Chapter 3

COMPILER OUTPUT AND DEBUGGING

COBOL DUMPS

CONTENTS

INTRODUCTION 3

IDENTIFYING THE PROBLEM 3
JCL ERRORS 4
ABEND CODES 4
User Error Codes 4
System Error Codes 5

TRACKING DATA ERRORS 6
CALCULATING AN INSTRUCTION ABSOLUTE ADDRESS 9
LOCATING THE COBOL SOURCE CODE LINE NUMBER 11
USING MAP for Data descriptions 13
Calculating Working Storage Data Addresses 14
USING XREF 14

IBM 370 COBOL DUMPS 16
LIST INFORMATION 16
TGT INFORMATION 17
LKED INFORMATION 19
FDUMP INFORMATION 19
DUMP INFORMATION 20

COMPILER OUTPUT AND DEBUGGING
COBOL DUMPS

INTRODUCTION

While specific methods do exist for tracing logic errors, there is no one preferred technique -- indeed, no guaranteed sequence of steps or foolproof debugging aide -- for determining what caused any specific error. There is, however, a structured approach to locating and solving computer program logic errors. That is not to say that what follows is the best, or even the only, method. It does have the advantage of working under most conditions.

The main purpose of this chapter is to identify and define some of the available tools and information provided in our environment for solving COBOL ABENDs. Therefore, it is an attempt to provide a framework for your investigation, recognizing that every problem is unique.

IDENTIFYING THE PROBLEM

One of the first steps toward identifying the cause of an ABEND is locating and interpreting any error code (or codes) given by the system. This code can appear in an on-screen message, as part of an output file, or as a system message. While the error code is indicative of the ABEND cause, it is only the first step in locating the true error. Along with the ABEND code, the system will frequently provide the contents of the operating system's addresses (also known as the contents of the General Purpose Registers) and the controlling memory address (as part of the Program Status word). These can frequently be used in conjunction with the assembly code listing to locate the exact instruction causing the ABEND -- coupled with a knowledge of data storage structures and assembly code, this can lead to problem solutions which affect file or database structures, as well as program logic corrections. However, while this technique will identify the error location and narrow the search for a correct fix, not every program ABEND requires this much effort to locate.

One method available to determine the program line number related to the error, if available, is the COBOL complier option FDUMP. This parameter often provides the line number where an ABEND occurred. There is one caution attached to confidently using this value: Report Writer and other precompilers (database precompilers, SQL, etc.) add lines to the compliable code, without adding these lines to the printed source code. When this occurs, the line number indicated by the FDUMP will be inaccurate, and could cause major problems in locating the ABEND.

If the error occurs outside of COBOL program code, within the transfer between subprograms, inside Report Writer or database code, or within an Assembly code subprogram, it is more difficult, but not impossible, to determine the location command causing the ABEND. Additional complier parameters, useful in tracking down the original source of the error, will be presented in a later section.

Error codes are divided into System ABEND errors, occurring as a result of incorrect program commands or data definitions, and JCL errors.

JCL ERRORS

Some of the least complicated JCL errors to correct occur due to inaccurate estimation of printed lines or CPU time requirements. The system is designed to allocate a small extention of resources prior to ABEND. Before simply increasing the request of time or printed lines, determine if the resource estimates were exceeded due to an ABEND condition, or an infinite loop. Look at the JCL execution listing, and the program output for potential errors before resubmitting.

When an error occurs during JCL compilation, the line number of the error is generally found at the beginning of the compilation listing. It may indicate an incorrect clause in a data definition statement, or incorrect punctuation. Check to see if the lines involved in the definition are in the correct format. Verify the file attributes of fixed length, 80 columns, in the original submit.

Another possibility is that JCL data files are defined under an incorrect step, or in the improper order. Examine the JCL compilation expansion for proper placement of data definitions and file replacements.

ABEND CODES

Program execution ABEND codes are separated into System errors, usually involving data operations, and User errors, typically involving files. These error codes are prefixed with S and U, respectively. In most cases, the error code is listed in the JES2 log on the first page, or as an interactive message.

User Error Codes

Most of the User error codes encountered in this course are caused by incompatible or incomplete file definitions (U1035, U1020). When one of these errors occurs, a short definition can be found in the Application Programmer's Guide, Appendix C. Appendix D of the Application Programmer's Guide provides possible causes for these types of ABENDs. This assists in the identification of specific types of errors

and their possible solution. It is often necessary to determine which file or files were involved.

When a file related error occurs, compare the definition given in the program code to all available external descriptions of that file. These descriptions can be found in the JCL for the program, JCL used to create that file, and prior documentation. When an inconsistency is located, remember that it could have been caused by an external change in the data file definition. Double check all file structures, and establish consistency between all definitions, remembering to examine variables for changes in length or type.

If the file definitions are consistent, examine the logic to see if an OPEN, CLOSE, READ or WRITE statement exists for the wrong file, or if there is a read attempted after End-Of-File. One common error occurs when a STOP RUN statement fails to execute through improper logic, and a sequential file is issued another input/output command. Attempting Input/Output after a file is closed, frequently through the use of a TERMINATE command, also causes an error.

System Error Codes

Among the possible system error codes exist a set regarding system execution (Out of Time -- S322, Printed Lines exceeded -- S222, Program not found -- S806). These codes require examination of the JCL, as previously discussed.

Other system error codes are the result of unsuccessful data operations during program execution (Read past EOF -- S001, Address out of Range -- S0C4, Divide by Zero -- S0C9) or incorrect use of variables within a Subroutine. Of the System Error Codes, perhaps the most common found in our environment is the Data Exception -- S0C7.

A Data exception error results from an arithmetic operation (ADD, SUBTRACT, MULTIPLY, DIVIDE or COMPUTE) or comparison (IF statement) performed on a numeric data field that contains an invalid formatted data. Remember that data operations may not be performed on a Zoned decimal (DISPLAY format) field without translating the value into the equivalent PACKED decimal format. This process occurs through a direct command (MOVE) issued by the programmer, or indirectly via compiler generated code. Inproper use of group MOVEs, VALUE clauses, mis-definition of Subprogram LINKAGE SECTION variables, or out of control looping logic can cause variables to be loaded with invalid data.

TRACKING DATA ERRORS

Use of the FDUMP compilation parameter may provide a line number and verb which attempted an invalid operation. Remember that the FDUMP will not provide a proper line number if a precompiler is used. To find a relevant COBOL source code line, implement the following procedure.

First, identify the COBOL program executing when the ABEND occurred. If the ABEND happened in a section of the program that did not require the precompiler, the line number given will be accurate. Otherwise, the JES2 Job Log or the DUMP parameter provides the value of the PSW (Program Status Word). The last six digits indicated the absolute address of the machine code instruction executing at ABEND, or the absolute address of the machine code instruction following the instruction that caused the ABEND. In this example, the absolute address value is 115968.

In the following example, note that the FDUMP parameter was requested at compile time (PARM.COB2=(FDUMP,NODYNAM,XREF,MAP)) in order to provide the listing of variable contents and the line number causing the ABEND. The line number identified by this parameter is verb 2 of line 35 of program XYZSUB (IF... ADD CRASH1 TO CRASH2.). The procedure to calculate the absolute and program relative addresses of the ABEND is illustrated following the FDUMP.

```
                    J E S 2  J O B  L O G  --  S Y S T E M  A C A D  --  N O D E  A C A D M V S
11.20.16 JOB  100  ICH70001I FE16      LAST ACCESS AT 11:19:08 ON FRIDAY, OCTOBER 2, 1992
11.20.16 JOB  100  $HASP373 FE16SUB  STARTED - INIT  2 - CLASS A - SYS ACAD
11.20.16 JOB  100  IEF403I FE16SUB - STARTED
11.20.58 JOB  100  IEF450I FE16SUB GO STEP2 - ABEND S0C7 U0000
11.20.58 JOB  100   PSW at ABEND = 'FF850007E0115968'
11.20.58 JOB  100   Line number or verb number being executed:  '0000035'/'2'
11.20.58 JOB  100   The GP registers at entry to ABEND were
11.20.58 JOB  100       Regs  0 - 3  - '00119754  001197C0  00119AF2  0011492C'
11.20.58 JOB  100       Regs  4 - 7  - '001157A4  0011492C  00158700  00119AF2'
11.20.58 JOB  100       Regs  8 - 11 - '00119AF0  00117290  001157D8  0011585C'
11.20.58 JOB  100       Regs 12 - 15 - '001157D0  001170F8  001158A2  8013E398'
11.20.59 JOB  100  IEF404I FE16SUB - ENDED
11.20.59 JOB  100  $HASP395 FE16SUB  ENDED
------ JES2 JOB STATISTICS ------
```

Absolute Address of Instruction

```
PP 5668-958 IBM VS COBOL II RELEASE 3.0 09/13/88                XYZSUB    DATE 10/03/92  TIME 11:20:20   PAGE     2
  LINEID  PL SL  ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7-|--+----8  CROSS REFERENCE
  000001                *
  000002                *
  000003                *    THIS EXAMPLE SUBPROGRAM HAS BEEN CHANGED TO CAUSE
  000004                *    IT TO TERMINATE ABNORMALLY.
  000005                *
  000006                *
  000007                 IDENTIFICATION DIVISION.
  000008                 PROGRAM-ID.     XYZSUB.
  000009                 AUTHOR.         BCIS.
  000010                 INSTALLATION.   UNT.
  000011                 DATE-WRITTEN.   1992.
  000012                 DATE-COMPILED.  10/03/92.
  000013                *
  000014                 ENVIRONMENT DIVISION.
  000015                *
  000016                 DATA DIVISION.
  000017                *
  000018                 WORKING-STORAGE  SECTION.
  000019                 01  SYSTEM-FLAGS.
  000020                     05  SWS-START       PIC X(09) VALUE 'SWS-START'.
  000021                     05  DUMMY           PIC 9(01) VALUE ZERO.
  000022                     05  CRASH1          PIC S9(01)          COMP-3.                          IMP
  000023                     05  CRASH2          PIC S9(04) VALUE +35 COMP-3.
  000024                     05  SWS-END         PIC X(07) VALUE 'SWS-END'.
  000025                *
  000026                 LINKAGE          SECTION.
  000027                 01  X   PIC 9(03) COMP-3.
  000028                 01  Y   PIC 9(04) COMP-3.
  000029                *
  000030                *
  000031                 PROCEDURE DIVISION USING X Y.
  000032                 000-MAIN-LOOP    SECTION.                                                    27 28
  000033                     ADD X TO Y.
  000034                     ADD 1 TO CRASH2.                                                        27 28
  000035                     IF CRASH2 > 38 ADD CRASH1 TO CRASH2.                                    23
  000036                 010-MAIN-LOOP-EXIT.                                                         23 22 23
  000037                     GOBACK.
  000038                *

INVOCATION PARAMETERS:
LIST,MAP,NOLIB,VBREF,XREF,FDUMP,DUMP,NODYNAM
PP 5668-958 IBM VS COBOL II RELEASE 3.0 09/13/88                EXAMPLE   DATE 10/03/92  TIME 11:20:32   PAGE     2
  LINEID  PL SL  ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7-|--+----8  CROSS REFERENCE
  000001                 IDENTIFICATION DIVISION.
  000002                 PROGRAM-ID.     EXAMPLE.
  000003                 AUTHOR.         BCIS.
  000004                 INSTALLATION.   UNT.
  000005                 DATE-WRITTEN.   1992.
  000006                 DATE-COMPILED.  10/03/92.
  000007                *
  000008                 ENVIRONMENT DIVISION.
  000009                 INPUT-OUTPUT     SECTION.
  000010                 FILE-CONTROL.
  000011                     SELECT CARD-FILE    ASSIGN TO UT-S-CARDIN.
  000012                     SELECT DISK-FILE    ASSIGN TO UT-S-FILE01.                              24
  000013                *                                                                            16
  000014                 DATA DIVISION.
  000015                 FILE             SECTION.
  000016                 FD  DISK-FILE
  000017                     RECORDING MODE IS F
  000018                     LABEL RECORDS ARE STANDARD
  000019                     RECORD CONTAINS 80 CHARACTERS
  000020                     BLOCK CONTAINS 0 RECORDS
  000021                     DATA RECORD IS DISK-REC.
  000022                 01  DISK-REC            PIC X(80).
  000023                *
  000024                 FD  CARD-FILE
  000025                     RECORDING MODE IS F
  000026                     LABEL RECORDS ARE STANDARD
  000027                     RECORD CONTAINS 80 CHARACTERS
  000028                     DATA RECORD IS CARD-REC.
  000029                 01  CARD-REC            PIC X(80).
  000030                *
  000031                *
  000032                 WORKING-STORAGE SECTION.
  000033                 01  SYSTEM-FLAGS.
  000034                     05  WS-START        PIC X(08)    VALUE 'WS-START'.
  000035                     05  EOD             PIC X(03)    VALUE 'NO'.
  000036                 01  TOTAL-AREA COMP-3.
  000037                     05  AMT             PIC 9(03)    VALUE 1.
  000038                     05  TOT             PIC 9(04)    VALUE ZERO.                             IMP
```

```
000039                 01  TOT-LINE.
000040                     05  FILLER              PIC X(30)     VALUE
000041                         '      TOTAL RECORDS PROCESSED ='.
000042                     05  PTOT                PIC Z(04).
000043                     05  FILLER              PIC X(40)     VALUE
000044                         '  THIS VALUE ACCUMULATED IN SUBXYZ'.
000045                 01  DATA-IN.
000046                     05  CARD-IN             PIC X(80).
000047                 01  WS-END                  PIC X(06)     VALUE 'WS-END'.
000048                *
000049                 PROCEDURE DIVISION.
000050                 000-MAIN-LOOP   SECTION.                                          56
000051                     PERFORM 100-INITIALIZE-SYS.                                   60 35
000052                     PERFORM 200-PROCESS UNTIL EOD = 'YES'.                        74
000053                     PERFORM 300-CLOSE-FILES.
000054                     STOP RUN.
000055                *
000056                 100-INITIALIZE-SYS   SECTION.                                     24
000057                      OPEN INPUT  CARD-FILE.                                       16
000058                      OPEN OUTPUT DISK-FILE.
000059                *
000060                 200-PROCESS          SECTION.
000061                 210-PROCESS-READ-RECORD.                                          24 46
000062                     READ CARD-FILE INTO CARD-IN
000063                          AT END                                                   35
000064      1                       MOVE 'YES' TO EOD                                   38 42
000065      1                       MOVE TOT TO PTOT                                    71
000066      1                       GO TO 290-PROCESS-EXIT
000067                     END-READ.
000068                 220-PROCESS-WRITE-RECORD.                                         22 45
000069                      WRITE DISK-REC FROM DATA-IN.                                 EXT 37 38
000070                      CALL 'XYZSUB' USING AMT TOT.
000071                 290-PROCESS-EXIT.
000072                     EXIT.
000073                *
000074                 300-CLOSE-FILES      SECTION.                                     24
000075                     CLOSE CARD-FILE                                               16
000076                           DISK-FILE.
000077                *
```

