping.
   b. A halt will probably be desired after loading so that word marks in the read area may be restored manually. As noted above, this halt may be generated automatically in the dump by placing a card pre-punched with an instruction like .000000 in columns 56-62 behind the intermediate output. Or, one might use /001080, which will cause a halt with a blank op-code after loading. The advantage of /001080 is that word marks left over from loading will be cleared automatically.
2. When used to condense an objeck deck:
   a. Any type of object deck may be transformed to the condensed format (see Listing of Sample Output) of the Core-to-Card Dump. The object deck should be loaded, but it should be prevented from executing any instructions after loading by either removing the end card or loading in the address-stop mode. Then dump as described above.
   b. As noted above, the proper end card instruction should be pre-punched in columns 56-62 of a card, and this card placed behind the intermediate output. An SPS end card is of the proper form for this use. If the program to be condensed begins execution at 333, this step is not necessary.

E. Method of Operation of Core-to-Card Dump

1. General Scheme (not in executive order)
    a. Characters are to be dumped first, without disturbing word marks.
    b. Word marks are to be dumped (by this is meant set-word-mark instructions are to be generated in the output with A- and B-addresses corresponding to the addresses of word marks in core storage being dumped).
    c. In order to dump everything outside the read area, some storage must be dumped in intermediate form before a. above.
    d. The intermediate output is to be read in, converted to final form, and merged with the main output at the appropriate times in a. and b. above.
2. Sequence of Routines
    a. Card 1 prepares the read area with 29 word marks and punches the contents of the punch area (this is the first intermediate output card).
    b. Cards 2 and 3 move the numerical and zone portions, respectively, of the characters in 081-100 to 101-120.
    c. Cards 4 and 5 move the zone and numerical portions, respectively, of the characters in 181-240 to 121-180. Card 5 punches the second intermediate output card. Indexing is now used, because the characters in the index registers have been dumped. No word marks have been disturbed.
    d. Cards 6 and 7 move the numerical and zone portions, respectively, of the characters in 241-320 to 101-180. Card 7 punches the third intermediate output card.
    e. Card 8 rearranges word marks in the read area for the subsequent routine.
    f. Card 9 causes word marks in 081-160 to be "dumped" in intermediate form. For each word mark in this area, an ampersand (12-punch) is moved to the corresponding location (101-180) in the punch area. For locations not containing a word mark, blanks are moved to the corresponding locations in the punch area.
    g. Cards 10-14 continue the "dumping" of word marks from 161-560, 80 per card. Cards 9-14 cause intermediate output cards 4-9 to be punched.
    h. Card 15 causes the read area to be cleared.
    i. Cards 16 and 17 load the duplication phase into 182-227.
    j. The duplication phase is executed. This phase duplicates the clear storage cards (cards 18 and 19) and Bootstrap I (card 20) into the center stacker (final output). The phase next duplicates cards 21-24 into the normal punch stacker, behind the intermediate output cards. Cards 21-24 are the main word mark phase, which will be read in and executed after the intermediate output cards

are converted to final form.

k. Cards 25-29 load the main character phase into 081-099 and 182-308.

l. The main character phase is executed.. Characters from 321 to the end of storage are dumped. There are two loops. Loop 1 examines increasingly higher locations for blanks. When a non-blank character is found, loop 2 is entered. Loop 2 moves the numerical and zone portions, separately, of the next 66 characters (including blanks, if any) to the punch area (unless the end of storage is reached), then assembles the loading instruction and punches the card into the center stacker. Control returns to loop 1. When the end of storage is reached, exit will be from loop 1.

m. Cards 30-40 load the character conversion phase, Bootstrap II phase, and word mark conversion phase into 081-099 and 182-539.

n. At this point, a last card halt allows the intermediate output to be placed in the read feed.

o. The character conversion phase is executed. This phase reads in the first three intermediate output cards and converts each to two cards in final form, punching the latter into the center stacker.

p. Bootstrap II phase is executed. This phase assembles and punches into the center stacker the second bootstrap card, and prepares the punch area with word marks for subsequent routines.