Formatted DUMP Information

```
 --- VS COBOL II Formatted Dump at ABEND ---                                  PROGRAM-ID executing at ABEND
>>-Program = 'XYZSUB'
  Completion code = 'S0C7'                                                    Absolute Address of Machine Code
>>PSW at ABEND = 'FF850007E0115968'                                                        Line Number
>>Line number or verb number being executed:  '0000035'/'2'
  The GP registers at entry to ABEND were
      Regs  0 - 3  - '00119754  001197C0  00119AF2  0011492C'
      Regs  4 - 7  - '001157A4  0011492C  00158700  00119AF2'
      Regs  8 - 11 - '00119AF0  00117290  001157D8  0011585C'
      Regs 12 - 15 - '001157D0  001170F8  001158A2  8013E398'

Data Division dump of 'XYZSUB'
00019 01 XYZSUB.SYSTEM-FLAGS AN-GR
00020 02 XYZSUB.SWS-START X(9)
      DISP     ===>SWS-START
00021 02 XYZSUB.DUMMY 9
      DISP     ===>0
00022 02 XYZSUB.CRASH1 S9
      CMP3     ===>+0
               HEX>0
                   0
00023 02 XYZSUB.CRASH2 S9999
      CMP3     ===>+0039
00024 02 XYZSUB.SWS-END X(7)
      DISP     ===>SWS-END
00027 01 XYZSUB.X 999
      CMP3     ===>001
00028 01 XYZSUB.Y 9999
      CMP3     ===>0004
```

```
Data Division dump of 'EXAMPLE'
00016 FD EXAMPLE.DISK-FILE FD
FILE SPECIFIED AS:
  ORGANIZATION=SEQUENTIAL    ACCESS MODE=SEQUENTIAL
  RECFM=FIXED BLOCKED
CURRENT STATUS OF FILE IS:
  OPEN STATUS=OUTPUT
  QSAM STATUS CODE=00
00022 01 EXAMPLE.DISK-REC X(80)
      DISP     ===>RECORD NUMBER 4
00024 FD EXAMPLE.CARD-FILE FD
FILE SPECIFIED AS:
  ORGANIZATION=SEQUENTIAL    ACCESS MODE=SEQUENTIAL
  RECFM=FIXED
CURRENT STATUS OF FILE IS:
  OPEN STATUS=INPUT
  QSAM STATUS CODE=00
00029 01 EXAMPLE.CARD-REC X(80)
      DISP     ===>RECORD NUMBER 4
00033 01 EXAMPLE.SYSTEM-FLAGS AN-GR
00034 02 EXAMPLE.WS-START X(8)
      DISP     ===>WS-START
00035 02 EXAMPLE.EOD XXX
      DISP     ===>NO
00036 01 EXAMPLE.TOTAL-AREA AN-GR
00037 02 EXAMPLE.AMT 999
      CMP3     ===>001
00038 02 EXAMPLE.TOT 9999
      CMP3     ===>0004
00039 01 EXAMPLE.TOT-LINE AN-GR
00040 02 EXAMPLE.FILLER X(30)
      DISP     ===>      TOTAL RECORDS PROCESSED =
00042 02 EXAMPLE.PTOT ZZZZ
      DISP     ===>****
               HEX>0000
                   0000
00043 02 EXAMPLE.FILLER X(40)
      DISP     ===>  THIS VALUE ACCUMULATED IN SUBXYZ
00045 01 EXAMPLE.DATA-IN AN-GR
00046 02 EXAMPLE.CARD-IN X(80)
      DISP     ===>RECORD NUMBER 4
00047 01 EXAMPLE.WS-END X(6)
      DISP     ===>WS-END
 --- End of VS COBOL II Formatted Dump at ABEND ---
```

When the program ABENDs and a source code line number is not immediately available, it is possible to calculate the location of the instruction relative to the beginning of the compiled program code. In order to determine this location, the LIST parameter must also be included in the program compilation options (PARM.COB2='FDUMP,NODYNAM,MAP,XREF,LIST,DUMP). Employmentof this parameter results in a series of outputs, as follows: Main program initialization (INIT 1):

```
PGM Rel
Address Machine Code    Assembly Code Instruction                Programming Comments
000000                  XYZSUB   DS   0H
                                 USING *,15
000000  47F0 F070                B    112(,15)                    BYPASS CONSTANTS. BRANCH TO @STM
000004  23                       DC   AL1(35)                     SIGNATURE: LEN. PGM. I.D. CHARS.
000005  E7E8E9E2E4C24040         DC   C'XYZSUB  '        @EPNAM:          PROGRAM NAME
000000D 40C3F240                 DC   C' C2 '                             COMPILER = COBOL II
000011  F14BF34BF040             DC   C'1.3.0 '                           VERSION/RELEASE/MOD.
000017  F1F061F0F361F9F2         DC   C'10/03/92 '                        DATE COMPILED
000020  F1F14BF2F04BF2F0         DC   C'11.20.20'                         TIME COMPILED
000028  00000054                 DC   A(*+44)            @APARM:          A(@PARMS)
00002C  69683C5C0000             DC   X'69683C5C0000'                     INFO. BYTES 1-6
000032  000040000100             DC   X'000040000100'                     INFO. BYTES 7-12
000038  000000000800             DC   X'000000000800'                     INFO. BYTES 13-18
00003E  0000000000               DC   X'0000000000'                       INFO. BYTES 19-23
000043  00                       DC   X'00'                               RESERVED
000044  00000008                 DC   X'00000008'                         # DATA DIV. STMTS.
000048  00000005                 DC   X'00000005'                         # PROC. DIV. STMTS.
00004C  0000                     DC   X'0000'                             INFO. BYTES 24-25
00004E  0000                     DC   2X'00'                              RESERVED
000050  40404040                 DC   C'    '                             USER LEVEL INFO (LVLINFO)
000054  00000000                 DC   A(XYZSUB)          @PARMS: 1) MAIN ENTRY POINT ADDRESS
000058  00000080                 DC   A(PGT)                     2) PGT ADDRESS
>> 00005C  000019A8              DC   A(TGT)                     3) TGT ADDRESS
000060  00000005                 DC   A(*-91)                    4) A(@EPNAM)= E. P. NAME ADDRESS
000064  00000000                 DC   A(XYZSUB)                  5) CURRENT ENTRY POINT ADDRESS
000068  00000152                 DC   A(START)           @BRVAL: PROCEDURE CODE ADDRESS
00006C  00000000                 DC   V(IGZEBST)                 INITIALIZATION ROUTINE
000070  90EC D00C                STM  14,12,12(13)       @STM:   SAVE CALLER'S REGISTERS
000074  5810 F028                L    1,40(,15)                  GET ADDR. OF PARMLIST FROM @APARM
000078  98EF F068                LM   14,15,104(15)              LOAD ADDRESSES FROM @BRVAL
00007C  07FF                     BR   15                         DO ANY NECESSARY INITIALIZATION
```

The Program Global Table (PGT, also known as INIT 2); A listing of all the constants defined or used in the program as well as the complete listing of the compiled program code (INIT3, containing assembly language instructions, related machine code instructions, and identifiers for operands when available); A table of addresses for file manipulation, and the Task Global Table (TGT). These sections comprise the complete Object Code file for the program code. Each of the relevant sections will be discussed.

Use of these sections (all of which are identified, line by line, with the <u>program relative address</u> of the instruction) along with the ABEND address given by the last seven digits of the PSW (Program Status Word), and ONE absolute address, will provide the address of the ABEND instruction and the internal organization of the complete program.

CALCULATING AN INSTRUCTION ABSOLUTE ADDRESS

An absolute address of any item can be calculated from a <u>program relative address</u> for the item in question, and the <u>absolute address</u> of the beginning of the program containing that item. Only the DUMP parameter may provide the absolute address of the beginning of the program, but it is possible to calculate this address from the

absolute address of the TGT (Task Global Table), (which is almost always provided in the ABEND page information) and the Program relative address of the TGT from the LIST, as shown above in INIT 1.

The program relative address, or relative address, of the TGT is also given in the TGT portion of the program listing, as shown.

```
                *** TGT MEMORY MAP *** XYZSUB
          PGMLOC  TGTLOC                                          Program Relative Address of TGT
>>        0019A8  000000  72 BYTE SAVE AREA
          0019F0  000048  TGT IDENTIFIER
          0019F8  000050  TGT LEVEL INDICATOR
          0019F9  000051  RESERVED - 3 SINGLE BYTE FIELDS
          0019FC  000054  32 BIT SWITCH
          001A00  000058  POINTER TO RUNCOM
          001A04  00005C  POINTER TO COBVEC
          001A08  000060  POINTER TO PROGRAM DYNAMIC BLOCK TABLE
          001A0C  000064  NUMBER OF FCB'S
          001A10  000068  WORKING STORAGE LENGTH
          001A14  00006C  POINTER TO PREVIOUS TGT IN TGT CHAIN
          001A18  000070  ADDRESS OF IGZESMG WORK AREA
          001A1C  000074  ADDRESS OF 1ST GETMAIN BLOCK (SPACE MGR)
          001A20  000078  FULLWORD RETURN CODE
          001A22  00007A  RETURN CODE SPECIAL REGISTER
          001A24  00007C  SORT-RETURN SPECIAL REGISTER
          001A26  00007E  MERGE FILE NUMBER
          001A28  000080  RESERVED - 4 HALF WORD FIELDS
          001A30  000088  PROGRAM MASK OF CALLER OF THIS PROGRAM
          001A31  000089  PROGRAM MASK USED BY THIS PROGRAM
          001A32  00008A  RESERVED - 2 SINGLE BYTE FIELDS
          001A34  00008C  NUMBER OF SECONDARY FCB CELLS
          001A38  000090  LENGTH OF THE VN(VNI) VECTOR
          001A3C  000094  ADDRESS OF IGZEBST TERMINATION ROUTINE
          001A40  000098  DDNAME FOR DISPLAY OUTPUT
          001A48  0000A0  SORT-CONTROL SPECIAL REGISTER
          001A50  0000A8  POINTER TO COM-REG SPECIAL REGISTER
          001A54  0000AC  CALC ROUTINE REGISTER SAVE AREA
          001A88  0000E0  ALTERNATE COLLATING SEQUENCE TABLE PTR.
          001A8C  0000E4  ADDRESS OF SORT G.N. ADDRESS BLOCK
          001A90  0000E8  ADDRESS OF IGZCLNK DYNAMIC WORK AREA
          001A94  0000EC  CURRENT INTERNAL PROGRAM NUMBER
          001A98  0000F0  POINTER TO 1ST IPCB
          001A9C  0000F4  RESERVED
          001AA0  0000F8  POINTER TO ABEND INFORMATION TABLE
          001AA4  0000FC  POINTER TO FDMP/TEST FIELDS IN THE TGT
          001AA8  000100  ADDRESS OF START OF COBOL PROGRAM
          001AAC  000104  POINTER TO VN'S IN CGT
          001AB0  000108  POINTER TO VN'S IN TGT
          001AB4  00010C  POINTER TO FIRST PBL IN THE PGT
          001AB8  000110  POINTER TO FIRST FCB CELL
          001ABC  000114  WORKING STORAGE ADDRESS
          001AC0  000118  POINTER TO FIRST SECONDARY FCB CELL

                *** VARIABLE PORTION OF TGT ***

          001AC4  00011C  BACKSTORE CELL FOR SYMBOLIC REGISTERS
          001ACC  000124  BASE LOCATORS FOR SPECIAL REGISTERS
          001AD4  00012C  BASE LOCATORS FOR WORKING-STORAGE
          001AD8  000130  BASE LOCATORS FOR LINKAGE-SECTION
          001AE4  00013C  FDUMP AND TEST INFORMATION AREA
          001B14  00016C  VARIABLE NAME (VN) CELLS
          001B18  000170  PERFORM SAVE CELLS
          001B1C  000174  INTERNAL PROGRAM CONTROL BLOCKS
          001B30  000188  TEMPORARY STORAGE-2
                                                                  Pgm Relative Address of TGT
TGT      LOCATED AT 0019A8 FOR 000D0198 BYTES
WRK-STOR LOCATED AT 001B40 FOR 00000015 BYTES
SPEC-REG LOCATED AT 001B58 FOR 00000031 BYTES
```

The TGT (Task Global Table, sometimes known as INIT 4) also provides very important DUMP information and addresses, which will be discussed later in this chapter. for now, it provides one of the few methods of calculating the absolute address of the beginning of the program. Using the **program relative address of the TGT**, found in several places in the COBOL II object code LIST (highlighted above). If you subtract the **program relative address of the TGT** from the **absolute address of the TGT** (found in the VS COBOL II ABEND Information shown below), the result will be the **absolute address of the beginning of the program**.

```
--- VS COBOL II ABEND Information ---
Program = 'XYZSUB' compiled on '10/03/92' at '11:20:20'
>>    TGT = '001170F8'                                    Absolute Address for XYZSUB TGT
No files were used in this program.
Contents of base locators for working storage are:
    0-00117290
Contents of base locators for the linkage section are:
    0-00000000        1-00119AF0        2-00119AF2
No variably located areas were used in this program.
No EXTERNAL data was used in this program.
No indexes were used in this program.

Program = 'EXAMPLE' compiled on '10/03/92' at '11:20:32'
>>    TGT = '00119610'
Contents of base locators for files are:                  Absolute Address for EXAMPLE TGT
    0-001143C0        1-00114580
Contents of base locators for working storage are:
    0-00119AE0
Contents of base locators for the linkage section are:
    0-00000000
No variably located areas were used in this program.
No EXTERNAL data was used in this program.
No indexes were used in this program.
--- End of VS COBOL II ABEND Information ---
```

Using the information provided in the TGTs for EXAMPLE (listed below) and XYZSUB (listed above), the calculations for absolute addresses is as follows:

Abs(TGT(XYZSUB)) - Rel(TGT(XYZSUB)) = Abs(PGM(XYZSUB))
1170F8 - 19A8 = 115750

Abs(TGT(EXAMPLE)) - Rel(TGT(EXAMPLE)) = Abs(PGM(EXAMPLE))
119610 - 2330 = 1172E0

These absolute addresses can also be calculated from one of the Absolute values of the TGT, and the information given on the High-Level Link-Editor Page. When a program ABENDS in the MAIN, instead of in the SUBprogram, it is not possible for the system to provide information related to the SUB. Therefore, you must calculate these values using the method shown later.

Once the Absolute address of the beginning of the relevant program (in this case, XYZSUB) has been obtained, it is then possible to calculate the **program relative address of the instruction** that is related to the ABEND (Whether this is the actual instruction causing the ABEND, or the instruction that follows the instruction causing the ABEND is left for later discussion.) in XYZSUB as follows:

Abs(Instr(XYZSUB)) - Abs(PGM(XYZSUB)) = Rel(Instr(XYZSUB))
115968 - 115750 = 218

This **program relative address** can then be located in the Object Code provided by the LIST parameter. The section under consideration is part of the compiler listing known as INIT 3, and begins with the translation of the PROCEDURE DIVISION code, indicated by the START * Assembly code statement.

LOCATING THE COBOL SOURCE CODE LINE NUMBER

The next step in locating the error is to find the program relative address of the machine code instruction previously calculated. The object code list, shown below,

is structured to include the COBOL source code line number, the COBOL command on that line, and the program relative address in addition to the machine code and Assembly code necessary to perform the operation. It is then a matter of locating the proper relative address and looking for the related COBOL line.

```
COBOL
Line#
   Pgm Rel
   Address Machine Code    Assembly Code                    Compiler Generated Comments
   000152                  START  EQU  *                    XYZSUB
   000152  58A0 C000               L    10,0(0,12)          CBL=1
   000156  5890 D12C               L    9,300(0,13)         BLW=0
   00015A  5880 D134               L    8,308(0,13)         BLL=1
   00015E  5870 D138               L    7,312(0,13)         BLL=2
   000162  5820 1000               L    2,0(0,1)
   000166  4120 2000               LA   2,0(0,2)
   00016A  5020 D134               ST   2,308(0,13)         BLL=1
   00016E  1882                    LR   8,2
   000170  5820 1004               L    2,4(0,1)
   000174  4120 2000               LA   2,0(0,2)
   000178  5020 D138               ST   2,312(0,13)         BLL=2
   00017C  1872                    LR   7,2
   00017E  D203 D0EC A004          MVC  236(4,13),4(10)     TGTFIXD+236                PGMLIT AT +0
   000184  9180 D174               TM   372(13),X'80'       IPCB=1
   000188  58B0 C004               L    11,4(0,12)          PBL=1
   00018C  4780 B098               BC   8,152(0,11)         GN=6(0001A4)
   000190  0520                    BALR 2,0
   000192  5020 D13C               ST   2,316(0,13)         TGT FDMP/TEST-INFO. AREA +0
   000196  5820 D05C               L    2,92(0,13)          TGTFIXD+92
   00019A  58F0 21CC               L    15,460(0,2)         V(IGZEMSG )
   00019E  4110 A065               LA   1,101(0,10)         PGMLIT AT +97
   0001A2  05EF                    BALR 14,15
   0001A4                  GN=6    EQU  *
   0001A4  9680 D174               OI   372(13),X'80'       IPCB=1
   0001A8  9140 D174               TM   372(13),X'40'       IPCB=1
   0001AC  4710 B0BC               BC   1,188(0,11)         GN=7(0001C8)
   0001B0  D203 D170 D16C          MVC  368(4,13),364(13)   PSV=1                      VN=1
   0001B6  4120 B0B6               LA   2,182(0,11)         GN=8(0001C2)
   0001BA  5020 D16C               ST   2,364(0,13)         VN=1
   0001BE  47F0 B000               BC   15,0(0,11)          GN=2(0D010C)
   0001C2                  GN=8    EQU  *                                              PSV=1
   0001C2  D203 D16C D170          MVC  364(4,13),368(13)   VN=1
   0001C8                  GN=7    EQU  *
   0001C8  9640 D174               OI   372(13),X'40'       IPCB=1
00032  *000-MAIN-LOOP
00033  ADD                                                                              X
   0001CC  D201 D18E 8000          MVC  398(2,13),0(8)      TS2=6
   0001D2  960F D18F               OI   399(13),X'0F'       TS2=7                      Y
   0001D6  D202 D195 7000          MVC  405(3,13),0(7)      TS2=13
   0001DC  960F D197               OI   407(13),X'0F'       TS2=15
   0001E0  FA21 D195 D18E          AP   405(3,13),398(2,13) TS2=13                     TS2=6
   0001E6  D202 7000 D195          MVC  0(3,7),405(13)      Y                          TS2=13
   0001EC  940F 7000               NI   0(7),X'0F'          Y
   0001F0  960F 7002               OI   2(7),X'0F'          Y+2
00034  ADD
   0001F4  FA20 900B A031          AP   11(3,9),49(1,10)    CRASH2                     PGMLIT AT +45
   0001FA  940F 900B               NI   11(9),X'0F'         CRASH2                     CRASH2
   0001FE  F822 900B 900B          ZAP  11(3,9),11(3,9)     CRASH2
00035  IF                                                                               PGMLIT AT +43
   000204  F921 900B A02F          CP   11(3,9),47(2,10)    CRASH2
   00020A  58B0 C004               L    11,4(0,12)          PBL=1
   00020E  47D0 B116               BC   13,278(0,11)        GN=3(000222)
00035  ADD                                                                              CRASH1
>> 000212  FA20 900B 900A          AP   11(3,9),10(1,9)     CRASH2
>> 000218  940F 900B               NI   11(9),X'0F'         CRASH2                     CRASH2
   00021C  F822 900B 900B          ZAP  11(3,9),11(3,9)     CRASH2
   000222                  GN=3    EQU  *
00036  *010-MAIN-LOOP-EXIT
00037  GOBACK
   000222  47F0 B136               BC   15,310(0,11)        GN=1(000242)
   000226  9120 D054               TM   84(13),X'20'        TGTFIXD+84
   00022A  47E0 B136               BC   14,310(0,11)        GN=1(000242)
   00022E  0520                    BALR 2,0
   000230  5020 D13C               ST   2,316(0,13)         TGT FDMP/TEST-INFO. AREA +0
   000234  5820 D05C               L    2,92(0,13)          TGTFIXD+92
   000238  58F0 21CC               L    15,460(0,2)         V(IGZEMSG )
   00023C  4110 A053               LA   1,83(0,10)          PGMLIT AT +79
   000240  05EF                    BALR 14,15
   000242                  GN=1    EQU  *
   000242  947F D174               NI   372(13),X'7F'       IPCB=1
   000246  0520                    BALR 2,0
   000248  5020 D13C               ST   2,316(0,13)         TGT FDMP/TEST-INFO. AREA +0
   00024C  5820 D05C               L    2,92(0,13)          TGTFIXD+92
   000250  58F0 2224               L    15,548(0,2)         V(IGZETRM )
   000254  4110 A052               LA   1,82(0,10)          PGMLIT AT +78
   000258  05EF                    BALR 14,15

   PROGSUM TABLE LOCATED AT D00264 FOR 00006C BYTES
   PROCTAB TABLE LOCATED AT 0002D8 FOR 00001E BYTES
ATTRIBUTE TABLE LOCATED AT 0002F8 FOR 0004CC BYTES
   SYMBOL TABLE LOCATED AT 0007C4 FOR 0001E0 BYTES
     HASH TABLE LOCATED AT 0009A4 FOR 001000 BYTES
```

Use of this technique indicates that COBOL Source Code line number 35 in XYZSUB caused the ABEND. Specifically, the AP instruction at program relative address 212 resulted in the ABEND processing. Determination of the actual instruction is left for later discussion. This result agrees with the line number provided in the FDUMP.

Once a valid ABEND line number is found, the first step is to examine, not only this line, but the one immediately prior to it in the program logic. This will allow identification of pertinant variables. Before continuing to investigate the relevant variables, check to see if a SECTION header is missing, or whether the line is part of an infinite loop.

Incorrectly called subroutines can cause a number of data errors, because of the use of variable "passing by reference" rather than "passing by value". This means that the absolute address of the original variable is communicated to the subroutine, without incidations of structure or length, rather than the contents. Since the COBOL subprogram redefines these variables in the LINKAGE SECTION, it is possible to introduce major data errors in the original variables.