q. The word mark conversion phase is executed. This phase reads and examines the six intermediate word mark cards for 12-punches, assembling set-word-mark instructions with the proper addresses in the punch area, punching cards into the center stacker whenever the punch area is "full."

r. The main word mark phase, which has been duplicated, is loaded into 182-313.

s. The main word mark phase is executed. This phase examines cores from 561 to the end of storage for word marks, and continues the assembly of output cards where the word mark conversion phase terminated.

t. An end card is produced as described above.

u. A blank card is punched and selected into stacker 4, causing the end card to drop into the center stacker.

v. The dump program halts, displaying an I-address of 313.

III. Limitations

A. Location 100 cannot be dumped, and all dumps will show a zero in location 100, regardless of what was there. This zero is a normal consequence of the punch operation.

B. As in any program, ¢ (A-bit) cannot be punched.

C. The clear storage cards must be retained in the dumped deck, since the dumped deck loads in the move mode.

IV. Reasons for Methods Employed

A. Separate character and word mark phases simplify assembly of output records.

B. The final phase is duplicated behind the intermediate output to eliminate the need for inserting the intermediate output between phases.

C. Multiple-purpose constants, constants in the B-address of store B-address instructions, chaining wherever feasible, and op-code modification have been employed for efficiency.

D. In order not to unduly lengthen the routine, no attempt was made in the character conversion phase to suppress the final form cards for 081-320 if the intermediate output cards were blank. These cards will be the last six before Bootstrap II and can be removed from the dump if they will only load blanks.

E. The punch area is not cleared between the assembly of consecutive word mark cards. Word mark addresses are assembled from right to left. If the last word mark card is incomplete, previous valid word mark addresses will still be in the punch area. No problems will be encountered, even if fewer than 20 word marks afe in storage.

F. For the sake of compactness, the dump program was written in machine language and the cards arranged in condensed format. Serial numbers have not been punched, and it is suggested that the cards of the Core-to-Card Dump be numbered with a Magic Marker.

V. Storage Requirements

A. The dump program was written for an 8K machine.

B. The dump program may be modified for any capacity storage by changing the address I9Z appearing on cards 19, 25, and 40.

VI. Output Card Forms (not including clear storage cards)

A. Bootstrap I (card 001 in Listing of Sample Output)

Columns 1-11: ,0080121001
Columns 12-77: blanks
Columns 78-80: 001 (serial identification)

B. Main character cards (cards 002-028 in Sample Output)

Columns 1-11: M077xxx1001
Columns 12-77: dumped characters
Columns 78-80: serial identification

Note: When xxx is the highest core (e.g., I9Z), the A-address in the move instruction may be less than 077.

C. Converted character cards (cards 029-034 in Sample Output)

These six cards have the following format:

Card	Columns 1-11	Dumped characters in	Blanks in
1	M0771661001	Columns 12-77	-----
2	M0251801001	Columns 12-25	Columns 26-77
3	M0311001001	Columns 12-31	Columns 32-77
4	M0712401001	Columns 12-71	Columns 72-77
5	M0773061001	Columns 12-77	-----
6	M0253201001	Columns 12-25	Columns 26-77

Serial identification is in columns 78-80. Note: Card 3 will have a zero in column 31.

D. Bootstrap II (card 035 in Listing of Sample Output)

```
Columns  1-14:   ■012012,015022
Columns 15-42:   ,029036,043050,057064,071075
Columns 43-70:   ,029036,043050,057064,071075
Columns 71-74:   1001
Columns 75-77:   blanks
Columns 78-80:   serial identification
```

E. All word mark cards (cards 036-050 in Sample Output)

```
Columns  1-70:   10 set-word-mark instructions
Columns 71-74:   1001
Columns 75-77:   blanks
Columns 78-80:   serial identification
```

F. End card (card 051 in Listing of Sample Output)

```
Columns   1-7:   /333080, or other as described above
Columns  8-77:   blanks
Columns 78-80:   serial identification
```