If the cause is not related to the immediate logic, or found in the Subroutine Linkage, investigation moves to the variables found in the relevant statements. For each variable, determine its definition, and whether or not it was properly initialized. If this does not indicate the difficulty, it is time to invoke additional compiler tools.

USING MAP for Data descriptions

The COBOL 2 compiler parameters, MAP and XREF, provide a listing of each variable found in the program code, its related structure, length, relative location in memory, and the lines in which each is used. For this example, XYZSUB contains:

```
DATA DIVISION MAP
DATA DEFINITION ATTRIBUTE CODES (RIGHTMOST COLUMN) HAVE THE FOLLOWING MEANINGS:
    D = OBJECT OF OCCURS DEPENDING     G = GLOBAL                          S = SPANNED FILE
    E = EXTERNAL                       O = HAS OCCURS CLAUSE               U = UNDEFINED FORMAT FILE
    F = FIXED LENGTH FILE              OG= GROUP HAS OWN LENGTH DEFINITION V = VARIABLE LENGTH FILE
   FB= FIXED LENGTH BLOCKED FILE       R = REDEFINES                       VB= VARIABLE LENGTH BLOCKED FILE
SOURCE   HIERARCHY AND                                   BASE        HEX-DISPLACEMENT  ASMBLR DATA                    DATA DEF
LINEID   DATA NAME                                       LOCATOR     BLK   STRUCTURE   DEFINITION    DATA TYPE        ATTRIBUTES
     8  PROGRAM-ID XYZSUB ------------------------------------------------------------------------------------------------*
    19   01 SYSTEM-FLAGS. . . . . . . . . . . . . . . . BLW=0000   000                 DS 0CL21      GROUP
    20     02 SWS-START . . . . . . . . . . . . . . . . BLW=0000   000   0 000 000    DS 9C         DISPLAY
    21     02 DUMMY . . . . . . . . . . . . . . . . . . BLW=0000   009   D 000 009    DS 1C         DISP-NUM
    22     02 CRASH1. . . . . . . . . . . . . . . . . . BLW=0000   00A   0 000 00A    DS 1P         PACKED-DEC
    23     02 CRASH2. . . . . . . . . . . . . . . . . . BLW=0000   00B   0 000 00B    DS 3P         PACKED-DEC
    24     02 SWS-END . . . . . . . . . . . . . . . . . BLW=0000   00E   0 000 00E    DS 7C         DISPLAY
    27   01 X . . . . . . . . . . . . . . . . . . . . . BLL=0001   000                 DS 2P         PACKED-DEC
    28   01 Y . . . . . . . . . . . . . . . . . . . . . BLL=0002   000                 DS 3P         PACKED-DEC
```

All variables, whether defined by the user, or included through a Report Writer precompiler, are described in this table. The Base Locators, for Working Storage (BLW), for Files (BLF), and for the Linkage Section variables (BLL), indicate which absolute address is used in the calculation of a variable's location. The Hex Displacement, added to the value of the Base locator, result in the absolute address of the variable. It is then possible to find the variable in the dump and determine

the exact contents. A second method of finding variable's contents is to examine the FDUMP listing. However, the hexidecimal contents of all variables at the time of ABEND can always be found in the Dump.

Calculating Working Storage Data Addresses

In order to calculate the absolute address of items in Working Storage, it is necessary to determine the value for each Base Locator given in the MAP table. Working from the COBOL II ABEND Information, the following addresses are obtained.

```
--- VS COBOL II ABEND Information ---                          PROGRAM-ID of addresses
Program = 'XYZSUB' compiled on '10/03/92' at '11:20:20'
    TGT = '001170F8'
No files were used in this program.
Contents of base locators for working storage are:             Base Locator of Working Storage
>>  0-00117290
Contents of base locators for the linkage section are:         Base Locators for Linkage Variables
>>  0-00000000        1-00119AF0        2-00119AF2
No variably located areas were used in this program.
No EXTERNAL data was used in this program.
No indexes were used in this program.

Program = 'EXAMPLE' compiled on '10/03/92' at '11:20:32'
    TGT = '00119610'
Contents of base locators for files are:
    0-001143C0        1-00114580
Contents of base locators for working storage are:
    0-00119AE0
Contents of base locators for the linkage section are:
    0-00000000
No variably located areas were used in this program.
No EXTERNAL data was used in this program.
No indexes were used in this program.
--- End of VS COBOL II ABEND Information ---
```

For XYZSUB, the value (absolute address) given by the Base Locator for Working Storage (BLW) is 117290. Therefore, the first variable in the Working Storage Section is found at the BLW + Hex Displacement (117290 + 0), or 117290. The remaining variable's addresses are calculated in the same manner, remembering that all displaements and lengths given in the MAP are in base 16.

Variable X and Y are defined in the Linkage Section, and have their absolute addresses calculated in like manner. For X, the value of the BLL cell is 119AF0, and the displacement is 0. Y is found at 119AF2. These addresses are located within the Main program, EXAMPLE, as one should expect.

Addresses of variables in EXAMPLE can be determined using the MAP information from EXAMPLE and the values given in the ABEND information. Mapping the Data Division found in the DUMP information can save valuable time in determining the cause of a data ABEND error.

USING XREF

The XREF parameter provides the investigator with a complete listing of all occurrences of each variable and all PERFORM, GOTO or CALL statements. Definition locations of each variable used in a COBOL command are provided in

the compiler listing. In addition, several tables of information follow the listing. These tables enumerate COBOL Verbs, Sections and Paragraphs, and Variable locations and usages.

```
COUNT   CROSS-REFERENCE OF VERBS          REFERENCES

    3   ADD. . . . . . . . . . . . . . . 33 34 35
    1   GOBACK . . . . . . . . . . . . . 37
    1   IF . . . . . . . . . . . . . . . 35
```

The data cross reference record indicates in which lines the value of a variable could be modified. It is then necessary to investigate each line of the logic which possibly changed the variable in question, both at the elementary item level and the group item level. For XYZSUB, this table is as follows.

```
AN "M" PRECEDING A DATA-NAME REFERENCE INDICATES THAT THE DATA-NAME IS MODIFIED BY THIS REFERENCE.

DEFINED   CROSS-REFERENCE OF DATA NAMES   REFERENCES

   22     CRASH1 . . . . . . . . . . . . 35
   23     CRASH2 . . . . . . . . . . . . M34 35 M35
   21     DUMMY
   24     SWS-END
   20     SWS-START
   19     SYSTEM-FLAGS
   27     X. . . . . . . . . . . . . . . 31 33
   28     Y. . . . . . . . . . . . . . . 31 M33
```

Paragraphs, Sections, and external references are included in a Procedure Reference listing. This table indicates the non-sequential access of named code sections, and the type of command used to transfer logic control. In the case of EXAMPLE, there were no PERFORM or GOTO statements. Therefore, the References are blank, indicating sequential logic.

```
CONTEXT USAGE IS INDICATED BY THE LETTER PRECEDING A PROCEDURE-NAME REFERENCE.
THESE LETTERS AND THEIR MEANINGS ARE:
    A = ALTER (PROCEDURE-NAME)
    D = GO TO (PROCEDURE-NAME) DEPENDING ON
    E = END OF RANGE OF (PERFORM) THROUGH (PROCEDURE-NAME)
    G = GO TO (PROCEDURE-NAME)
    P = PERFORM (PROCEDURE-NAME)
    T = (ALTER) TO PROCEED TO (PROCEDURE-NAME)
    U = USE FOR DEBUGGING (PROCEDURE-NAME)

DEFINED   CROSS-REFERENCE OF PROCEDURES   REFERENCES

   32     000-MAIN-LOOP
   36     010-MAIN-LOOP-EXIT

DEFINED   CROSS-REFERENCE OF PROGRAMS     REFERENCES

    8     XYZSUB
```

XREF listings of the paragraphs and sections which comprise the program logic include indications of any GO TO statements and PERFORMs. GO TO statements are frequently the source of uncontrolled logic and unexpected data errors.

If the error is not immediately obvious from these tools, the TRACE, DISPLAY, and USE FOR DEBUGGING statements can be employed to trace program logic at various levels of detail. For a discussion of these commands, please refer to any COBOL textbook.

IBM 370 COBOL DUMPS

In order to make full use of the information provided in an IBM 370 Dump, one must locate all the program structure data detailed in the compile step. Combining these with the absolute addresses of the **Beginning of the program** (program relative address 0000), the **Task Global Table** (TGT), and the **beginning of Working Storage** for each program and subprogram gives you a basis for locating and evaluating any portion of the program at the time of ABEND.

In addition to the information discussed in preceding sections, the following data is provided.

LIST INFORMATION

The Program Global Table (PGT) follows INIT 1, and can be known as INIT 2. This provides the access connection to any input/output files used in the program, and subsequently defined in the CGT (Constant Global Table)

```
PROGRAM GLOBAL TABLE BEGINS AT LOCATION 000080 FOR 000008 BYTES
THE PGT CONTAINS 000001 CELL(S) FOR ADDRESSABILITY TO THE CGT
PBL1    AT LOCATION 00010C FOR LINE 8
```

The Constant Global Table (CGT) begins the section referred to as INIT 3. It contains the definition, in the proper format (DISPLAY, COMP, COMP-1, COMP-2, COMP-3 for numerics) and length, for all literals used anywhere in the program. This includes VALUE statements as well as values found in MOVE, COMPUTE, or related statements. These addresses are used in the object code instead of attempting to code the relevant HEX contents.

For EXAMPLE, this includes the following.

```
CONSTANT GLOBAL TABLE BEGINS AT LOCATION 000088 FOR 000081 BYTES
LITERAL POOL MAP FOR LITERALS IN THE CGT:

0008C  (LIT+0)   00000001 00000000 E2E6E260 E2E3C1D9 E3E7E8E9 E2E4C240 40E2E8E2 D6E4E340 |........SWS-STARTXYZSUB  SYSOUT |
000AC  (LIT+32)  40E2E6E2 60C5D5C4 00035C03 8F1F0000 00000000 012C0000 00010000 01300000 | SWS-END..*.....................|
000CC  (LIT+64)  00030000 00000000 00008000 00004080 00000000 25C00001 40000808 00000000 |...............  ........  .......|
000EC  (LIT+96)  15800000 000040C0 00014000 06080000 00001502 40000808 00000000 15       |......  ...  ........  ........|
                       GN=2      EQU   *
    00010C                       MVC   0(9,9),12(10)          SWS-START                    PGMLIT AT +8
    00010C  D208 9000 A00C       MVI   9(9),X'F0'             DUMMY
    000112  92F0 9009            MVC   11(3,9),44(10)         CRASH2                       PGMLIT AT +40
    000116  D202 900B A02C       MVC   14(7,9),37(10)         SWS-END                      PGMLIT AT +33
    00011C  D206 900E A025       L     2,296(0,13)            BL=1
    000122  5820 D128            MVC   0(4,2),8(10)           SORT-CORE-SIZE               PGMLIT AT +4
    000126  D203 2000 A008       MVC   8(4,2),8(10)           SORT-FILE-SIZE               PGMLIT AT +4
    00012C  D203 2008 A008       MVC   16(4,2),8(10)          SORT-MODE-SIZE               PGMLIT AT +4
    000132  D203 2010 A008       MVC   24(8,2),29(10)         SORT-MESSAGE                 PGMLIT AT +25
    000138  D207 2018 A01D       MVC   32(4,2),8(10)          TALLY                        PGMLIT AT +4
    00013E  D203 2020 A008       MVI   40(2),X'0E'            SHIFT-OUT
    000144  920E 2028            MVI   48(2),X'0F'            SHIFT-IN
    000148  920F 2030            L     3,364(0,13)            VN=1
    00014C  5830 D16C            BCR   15,3
    000150  07F3
```

The remainder of INIT 3 provides the machine code and assembly command listing of all COBOL commands used in the program. As shown previously, it is possible

to determine a COBOL instruction line number from a program relative location, and thereby track down a specific ABEND.

TGT INFORMATION

A Task Global Table (TGT) provides the operating system with all of the relevant addresses needed for program execution. At compile time, these addresses are stored in program relative format, and the first requirement of loading this program into memory for execution is to resolve (assign an absolute address to) these values. The TGT structure is divided into a FIXED portion, which structure and length do not change from program to program, and a VARIABLE portion.

Examination of the TGT Memory Map for EXAMPLE provides the following :

```
          *** TGT MEMORY MAP ***  EXAMPLE
 PGMLOC  TGTLOC
 002330  000000  72 BYTE SAVE AREA
```

The 72 byte (18 word) Save Area is used to store the contents of the General Purpose Registers during a Subprogram CALL. These values record the current operating state of this program, and are restored to the General Purpose Registers when control is returned to this program from the Called program.

```
 002378  000048  TGT IDENTIFIER
 002380  000050  TGT LEVEL INDICATOR
 002381  000051  RESERVED - 3 SINGLE BYTE FIELDS
 002384  000054  32 BIT SWITCH
 002388  000058  POINTER TO RUNCOM
 00238C  00005C  POINTER TO COBVEC
 002390  000060  POINTER TO PROGRAM DYNAMIC BLOCK TABLE
 002394  000064  NUMBER OF FCB'S
```

FCBs (File Control Blocks) exist when I/O occurs. The value stored in this location indicates how many files are defined for this program, whether or not they are opened.

```
 002398  000068  WORKING STORAGE LENGTH
```

This value is the number of bytes used for Working Storage.

```
 00239C  00006C  POINTER TO PREVIOUS TGT IN TGT CHAIN
```

When CALLED from another program, this field will contain the absolute address of the TGT in the CALLING program in order to reestablish proper control at the GOBACK or EXIT PROGRAM statement.

```
 0023A0  000070  ADDRESS OF IGZESMG WORK AREA
 0023A4  000074  ADDRESS OF 1ST GETMAIN BLOCK (SPACE MGR)
 0023A8  000078  FULLWORD RETURN CODE
 0023AA  00007A  RETURN CODE SPECIAL REGISTER
 0023AC  00007C  SORT-RETURN SPECIAL REGISTER
 0023AE  00007E  MERGE FILE NUMBER
 0023B0  000080  RESERVED - 4 HALF WORD FIELDS
 0023B8  000088  PROGRAM MASK OF CALLER OF THIS PROGRAM
 0023B9  000089  PROGRAM MASK USED BY THIS PROGRAM
 0023BA  00008A  RESERVED - 2 SINGLE BYTE FIELDS
 0023BC  0008BC  NUMBER OF SECONDARY FCB CELLS
 0023C0  000090  LENGTH OF THE VN(VNI) VECTOR
 0023C4  000094  ADDRESS OF IGZEBST TERMINATION ROUTINE
 0023C8  000098  DDNAME FOR DISPLAY OUTPUT
 0023D0  0000A0  SORT-CONTROL SPECIAL REGISTER
 0023D8  0000A8  POINTER TO COM-REG SPECIAL REGISTER
 0023DC  0000AC  CALC ROUTINE REGISTER SAVE AREA
 002410  0000E0  ALTERNATE COLLATING SEQUENCE TABLE PTR.
 002414  0000E4  ADDRESS OF SORT G.N. ADDRESS BLOCK
 002418  0000E8  ADDRESS OF IGZCLNK DYNAMIC WORK AREA
 00241C  0000EC  CURRENT INTERNAL PROGRAM NUMBER
 002420  0000F0  POINTER TO 1ST IPCB
 002424  0000F4  RESERVED
 002428  0000F8  POINTER TO ABEND INFORMATION TABLE
 00242C  0000FC  POINTER TO FDMP/TEST FIELDS IN THE TGT
 002430  000100  ADDRESS OF START OF COBOL PROGRAM
```

This field contains the absolute address of INIT 1 for this program. If the pointer address used by the system is somehow modified during execution, it is compared against this address, and corrected, if possible.

```
 002434  000104  POINTER TO VN'S IN CGT
```

```
002438  000108  POINTER TO VN'S IN TGT
00243C  00010C  POINTER TO FIRST PBL IN THE PGT
002440  000110  POINTER TO FIRST FCB CELL
002444  000114  WORKING STORAGE ADDRESS
```

The Absolute Address of the beginning of Working Storage is found here. This is not used in the calculation of addresses, but serves as a checkpoint, much like the Address of Start of COBOL Program.

```
002448  000118  POINTER TO FIRST SECONDARY FCB CELL

        *** VARIABLE PORTION OF TGT ***
00244C  00011C  BACKSTORE CELL FOR SYMBOLIC REGISTERS
00245C  00012C  BASE LOCATORS FOR SPECIAL REGISTERS
002464  000134  BASE LOCATORS FOR WORKING-STORAGE
```

This variable length field contains all the absolute addresses necessary to calculate addresses of variables defined in Working Storage. It is possible to have more than one value in this list, especially when using Report Writer. The system uses these values, in conjunction with the MAP Hex Displacement information, to calculate the absolute address of any variable defined.

```
002468  000138  BASE LOCATORS FOR LINKAGE-SECTION
```

Base Locators for the Linkage Section are used in much the same manner as Base Locators for Working Storage. Variables defined in the Linkage Section are assigned to a Base Linkage Locator, and subordinate variables' locations are calculated by Hex Displacement from this absolute value. **These values can change in the case of multiple CALLs to the same procedure using different variables, as well as multiple ENTRY points.** The first Base Locator in this list (BLL=0) is used to map variations from the calling variable lists onto the existing structure.

```
00246C  00013C  BASE LOCATORS FOR FILES
```

Base Locators for Files are provided as linkages between the buffers and the data being read or written.

```
002474  000144  CLLE ADDR. CELLS FOR CALL LIT. SUB-PGMS.
002478  000148  FDUMP AND TEST INFORMATION AREA
0024A8  000178  VARIABLE NAME (VN) CELLS
0024B8  000188  PERFORM SAVE CELLS
0024C8  000198  FCB CELLS
0024D0  0001A0  INTERNAL PROGRAM CONTROL BLOCKS

0024E0  0001B0  TEMPORARY STORAGE-2
```

Temp Storage 2 contains the absolute addresses of the variables used in a CALL statement. These address are communicated to the Base Linkage Locators in the CALLED program.

Following the TGT is a secondary listing of the most important program relative addresses and their HEXIDECIMAL lengths. This is a backup location for deriving the absolute addresses of the TGT, Working Storage, and File Control Blocks for use in debugging.

```
TGT       LOCATED AT 002330 FOR 000001C0 BYTES
DCB00001  LOCATED AT 0024F0 FOR 00000060 BYTES
FCB00001  LOCATED AT 002550 FOR 00000100 BYTES
DCB00002  LOCATED AT 002650 FOR 00000060 BYTES
FCB00002  LOCATED AT 0026B0 FOR 00000100 BYTES
GDT00001  LOCATED AT 0027B0 FOR 00000050 BYTES
WRK-STOR  LOCATED AT 002800 FOR 000000BE BYTES
SPEC-REG  LOCATED AT 0028C0 FOR 00000031 BYTES
```

Until the program is linked and loaded into memory for execution, all addresses are stored in PROGRAM RELATIVE format. Once the program is executing, these addresses are ABSOLUTE, and can be used to locate relevant values in the core memory dump.

LKED INFORMATION

The Linkage Editor Information page provides data regarding the structure of the Load Module. This includes a list of all Program-IDs and Entry points found in the Load Module, their length (in hexidecimal), and their relative locations within the complete program. In addition, if a CALL statement is issued, and the proper Program-ID or ENTRY is not found in the object code provided to the Linkage step, an $UNRESOLVED$ message appears.

```
H96-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,LET,MAP
          DEFAULT OPTION(S) USED -  SIZE=(186368,55296)
IEW0000      ENTRY EXAMPLE
```

The ENTRY point of the resulting executable module is provided by the programmer. If no entry point is defined in the LKED.SYSIN information, the entry point is assumed to be the first module. In this case, program execution would begin in XYZSUB.

```
                                        CROSS REFERENCE TABLE

CONTROL SECTION                    ENTRY
  NAME    ORIGIN  LENGTH             NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
XYZSUB       00    1B89
EXAMPLE    1B90    28F1
IGZEBST *  4488     428
                                   IGZEBS2     4716

LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
    6C            IGZEBST          IGZEBST                 1BFC            IGZEBST          IGZEBST
  1C24            XYZSUB           XYZSUB                  4884            IGZETUN          $UNRESOLVED(W)
  4888            IGZEOPT          $UNRESOLVED(W)
ENTRY ADDRESS     1B90
TOTAL LENGTH      4880
```

The Linkage Editor also indicates the relative address of the defined ENTRY point, and the hexidecimal length of the complete executable program. At the conclusion of Link-Edit, the resulting executable code is stored in a Partitioned Data Set. The default name for this entry is GO.

```
****GO        DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET    AMODE ANY
RMODE IS 24
AUTHORIZATION CODE IS          0.
```

As with the COMPILE step, a return code (Authorization Code) of 0 indicates no errors occurred in the Link-Edit step, and the job continues to the next step of the JCL.

FDUMP INFORMATION

Sixteen General Purpose Registers are used to control operation of any program in memory. These registers are absolute addresses of specific information, or contain a storage word of operation flags.

Provided by the FDUMP at ABEND, or found in the JES2 LOG, the contents of specific registers may provide additional information.

```
--- VS COBOL II Formatted Dump at ABEND ---                    PROGRAM-ID executing at ABEND
-Program = 'XYZSUB'
Completion code = 'S0C7'                                       Absolute Address of Machine Code
PSW at ABEND = 'FF850007E0115968'                                          Line Number
Line number or verb number being executed:  '0000035'/'2'
The GP registers at entry to ABEND were
    Regs  0 - 3  - '00119754  001197C0  00119AF2  0011492C'
    Regs  4 - 7  - '001157A4  0011492C  00158700  00119AF2'
    Regs  8 - 11 - '00119AF0  00117290  001157D8  0011585C'
    Regs 12 - 15 - '001157D0  001170F8  001158A2  8013E398'
```

Registers 0 and 1 are used primarily for Input/Output operations. Register 1 also serves as the communication path between a calling program and the called program. When variables exist in the USING clause, Register 1 contains the absolute address of the calling program's Temp Storage 2. At any time after the beginning of the called program, the contents of Register 1 may contain different information.

Register 13 contains the absolute address of the current program's Save Area (in the case of a COBOL program, this is the first entry in the TGT). Register 14 contains the absolute address of the first machine code instruction in the next command. For COBOL programs, this may be the next COBOL command to be executed in the sequence, or the command to execute after a PERFORM is complete, among other possibilities. Register 15 is primarily used as an operating condition code flag set upon return from a Called program, contains the absolute address of the Entry Point when entering a Called program.

Depending on when the ABEND occurred, contents of the General Purpose Registers can provide double checks on the addresses previously calculated.

DUMP INFORMATION

A memory dump (core dump) consists of several major sections. The two of current interest are the formatted system control information, provided first, and the hexidecimal memory dump which follows.

Within the first section are several items of interest. The PSW, or program status word) includes a code, frequently at the binary level, to indicate the current operating conditions at time of ABEND. Specifics regarding these flags will be covered later in the course. In addition to the flags, the PSW contains the absolute address of the machine instruction <u>following</u> the machine instruction causing the ABEND, <u>unless</u> the INTC code (interrupt code) is an SVC code. Most frequently, the machine code instruction provided is that of the next instruction.

In order to calculate the absolute address of the machine code instruction that caused the ABEND, it is necessary to subtract the ILC (Instruction Length Code) from the PSW address. In this case, the correct absolute address is 115968 - 6, or

115962. Subtracting the absolute address of the XYZSUB gives the program relative address 212. As discovered earlier, this is the machine code instruction AP (Add Packed) which encountered an invalid data type in one of the variables.