VII. Equipment Specifications

A. One IBM 1401 with indexing, modify address, and store A- and B-address. Any storage capacity.

B. One IBM 1402 Card Read-Punch.

VIII. Testing

A. The Core-to-Card Dump has been tested as a condensing program on several decks.

B. The present version includes one generation of improvements and modifications over the original.

C. The author would appreciate notification of further individual improvements and modifications by users.

```
,008015,022029,036043,050057,076080,058059,060061,075066,,,,,,,,,NNNNNNNNNNN5050
                                                        D100120DDDDDDDDDDDDDDDDDDD1050
                                                        Y100120YYYYYYYYYYYYYYYYYYY1029
1036I9CM004079B036                 H089000H0890S0#007079Y1Y01S0YYYYYYYYYYYYYYYYYYYB022
5036I9CM004079B036                 H089000H0890S0#007079D1Y01S0DDDDDDDDDDDDDDDDDDDB022
1036I9CM004079B036     B036        H089000H0890S0#007079D2U01‡0DDDDDDDDDDDDDDDDDDDB029
5022I9CM004079B036     B036        H089000H0890S0#007079Y2U01‡0YYYYYYYYYYYYYYYYYYYB029
                     ,002009,017024,028035¤008015¤022029¤036036NNNNNNNNNNNNNNNNNNN1002
5H089001V0280Y01M0581‡0B035M0571‡0B0010888#008089B009   &
5H089001V0281W01M0581‡0B035M0571‡0B0010888#008089B009   &
5H089001V0282U01M0581‡0B035M0571‡0B0010888#008089B009   &
5H089001V0283S01M0581‡0B035M0571‡0B0010888#008089B009   &
5H089001V0284‡01M0581‡0B035M0571‡0B0010888#008089B009   &
5H089001V0284Y01M0581‡0B035M0571‡0B0010888#008089B009   &
 ,016190L031189L/182099,0011001
,008015,022029,036040L080222,1861871001,0011B2081801M0801804K1868B0011802M080180
L043226,195202,203208,216227,044051B0445208,055062/180/182080
,008015,022026,030037N000,044,049053,069072N00001026
L068116,105106,110117B101/I9Z#071029C029074B026/B001/0991,001/001117I0&099
,0080121001
                                                                                   001
,008015,022029,036040L077219,1901971001V2130&01C099539B256S#084099B182M0990‡0
L075255,202209,213224,231236,2372441001H089C089083B190/4A081180H089170K1908
L078294,251263,268269,271278,2822871001C089536B278S4K8A081180/177B295A1M062533
L058313,288302,303304,306307,309313B182M5331074/K84K4.309.                         002
,008015,022029,036040L058099,1780851001100177000I9ZM077320
L080222,087090,093189,196203,2082151008M084111H089012C099092B022S#084099B1960&0
       L077260,230237,244249,2562561008D0&01‡0Y0&01‡0C089086B286SC099092B279S
       L074295,268275,279286,2932941015#084099#084089B223M089096M0991074K8
              L052308,303304/0361771182A081180/B182
                     L058099,0850961001100156170000010,081
L080222,186193,197201,205206,2082151001,112M066177M084M526Q2004K8A081180/M442192
L080263,216227,234238,242249,2532541001Q219M407238Q2301182L084174L484LLLLLLLL484
L077301,255256,257258,259260,2652661001LLLLLLLLLLL4K8A081180L114107L1V3280-08
L076338,267268,269270,271272,2732741001#084099B3710938#084094B294M0990‡0H089
L078377,275276,278285,292293,2943091001C089083B302/4A081180H089170K3028C099087
L076414,317324,328335,346351,3523591001B394SH0940011294,0011001NB1B18M080125
L078453,366371,383390,394398,4024081001M066177M080171M020131M080125¤012012,015
L077491,422429,436443,447450,4574611001022,029036,043050,057064,071075M025320
L074526,464468,471475,478482,4854991001M077306M071240M031100M025180M077166
L052539,506513,520534,537537/238080    /333080170I9Z
```