Most of the initial page of information details the state of the operating system at the time of ABEND, and is not useful to our discussion. There are several locations in the CDE section, the TIOT, and the SAVE AREA TRACE which provide data pertinent to our task.

```
JOB FE16SUB          STEP GO              TIME 112052   DATE 92276    ID = 000    CPUID = FF0101523081        PAGE 0001
COMPLETION CODE      SYSTEM = 0C7
PSW AT ENTRY TO ABEND  078D2000 00115968          ILC 6   INTC 0007

ASCB 00FC60B8
      +0    ASCB   C1E2C3C2   FWDP   00F08DA0   BWDP   00FE4140   CMSF   0* ADDRESS SPACE SWITCH EVENT MASK OFF (ASTESSEM = 0) *

TCB  8EF6B0

-ACTIVE RBS
PRB  8EF600  XSB  008EF668  RESV 00000000  RTPSW1 078D2000   00115968  RTPSW2 00060007   0090C000
             FLG1  02000000  WC-L-IC 00060007
             RESV  00000000  APSW    00000000  SZ-STAB 00110082       FL-CDE 00936050  PSW 078D2000   00115968
             Q/TTR 00000000  WT-LNK  008EF6B0
             RG 0-7   0094C010   00114FF8   00000040   0093C654   0093C630   009368F8   008F6FF8   FD000000
             RG 8-15  008EFE48   808EFBB0   00000000   0094C4C8   50E8FD82   00114FB0   40E9011C   008EFE78

XSB AT 008EF668
SVRB 94C880

XSB AT 0094C950
SVRB 94C998

XSB AT 0094CA68
-LOAD LIST
        NE 00923328   RSP-CDE 00F0F6D0   CNT 00010001          NE 008DF340   RSP-CDE 00FDF730   CNT 00010001
        NE 0091F118   RSP-CDE 00FDF4C0   CNT 00020002          NE 00936448   RSP-CDE 00925000   CNT 00010000
        NE 0080F470   RSP-CDE 00920000   CNT 00010000          NE 00000000   RSP-CDE 0091F000   CNT 00010000
```

The CDE lists the Entry Point Addresses (EPAs) of all active program modules. The module identification comes from the entry name in the Partitioned Data Set containing the executable code. In this case, the SYSLMOD Data Set Name is &&GOSET(GO). In the CDE, the entry for GO is the absolute address of the beginning of the program code for the Entry Point designated in the LinkEdit step. Since the Entry given was EXAMPLE, the absolute address given here is that of the beginning of EXAMPLE.

```
-CDE
     936050     NCDE 00000000   RBP 008EF600 NM GO          EPA 001172E0   XL/MJ 0094C000   USE 000100FB   ATTR 0B20000
     FDF6D0     NCDE 00FDF730   RBP 00000000 NM IGG019DK    EPA 00FB92F0   XL/MJ 00FDF6F0   USE 00010000   ATTR B122000
     FDF730     NCDE 00FDF610   RBP 00000000 NM IGG019AQ    EPA 00F85DC0   XL/MJ 00FDF750   USE 00010000   ATTR B122000
     FDF4C0     NCDE 00FEC340   RBP 00000000 NM IGG019DJ    EPA 00BA8930   XL/MJ 00FDF4E0   USE 00050000   ATTR B922000
     925000     NCDE 00920000   RBP 00000000 NM IGZCPC0     EPA 0013CBA8   XL/MJ 0091F178   USE 000100FB   ATTR 3320000
     920000     NCDE 0091F000   RBP 00000000 NM IGZCPAC     EPA 0011C448   XL/MJ 00936788   USE 000100FB   ATTR 3320000
     91F000     NCDE 00936050   RBP 00000000 NM IGZCTC0     EPA 00115538   XL/MJ 0091F0B8   USE 000100FB   ATTR 1320000
-XL                                            LN           ADR        LN          ADR          LN           ADR
```

The Task Input Output Table (TIOT) establishes the connections between the files identified in the SELECT/ASSIGN clauses in the COBOL program and the associated Data Definition (DD) statements in the JCL. It also documents the connections to the buffer control information.

```
TIOT  8F7000      JOB  FE16SUB     STEP   GO          PROC  STEP2
      OFFSET      LN-STA      DDNAME      TTR-STC     STB-UCB
      + 0018      14010100    PGM=*.DD    91BB8000    90002CC8
      + 002C      14010100    STEPLIB     91B34000    80003E08
      + 0040      14010100                91AF4000    80003D18
      + 0054      14010100    SYS02891    916DC000    80003018
      + 0068      14010102    SYSABOUT    91ADC000    80000000
      + 007C      14010102    SYSDBOUT    91AC4000    80000000
      + 0090      14010102    SYSUDUMP    91AAC000    80000000
      + 00A4      14010102    SYSOUT      91A94000    80000000
      + 00B8      14010102    CARDIN      91A7C000    80000000
      + 00CC      14010102    FILE01      91A64000    80000000
-VSM
```

As previously indicated, the 72 byte Save Area is used to store the contents of the General Purpose Registers whenever a program is CALLed. In this case, the executable code GO was entered at Program-Id EXAMPLE, which caused the system to store the contents of the 16

Registers at absolute address 114FB0. The HSA (High-order Save Area) is blank, because the Operating System was not called by any other program. Stored in the LSA (Last Save Area) is the absolute of the Last Save Area accessed by the system. The Return Address (normally Register 14) contains the absolute address of the next instruction to be executed at the conclusion of this program (EXAMPLE). The EPA (Entry Point Address or Register 15) contains the absolute address of the beginning of the program.

```
                          SCB BIT FLAG SUMMARY
                          --------------------

-SAVE AREA TRACE
GO        WAS ENTERED VIA LINK          AT EP EXAMPLE..C2.1.3.0.10.03.92.11.20.32
SA    114FB0  WD1 00000000   HSA 00000000   LSA 00119610   RET 00012750   EPA 001172E0   R0  0094C010
              R1  00114FF8   R2  00000040   R3  0093C654   R4  0093C630   R5  009368F8   R6  008F6FF8
              R7  FD000000   R8  008EFE48   R9  808EFBB0   R10 00000000   R11 0094C4C8   R12 50E8FD82
```

Contained in the Save Area for EXAMPLE (as indicated by the SA address of 119610) are the EPA for XYZSUB (Register 15), the absolute address of Temporary Storage 2 (Register 1), the address of the next machine instruction in EXAMPLE (Register 14, or RET), and the absolute address of the Save Area for XYZSUB (LSA, or Register 13).

```
GO        WAS ENTERED VIA CALL          AT EP XYZSUB...C2.1.3.0.10.03.92.11.20.20
SA    119610  WD1 00108001   HSA 00114FB0   LSA 001170F8   RET 501177FE   EPA 00115750   R0  00119754
              R1  001197C0   R2  501177F4   R3  0011492C   R4  00119830   R5  401177C8   R6  008F6FF8
              R7  001143C0   R8  00114580   R9  00119AE0   R10 00117374   R11 001175B8   R12 00117360
```

Contained in the Save Area for XYZSUB are very few addresses relevant to the debugging process. Register 1 may or may not contain the absolute address of Temp Storage 2 from EXAMPLE, depending on where the ABEND occurred, and whether or not Register 1 was used for other data addresses before the ABEND. In this case, the HSA contains the absolute address of the Save Area of the calling program, EXAMPLE. The remaining registers address variable locations (Register 7 is used to access Y and Register 8 is used to access X, for example) or machine code instructions.

```
IGZCPCO  WAS ENTERED VIA CALL          AT EP .IGZETRM...C23.009.29.88.21.28...O.....
SA    1170F8  WD1 00108001   HSA 00119610   LSA 00000000   RET 501159AA   EPA 801521F8   R0  00119E66
              R1  0011582A   R2  0011492C   R3  00115912   R4  001157A4   R5  00119610   R6  00000000
              R7  00119AF2   R8  00119AF0   R9  00117290   R10 001157D8   R11 0011585C   R12 001157D0


-INTERRUPT AT 115968
-PROCEEDING BACK VIA REG 13

IGZCPCO  WAS ENTERED VIA CALL          AT EP .IGZETRM...C23.009.29.88.21.28...O.....
SA    1170F8  WD1 00108001   HSA 00119610   LSA 00000000   RET 501159AA   EPA 801521F8   R0  00119E66
              R1  0011582A   R2  0011492C   R3  00115912   R4  001157A4   R5  00119610   R6  00000000
              R7  00119AF2   R8  00119AF0   R9  00117290   R10 001157D8   R11 0011585C   R12 001157D0

GO        WAS ENTERED VIA CALL          AT EP XYZSUB...C2.1.3.0.10.03.92.11.20.20
SA    119610  WD1 00108001   HSA 00114FB0   LSA 001170F8   RET 501177FE   EPA 00115750   R0  00119754
              R1  001197C0   R2  501177F4   R3  0011492C   R4  00119830   R5  401177C8   R6  008F6FF8
              R7  001143C0   R8  00114580   R9  00119AE0   R10 00117374   R11 001175B8   R12 00117360
```

After a program interrupt occurs (ABEND), the Save Areas are re-established until the operating system level is reached. The contents of these registers can be used to double check absolute addresses within the programs involved.

```
-REGS AT ENTRY TO ABEND
     FLTR 0-6     0000000000000000   0000000000000000   0000000000000000   0000000000000000
     REGS 0-7     00119754   001197C0   00119AF2   0011492C   001157A4   0011492C   00158700   00119AF2
     REGS 8-15    00119AF0   00117290   001157D8   0011585C   001157D0   001170F8   001158A2   8013E398
```

Following the initial DUMP information is the hexidecimal printout of the contents of memory. This displays the values stored in memory at the time of ABEND.

```
ACTIVE LOAD MODULES
LPA/JPA MODULE     IGZCTCO
115520                                                      00000000 0000C9C7  *                        .....IG*
115540   E9C3E3C3 D6404000 C3F2F34B F0F0F961   F2F961F8 F840F2F0 4BF4F540 4040D500  *ZCTCO  .C23.009.29.88 20.45   N.*
115560   00000000 C3F2E3C8 C4C3D6D4 000001EB   C9000110 00002000 008EF6B0 00114878  *....C2THDCOM...YI.........6.....*
115580   0011492C 00157CC8 00000000 C5E7C1D4   D7D3C540 00000000 00000000 00000000  *.......H....EXAMPLE ............*
1155A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      LINE 1155C0 SAME AS ABOVE
1155E0   00000000 00000000 008EF600 00000001   00114BB0 00157080 00000000 00000000  *...........6....................*
```

```
115600   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
115620   F0F0F0F0 F0F0F0F0 00000000 00000000   00000000 00000000 00000000 00000000   *00000000.......................*
115640   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 115660 SAME AS ABOVE
115680   00000000 00000000 D6010000 E4C50000   00100001 00114C56 00157020 4014C972   *........O...UE............. .I.*
1156A0   00000000 00000000 00157CCB 00012750   00114BF8 001172E0 00000000 00114FB0   *............H.......8.........*
1156C0   00114878 00115564 00000000 00114BF8   0011492C 4014C8C2 00000000 00157CC0   *................8.... .HB........*
1156E0   00000000 00000C00 001570B0 00157CB0   001570B0 00000000 00157000 00002000   *................................*
115700   00001000 00001000 00002000 00002000   00158738 00000000 00000000 00000000   *................................*
115720   00000000 00004000 00004000 00004000   00000000 00000000 00000000 00000000   *................................*
115740   00000000 00000000 00000000 00000000                                         *...... ... ... ................*
                                                                                      *................               *
```

The absolute value of the beginning of the Load Module can also be determined by locating the MODULE name in the listing. As before, the module name is found in the JCL. In this example, the member name in the Partitioned Data Set &&GOSET is GO, and can be found at absolute address 115750.