Listing of Core-to-Card Dump

```
,008015,022026,030037N000,044,049053,069072N00001026
L068116,105106,110117B101/I9Z#071029C029074B026/B001/0991,001/001117I0&099
,0080121001                                                                 001
M0773861001L0521521K1L074J811K1M028-67M066-781K1M080166M-101804LJ81174,179AZ5002
M0774521001417948З991K1C056Z57B438/C016Z71B396/,018L020X91B445LZ74X91MX91-06A003
M0775181001Z77-06LX91J00#Z77X91¤X89/176MY05163AZ541794/176MY34129AZ541794/180004
M0775841001C056Z57B537/1K5191SZ54J00Y-48J00C056Z58B‡77/B913¤067063M074-97MZ83005
M0776501001062LZ58056,067024MV66066LZ59063B056CZ86Z83B709U/0801K1B636056NC056006
M0777161001Z608551/B697015X,057M059Z35AZ54Z62MZ621804BT36M%T0V76W.705H912CZ89007
M0777821001Z83B860SLZ83J03AZ55J03MZ83760MJ03763,179L000000AZ54Z62MZ62180¤1794008
M0778481001/180C-22-06B849USZ80Z83Y-48Z83MJ03830MZ83833M000000¤181AZ80-06B909009
M0779141001MV69U88BT36AZ54Z62MZ621804/180C-22-06B849USZ80Z83Y-48Z83B838B000M-010
M0779801001270O5#007J00C062J00B976SL-30J07#Z92J00AZ54J07C062J00B987/B‡14#007Z011
M077‡46100183B567CZ95J00B946/M%T0W35W.111111AZ54Z83MZ83‡34M-47000SZ54J07Y-48J012
M077/12100107CJ07-30B976SCZ86Z83B709UB‡14AZ54Z64M007-50M062-53CZ92-50B/35SSZ5013
M077/7810014-50Y-48-50#-34-53B‡98CZ92Z64B/82/#-37V63MV63/67L-07000#-43V63AZ98014
M077S441001-02MV63/95M-53000BS57-312MV63S24#-40S24M-53000MZ55-31#-40V63SZ56-0015
M077T1010012Y-48-02B913CZ66Z64BS80SMZ54-31BS32C-26-02BT18TMV69U88AZ54Z62MZ621016
M077T761001804BT36M-28Z64MZ54-31B913/180MY83166MX59117M-141804/176MZ08125,179017
M077U421001AZ541794/180MV72U02BU37000 MV75U14M0001804#-46V72#-46V75BT89/180MZ018
M077V08100131136MX55113M-181804/176MZ53136,179AZ54179BU894V41MZ54133/1214/136019
M077V741001M-67128MZ68128M-78166AZ541794237/1804M%T0X31W.999999J06620U93J82K6020
M077W4010011CANNOT PROCESS EXECUTE CARDS. PROGRAM CANNOT BE CONDENSED. REMOVE021
M077X071001AND REARRANGE DECK SO THAT LOAD LOCATIONS ARE IN ASCENDING ORDER. 022
M077X731001MAY NEED NEW ORG CARD. END OF JOB. ,008015,022026L063I7Y1001/080,0023
M077Y391001011B001080*M078   #I7YI6UBI4Z078,I4VI4Z,I5‡I5Y,I6VI7S,I7WBI4/7I9Z1024
M077Z051001001,075M080I7V,      BI4/074 ¤075L074080BI5Y,I6SI6Z,I7WI8U,I8YI9VB025
M077Z711001I4/9,033037L066I9Z¤I7‡1022/   080L065I6Z¤I4/B015123NMB/0000265ORG026
M077-371001333077078100177178001I9Z1552106000O*A1**B1**C1*78617843000I1I9I076027
M077J031001I9G079080                                                        028
M0771661001                                                                 029
M0251801001                                                                 030
M0311001001                 0                                               031
M0712401001                                 AZ54Z99B289AZ55Z99B297L-97      032
M0773061001074¤056024,001L0801804BV41A1K2551MV63-02V215-00SV226-00KY-48-02Y-4033
M0253201001 8-001K1,001076                                                  034
¤012012,015022,029036,043050,057064,071075,029036,043050,057064,0710751001  035
,314312,311304,297289,281274,269268,263262,255251,244237,233226,2222151001  036
,397396,392391,384380,373372,365358,356355,348341,339338,331329,3283211001  037
,507500,496495,488481,477473,466459,452445,438434,427423,418411,4063991001  038
,620616,609602,595588,581574,567563,558551,544537,532531,526519,5155141001  039
,725720,713709,705697,693692,685678,671667,659654,647639,637636,6326271001  040
,845838,834827,820813,806799,794787,783782,778771,764757,753746,7397321001  041
,960953,946939,934927,920913,909905,898891,886879,875874,867860,8568491001  042
,‡84‡77,‡73‡68,‡61‡56,‡49‡42,‡35‡28,‡21‡14,‡07999,994987,983976,9729671001  043
,S18S11,S04/96,/89/82,/75/68,/61/54,/47/42,/35/31,/24/17,/10/05,‡98‡911001  044
,T36T32,T25T18,T14T13,T06S99,S92S87,S80S76,S69S64,S57S53,S46S39,S32S251001  045
,U41U37,U33U26,U19U18,U11U04,T96T89,T85T84,T77T73,T66T62,T61T54,T47T401001  046
,V45V41,V37V30,V23V16,V09V05,V04V00,U93U89,U85U78,U74U67,U63U62,U55U481001  047
,Z57Z56,Z55Z54,Z32Z09,Y84Y35,Y06X43,X42X30,W34V73,V70V67,V64V61,V54V461001  048
,-07-03,Z99Z96,Z93Z90,Z87Z84,Z81Z78,Z75Z72,Z69Z67,Z65Z63,Z61Z60,Z59Z581001  049
,-07-03,Z99Z96,Z93Z90,Z87Z84,Z81-79,-68-54,-51-49,-47-27,-23-19,-15-111001  050
/333080                                                                     051
```