Since XYZSUB is found before EXAMPLE in the Load Module, as documented on the Link Edit page, the first program in the Executable code is XYZSUB.

```
LPA/JPA MODULE    GO
115740                                                                                47F0F070 23E7E8E9 E2E4C240 4040C3F2   *                   .00..XYZSUB   C2*
115760   40F14BF3 4BF040F1 F061F0F3 61F9F240   F1F14BF2 F04BF2F0 001157A4 69683C5C   * 1.3.0 10.03.92 11.20.20........*
115780   00000000 40000100 00000000 08000000   00000000 00000008 00000005 00000000   *.... ..........................*
1157A0   40404040 00115750 001157D0 001170F8   00115755 00115750 001158A2 00119BD8   *     ..........8................Q*
1157C0   90ECD00C 5810F028 98EFF068 07FF0000   001157D8 0011585C 001158A2 00000001   *......O...O........Q............*
1157E0   00000000 E2E6E260 E2E3C1D9 E3E7E8E9   E2E4C240 40E2E8E2 D6E4E340 40E2E6E2   *....SWS.STARTXYZSUB  SYSOUT  SWS*
115800   60C5D5C4 00035C03 8F1F0000 00000000   012C0000 00010000 01300000 00030000   *.END...........................*
115820   00000000 00008000 00004080 00000000   25C00001 40000808 00000000 15800000   *................................*
115840   000040C0 00014000 06080000 00001502   40000808 00000000 15404040 D2089000   *.. ..... ........ .......   K...*
115860   A00C92F0 9009D202 900BA02C D206900E   A0255820 0128D203 2000A008 D2032008   *...O..K.....K......J.K....K...*
115880   A008D203 2010A008 D2072018 A01DD203   2020A008 920E2028 920F2030 5830D16C   *..K.....K.....K................J.*
1158A0   07F358A0 C0005890 D12C5880 D1345870   D1385820 10004120 20005020 D1341882   *.3......J...J...J...........J...*
1158C0   58201004 41202000 50200138 1872D203   D0ECA004 9180D174 58B0C004 4780B098   *..........J...K.......J........*
1158E0   05205020 D13C5820 D05C58F0 21CC4110   A06505EF 96800174 91400174 4710B0BC   *....J......0.........J.. J.....*
115900   D2030170 D16C4120 B0B65020 D16C47F0   B000D203 D16C0170 96400174 D201D18E   *K.J.J......J..O..K.J.J.. J.K.J.*
115920   8000960F 018FD202 D1957000 960FD197   FA21D195 D18ED202 7000D195 940F7000   *....J.K.J.....J...J.J.K...J.....*
115940   960F7002 FA20900B A031940F 900BF822   900B900B F921900B A02F58B0 C00447D0   *.............8.....9...........*
115960   B116FA20 900B900A 940F900B F822900B   900B47F0 B1369120 D05447E0 B1360520   *............8......0...........*
115980   5020013C 5820D05C 58F021CC 4110A053   05EF947F 01740520 5020D13C 5820D05C   *..J......0.........J......J.....*
1159A0   58F02224 4110A052 05EF4040 D7D9D6C7   E2E4D440 E7E8E9E2 E4C24040 00115A28   *.0........  PROGSUM XYZSUB  ...*
1159C0   0000001E 00115A48 000004CD 00000465   001160F4 00000000 001160F4 00000400   *.................4.......4....*
1159E0   00000000 0000012C 00000130 00000000   00000000 00000124 00000000 00000000   *................................*
115A00   00000000 00000003 0000025A 00000152   00000000 2000006C 00000000 00000000   *................................*
115A20   D7E3C1C2 00000005 00021200 01CC0002   220001F4 00023200 02040002 34000212   *PTAB.............4.............*
115A40   00025200 02224040 00000000 01010034   00000003 00000000 00000000 00000039   *......  ......................*
115A60   00000000 00000024 08000000 00020000   00000002 04000000 80800000 00000000   *................................*
115A80   01010034 00000006 00000000 00000000   0000006D 00000000 00000024 0B000000   *................................*
115AA0   00020000 00000002 04000000 80800000   00000000 02010034 00000009 00000000   *................................*
115AC0   00000000 000000A1 00000000 00000024   0C000000 00020000 00000008 00000000   *................................*
115AE0   40000000 00000000 01010034 0000000C   00000000 00000000 000000D5 00000000   *  ........................N....*
115B00   00000024 05000010 00028000 00000004   08000000 80800000 00000000 01010034   *................................*
115B20   0000000F 00000000 00000000 00000109   00000000 00000024 05000010 08028000   *................................*
115B40   00000004 08000000 80800000 00000000   01010034 00000012 00000000 00000000   *................................*
115B60   0000013D 00000000 00000024 05000010   10028000 00000004 05000000 80800000   *................................*
115B80   00000000 02010034 00000015 00000000   00000000 00000171 00000000 00000024   *................................*
115BA0   05000010 18028000 00000008 00000000   40000000 00000000 01010034 00000017   *................................*
115BC0   00000000 00000000 000001A5 00000000   00000024 05000010 20028000 00000004   *.............. .................*
115BE0   05000000 80000000 00000000 02010034   0000001A 00000000 00000000 000001D9   *................................R*
115C00   00000000 00000024 05000010 28028000   00000001 00000000 40000000 00000000   *.......................... .......*
115C20   02010034 0000001D 00000000 00000000   0000020D 00000000 00000024 05000010   *.......................... .......*
115C40   30028000 00000001 00000000 40000000   00000000 02010034 00000020 00000000   *.......... ...................*
115C60   00000000 00000000 00000000 00000024   09000000 00020000 00000008 00000000   *.......... ...................*
115C80   40000000 00000000 14000084 00000022   00000000 00000000 00000000 00000000   *.  ............................*
115CA0   00000800 00000000 00011800 00000000   00000000 00000001 00000001 00000000   *................................*
115CC0   00000000 00000000 00000005 000004CD   00000000 00000000 00000000 00000000   *................................*
115CE0   00000000 80000002 00000001 00000000   000002C5 00000000 00000000 00000000   *.....................E..........*
115D00   00000000 00000000 00000000 03010034   00000025 00000000 00000000 00000000   *.....................E..........*
115D20   00000000 00001340 02000000 00050000   00000015 00000000 40000000 00000000   *.......... ...................*
115D40   02020034 00000028 00000000 00000000   0000032D 000002C5 00001440 02000000   *.......................E... ....*
115D60   00058000 00000009 00008000 40000000   00000000 01020034 0000002A 00000000   *.......... ...................*
115D80   00000000 00000361 000002C5 00001540   02000000 09058000 00000001 01008000   *...........E..................*
115DA0   08000000 00000000 01020034 0000002C   00000000 00000000 00000395 000002C5   *.....................................E*
115DC0   00001640 02000000 0A058000 00000001   01000000 10800000 00000000 01020034   *.......... ...................*
115DE0   0000002E 00000000 00000000 000003C9   000002C5 00001740 02000000 0B05C000   *...............I...E... ........*
115E00   00000003 04008000 10800000 00000000   02020034 00000031 00000000 00000000   *.......... ...................*
115E20   00000000 000002C5 00001840 02000000   0E058000 00000007 00008000 40000000   *.......E... ..................*
115E40   00000000 01010034 00000033 00000000   00000000 00000431 00000000 00001B40   *.......... ...................*
115E60   03000010 00068000 00000002 03000000   10000000 00000000 01010034 00000035   *.......... ...................*
115E80   00000000 00000000 00000000 00000000   00001C40 03000020 0006C000 00000003   *.......... ...................*
115EA0   04000000 10000000 00000000 12000034   00000038 00000000 00000000 00000499   *.......... ...................*
115EC0   00000000 00002000 00000100 C0070010   00000100 00000000 00000000 00000000   *.......... ...................*
115EE0   13000034 0000003C 00000000 00000000   00000000 00000465 00002400 00000101   *.......... ...................*
115F00   16070010 00000200 00000000 00000000   00185A44 00000499 00000000 2012F0F1   *.......... ...................01*
115F20   F060D4C1 C9D560D3 D6D6D760 C5E7C9E3   00000000 00000465 00000000 200DF0F0   *O.MAIN.LOOP.EXIT..............00*
115F40   F060D4C1 C9D560D3 D6D60700 00000431   00000033 4001E800 00000000 000003FD   *O.MAIN.LOOP......... Y........*
115F60   00000000 4001E700 00000000 000003C9   00000000 4007E2E6 E260C5D5 C4000000   *.... .X........I.... .SWS.END...*
115F80   00000000 00000395 0000002C 4006C3D9   C1E2C8F2 00000361 00000022 4006C3D9   *............ .CRASH2......... .CR*
115FA0   C1E2C8F1 0000032D 00000000 4005C4E4   D4D4E800 000002F9 00000000 4009E2E6   *ASH1......... .DUMMY....9.... .SW*
115FC0   E260E2E3 C1D9E300 00000000 000002C5   00000009 400CE2E8 E2E3C5D4 60C6D3C1   *S.START........E.... .SYSTEM.FLA*
115FE0   C7E20000 00000241 00000000 0806E7E8   E9E2E4C2 0000020D 00000000 4007C3D6   *GS............XYZSUB......... .CO*
116000   D460D9C5 C7000000 00000000 000001D9   00000000 4008E2C8 C9C6E360 C9D50000   *M.REG..........R.... .SHIFT.IN..*
116020   00000000 000001A5 00000000 4009E2C8   C9C6E360 D6E4E300 00000000 00000171   *............ .SHIFT.OUT.........*
```

```
116040   00000000 4005E3C1 D3D3E800 0000013D   00000000 400CE2D6 D9E360D4 C5E2E2C1   *.... .TALLY......... .SORT.MESSA*
116060   C7C50000 00000109 00000000 400EE2D6   D9E360D4 D6C4C560 E2C9E9C5 00000005   *GE........... .SORT.MODE.SIZE...N*
116080   00000000 400EE2D6 D9E360C6 C9D3C560   E2C9E9C5 000000A1 00000000 400EE2D6   *.... .SORT.FILE.SIZE......... .SO*
1160A0   D9E360C3 D6D9C560 E2C9E9C5 0000006D   00000000 400CE2D6 D9E360C3 D6D5E3D9   *RT.CORE.SIZE........ .SORT.CONTR*
1160C0   D6D30000 00000039 00000000 400BE2D6   D9E360D9 C5E3E4D9 D5000000 00000005   *OL........... .SORT.RETURN........*
1160E0   00000000 400BD9C5 E3E4D9D5 60C3D6C4   C5000000 00000035 00000000 00000000   *.... .RETURN.CODE................*
116100   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116120-1161A0 SAME AS ABOVE
1161C0   00000000 00000000 00000000 00000000   00000000 00000000 00000028 00000000   *................................*
1161E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116200-116360 SAME AS ABOVE
116380   00000000 00000000 00000000 00000000   00000000 00000000 00000000 0000001A   *................................*
1163A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 1163C0-1163E0 SAME AS ABOVE
116400   00000000 00000000 00000000 00000006   00000000 00000000 00000000 00000000   *................................*
116420   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116440   00000000 0000002A 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116460   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116480-116500 SAME AS ABOVE
116520   00000038 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116540   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116560-116760 SAME AS ABOVE
116780   00000000 00000000 00000000 00000000   00000003 00000000 00000000 00000000   *................................*
1167A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 1167C0-116840 SAME AS ABOVE
116860   00000000 00000000 00000000 00000000   00000000 00000000 00000000 0000002E   *................................*
116880   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 1168A0-116A40 SAME AS ABOVE
116A60   00000000 00000000 00000000 00000000   00000000 00000031 00000000 00000000   *................................*
116A80   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116AA0-116AC0 SAME AS ABOVE
116AE0   00000000 00000000 00000000 00000000   0000000F 00000000 00000000 00000000   *................................*
116B00   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116B20-116C80 SAME AS ABOVE
116CA0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 0000003C   *................................*
116CC0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116CE0-116E40 SAME AS ABOVE
116E60   00000000 00000000 00000000 00000000   00000000 00000020 00000000 00000000   *................................*
116E80   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116EA0   00000000 00000000 00000000 00000000   00000000 00000012 00000000 00000000   *................................*
116EC0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 116EE0 SAME AS ABOVE
116F00   00000000 00000000 0000000C 00000000   00000000 00000000 00000000 00000015   *................................*
116F20   00000025 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116F40   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116F60   0000001D 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
116F80   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 116FA0-117020 SAME AS ABOVE
117040   00000017 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
117060   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 117080-1170C0 SAME AS ABOVE
```

The TGT for XYZSUB begins at 1170F8, with the 18 word Save Area. Comparing these entries with the Save Area Trace provided above confirms that the contents are used for referencing locations in memory.

```
1170E0   00000000 00000000 00000000 00000000   00000000 00000000 00108001 00119610   *................................*
1171000  00000000 501159AA 801521F8 00119E66   0011582A 0011492C 00115912 001157A4   *............8...............*
1171200  00119610 00000000 00119AF2 00119AF0   00117290 001157D8 0011585C 00115700   *............2...0.......Q........*
1171400  C3F2E3C7 E34EF4F8 02000000 49000A20   00157CC8 0011492C 00000000 00000000   *C2TGT.48...........H............*
1171600  00000015 00119610 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
1171800  40400000 00000000 00000001 00119E66   E2E8E2D6 E4E34040 C9C7E9E2 D9E3C3C4   *  ..............SYSOUT  IGZSRTCD*
1171A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 1171C0 SAME AS ABOVE
```

Found at 1171F8 is the absolute address of the beginning of the COBOL program XYZSUB.

```
1171E0   00000000 00000001 0011726C 00000000   0011580A 00117234 00115750 001157D8   *...............................Q*
```

Working Storage Address XYZSUB

```
1172000  00117264 001157D4 00000000 00117290   00000000 40404040 40404040 40404040   *.......M...........            *
```

Base Locator for WS

```
                                              Base Locators for Linkage Section
1172200  001172A8 00117290 00000000 00119AF0   00119AF2 50115998 001159B4 07FE07FE   *...............0...2...........*
1172400  00000000 00000000 00001FFF 07FE0000   00000000 00000000 00000000 00000000   *................................*
1172600  00000000 001158A2 001158A2 C0000000   00000000 00000000 00000000 00000000   *................................*
                                              Beginning of Working Storage XYZSUB
1172800  00000000 0000001F 00000000 0000004C   E2E6E260 E2E3C1D9 E3F00000 039CE2E6   *...............SWS.STARTO....SW*
1172A0   E260C5D5 C4000000 00000000 00000000   00000000 00000000 00000000 00000000   *S.END..........................*
1172C0   E2E8E2D6 E4E34040 00000000 00000000   0E000000 00000000 0F000000 00000000   *SYSOUT ........................*
```