Listing of Sample Output
This program occupies 2000 cores.

IBM

Data Processing Division
112 East Post Road
White Plains, New York

International Business Machines Corporation

WHite Plains 9-1900

March 9, 1964

Mr. Kenneth W. Dritz
1007 Seven Springs Road
Spartanburg, South Carolina

Dear Mr. Dritz:

Thank you for your letter of February 10, 1964 submitting data on the program entitled, "Core-To-Card Dump" for the IBM 1401 Data Processing System, to be listed under File Number 1.4.148.

We have received an abstract, writeup, listing and card deck for this program.

Your program will be available from the Program Information Department by March 30, 1964 and will be announced in the next edition of the Catalog of Programs for IBM Data Processing Systems or its Supplements.

To insure currency of the program libraries, the Program Information Department monitors requests for each program. If a program remains at a minimum activity level, the author is consulted concerning the advisability of retaining the program as a part of the Program Information Department's Library.

We should like to congratulate you for your significant addition to the 1401 Program Library. We should like to encourage you to continue your efforts for and contribution to the programming and systems community.

Very truly yours,

F.A. Marks Jr

F. A. Marks, Jr., Manager
DP Program Information Department

FAM/JFS/ao

IBM

International Business Machines Corporation

40 Saw Mill River Road
Hawthorne, New York 10532
914/592-5790

September 22, 1967

Mr. Kenneth W. Dritz
1007 Seven Springs Road
Spartanburg, South Carolina

Core-To-Card Dump 1401-01.4.148

Dear Mr. Dritz:

To insure a library of current programs, the IBM Program Information Department monitors the number of requests made for each program. When the number of requests for a particular program falls below an established level, that program is removed from the list of programs available from the Library.

Your program has made a significant contribution to the programming community. The current number of requests, however, is below the level needed to maintain it in the Library. Hence, this program is being removed from the next issue of the IBM Catalog of Programs.

Thank you for making this contribution, and we hope that you will continue to make your programs available.

Sincerely yours,

F. A. Marks, Jr., Manager
Program Information Department

FAM/mc