EXAMPLE begins at absolute address 1172E0.

```
1172E0   47F0F070 23C5E7C1 D407D3C5 4040C3F2   40F14BF3 4BF040F1 F061F0F3 61F9F240   *.00..EXAMPLE  C2 1.3.0 10.03.92 *
1173000  F1F14BF2 F04BF3F2 00117334 69683C5C   00800000 12000A0B 80204000 00000000   *11.20.32.................. .....*
1173200  00800000 0000000D 0000000E 00000000   40404040 001172E0 00117360 00119610   *...............  ............*
1173400  001172E5 001172E0 0011761C 00119BD8   90ECD00C 5810F028 98EFF068 07FF0000   *...V...........Q......0...0.....*
1173600  40404040 40400000 0F404040 00117374   001175B8 00115750 0011761C 00117734   *       ...  ......................*
1173800  0011780E 00117844 00000001 00000000   4040E3C8 C9E240E5 C1D3E4C5 40C1C3C3   *................ THIS VALUE ACC*
1173A0   E4D4E4D3 C1E3C5C4 40C9D540 E2E4C2E7   E8E94040 404040E3 D6E3C1D3 40D9C5C3   *UMULATED IN SUBXYZ     TOTAL REC*
1173C0   D6D9C4E2 4007D9D6 C3C5E2E2 C5C4407E   C5E7C1D4 D7D3C540 E2E8E2D6 E4E34040   *ORDS PROCESSED .EXAMPLE SYSOUT  *
1173E0   E6E260E2 E3C1D9E3 E6E260C5 D5C4E8C5   E2D5D600 1F000001 3C000000 02000001   *WS.STARTWS.ENDYESNO............*
1174000  34000000 01000001 38000000 01000000   00000000 00800000 00402020 20202008   *.......................... .....*
1174200  00000004 00000040 00004080 00000000   25C00001 40000808 00000C00 5C400002   *....... .. ........ ........ ..*
```

```
117440  00018000 80000180 00800002 80008080  00000000 40C00001 40000708 00000C00  *.................... ..........*
117460  5C024000 08080000 0C005C09 00014000  00090001 3C000000 C6C9C200 0102C6C9  *.. ............ .........FIB...FI*
117480  D3C5F0F1 40400084 80804400 00009000  00000000 00000050 00010000 00000000  *LE01  ......................*
1174A0  00000000 00000000 42041000 00000000  00117471 00000000 00000000 00000000  *.............................*
1174C0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *.............................*
1174E0  00000000 00000000 00000000 00000000  00000000 00000000 0000C4C9 E2D260C6  *.............................*
117500  C9D3C540 40404040 40404040 40404040  40404040 40404040 C6C9C200 0202C3C1  *.........................DISK.F*
117520  D9C4C9D5 40400084 80800400 00008000  00000050 00000050 00010000 00000000  *ILE                     FIB...CA*
117540  00000000 00000000 82080000 00000000  0011746B 00000000 00000000 00000000  *RDIN  ......................*
117560  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *.............................*
117580  00000000 00000000 00000000 00000000  00000000 00000000 0000C3C1 D9C460C6  *.............................*
117A0   C9D3C540 40404040 40404040 40404040  40404040 40404040 D2079000 A06CD201  *.........................CARD.F*
117C0   9008A07D 9240900A D2019010 A07FD202  9012C006 D21D9018 A03ED221 903AA01C  *ILE                     K.....K.*
117E0   D205905C C000D205 90B8A074 5820D130  D2032000 A018D203 2008A018 D2032010  *..... ..K.....K.....K.....K.....*
117600  A018D207 2018A064 02032020 A018920E  2028920F 20305830 017807F3 58A0C00C  *K.....K.......J.K.....K.....K...*
117620  5890D134 5880D140 5870D13C D203D0EC  A0149180 D1A058B0 C0104780 B09A0520  *..K.....K.............J..3....*
117640  5020D148 5820D05C 58F021CC 4110A0DB  05EF9680 D1A09140 D1A04710 B0BE0203  *..J....J ..J.K.........J.......*
117660  D194D178 4120B0B8 5020D178 47F0B000  D203D178 D194D20B D17CA008 964001A0  *..J.......O.........J.. J.....K.*
117680  D203D188 D17C5880 C0104120 B0DE5020  D17C47F0 B136D203 D17CD188 D203D18C  *J.J.......J..O..K.J.J.K.J.... J.*
1176A0  D18058B0 C0104120 B0F65020 D180D502  9008A07A 58B0C010 4770B17C D203D180  *K.J.J...........J..O..K.J.J.K.J.*
1176C0  D18CD203 D190D184 4120B11C 50200184  47F0B256 D203D184 D1900520 5020D148  *J........6..J.N.............K.J.*
1176E0  5820D05C 58F02224 4110A0DA 05EF5820  D19CD202 2034A0A1 05305030 D1485830  *J.K.J.J.......J..O..K.J.J.....J.*
117700  D05C58F0 31E44110 A0D505EF 58B0D140  5840D198 D2024034 A0B30550 5050D148  *.....O.........J.K.........J...*
117720  58F031E4 4110A0D0 05EF5870 D13C5850  D17C07F5 5840D19C 920040C9 920040B3  *...O.U...N....J . J.K.......J.*
117740  D2034074 A0AF0520 5020D148 58F0403C  055F9500 40C94770 B1B25010 D1401881  *.O.U.......J...J..5. J... I.. .*
117760  D24F9068 800047F0 B1EC9510 40C94780  B1C69500 40C94780 B1EC47F0 B1EC0202  *K. .......J..O ..... I......J ..*
117780  9008A07A D205D1B2 A0A5D202 D1BD9012  960F01BF DE05D1B2 D1BDD203 90360184  *K......O.... I...F.. I.....O..K.*
117A0   47F0B250 D24F7000 90685840 D1989200  40C99200 40B30203 4074A0AB 05205020  *....K.J...K.J.....J...J.J.K...J.*
117C0   D14858F0 4040055F 950040C9 4770B21E  5010D13C 18714120 90105020 D1B04120  *.O..K...... J... I.. .K. .......*
117E0   90125020 D1B49680 D1B44110 01B04100  D1440520 5020D148 58F0A000 05EF50F0  *J..O  .... I......J........J...*
117800  D078BF28 D0890420 5820D180 07F25820  D19CD201 2037A0A1 5830D198 D2013037  *....J...J...J..J.....J..O.....0*
117820  A0A10540 5040D148 5840D05C 58F041E4  4110A0C9 05EF5880 01405870 D13C5850  *..........J..2..J.K.......J.K...*
117840  D18407F5 9120D054 47E0B2A4 05505050  D14858F0 41CC4110 A0B705EF 947FD1A0  *... J.. ...O.U...I....J ..J...*
117860  05505050 D14858F0 42244110 A0B605EF  D7D9D6C7 E2E4D440 C5E7C1D4 07D3C540  *J..5............J..O.........J.*
117880  001178EC 00000054 00117940 00000869  00000679 00118500 00118500 00118610  *....J..O........PROGSUM EXAMPLE *
1178A0  00000400 0000013C 00000134 00000138  00000000 00000000 0000012C 00000000  *.............................*
1178C0  00000000 00000198 00000001 00000590  0000033C 00000000 2000006C 00000000  *.............................*
1178E0  00000000 D7E3C1C2 0000000E 00033200  03A00003 420003BC 00035200 03E20003  *.............................*
117900  620003FA 00039200 040E0003 A2000430  0003E200 04540004 0200049E 00041200  *....PTAB.......................S..*
117920  04A40004 220004C0 00045200 04C40004  620004F6 00048200 05280004 B200052E  *...............S.............*
117940  00000000 01010034 00000003 00000000  00000000 00000039 00000000 00000024  *.............D.....6..........*
117960  08000000 00020000 00000002 04000000  80800000 00000000 01010034 00000006  *.............................*
117980  00000000 00000000 00000060 00000000  00000024 0B000000 00020000 00000002  *.............................*
117A0   04000000 80800000 00000000 02010034  00000009 00000000 00000000 000000A1  *.............................*
117C0   00000000 00000024 0C000000 00020000  00000008 00000000 40000000 00000000  *.............................*
117E0   01010034 0000000C 00000000 00000000  000000D5 00000000 00000024 05000010  *.............................*
117A00  00028000 00000004 08000000 80800000  00000000 01010034 0000000F 00000000  *.....................N.......*
117A20  00000000 00000109 00000000 00000024  05000010 08028000 00000004 08000000  *.............................*
117A40  80800000 00000000 01010034 00000012  00000000 00000000 00000130 00000000  *.............................*
117A60  00000024 05000010 10028000 00000004  05000000 80800000 00000000 02010034  *.............................*
117A80  00000015 00000000 00000000 00000171  00000000 00000024 05000010 18028000  *.............................*
117AA0  00000008 00000000 40000000 00000000  01010034 00000017 00000000 00000000  *.............................*
117AC0  000001A5 00000000 00000024 05000010  20028000 00000004 05000000 80000000  *.............................*
117AE0  00000000 02010034 0000001A 00000000  00000000 000001D9 00000000 00000024  *.........................R.....*
117B00  05000010 28028000 00000001 00000000  40000000 00000000 02010034 0000001D  *.............................*
117B20  00000000 00000000 00000200 00000000  00000024 05000010 30028000 00000001  *.............................*
117B40  00000000 40000000 00000000 02010034  00000020 00000000 00000000 00000000  *.............................*
117B60  00000000 00000024 09000000 00020000  00000008 00000000 40000000 00000000  *.............................*
117B80  14000084 00000023 00000000 00000000  00000000 00000000 00000200 00000000  *.............................*
117BA0  00010800 00000000 00000000 00000001  00000001 00000000 00000002 00000004  *.............................*
117BC0  00000005 000007E5 00000000 00000000  00000000 00000000 00000000 80000002  *.............................*
117BE0  00000001 00000000 000003D5 000002C5  00000000 00000000 00000000 00000000  *.......V.....................*
117C00  00000000 0F000050 00000029 00000001  00000000 00000340 00000000 00001040  *...........N...E.............*
117C20  01000000 01048000 00000000 00000315  00000001 20C04400 C5420410 00280000  *.............................*
117C40  00000000 00000000 00000000 00000000  00000000 02010038 0000002C 00000000  *.........................E.....*
117C60  00000000 0000034D 000002C5 00001640  01000000 0004C000 00000050 00000000  *.............................*
117C80  40000000 00005000 00000000 0F000050  00000026 00000021 00000000 00000000  *...........E.................*
117CA0  00000000 00001840 01000100 01048000  00000000 0000039D 00000002 20C04400  *.............................*
117CC0  C5820800 00080000 00000000 00000000  00000000 00000000 00000000 02010038  *.............................*
117CE0  0000002F 00000000 00000000 00000000  00000340 00001D40 01000010 00048000  *E............................*
117D00  00000050 00000000 40000000 00005000  00000000 03010034 00000032 00000000  *.............................*
117D20  00000000 00000471 00000000 00002140  02000000 00050000 00000008 00000000  *.............................*
117D40  40000000 00000000 02020034 00000035  00000000 00000000 0000043D 000003D5  *.............................*
117D60  00002240 02000000 00058000 00000008  00008000 40000000 00000000 02020034  *.........................N...*
117D80  00000037 00000000 00000000 00000471  000003D5 00002340 02000000 0805C000  *.............................*
117DA0  00000003 00008000 40000000 00000000  03010034 0000003A 00000000 00000000  *.....................N.......*
117DC0  0000050D 00000000 00002440 02000000  10050000 00000005 00000000 40000000  *.............................*
117DE0  00000000 01020034 0000003C 00000000  00000000 000004D9 00000471 00002540  *.............................*
117E00  02000000 1005C000 00000002 03008000  10000000 00000000 01020034 0000003E  *.........................R.....*
117E20  00000000 00000000 00000500 00000471  00002640 02000000 1205C000 00000003  *.............................*
117E40  04008000 10000000 00000000 03010034  00000041 00000000 00000000 0000050D  *.............................*
117E60  00000000 00002740 02000000 18050000  0000004A 00000000 40000000 00000000  *.............................*
117E80  02020034 00000043 00000000 00000000  00000575 0000050D 00002840 02000000  *.............................*
117EA0  18058000 0000001E 00008002 40000000  00000000 01020034 00000045 00000041  *.............................*
117EC0  00000000 000005A9 0000050D 00002A40  02000000 3605C000 00000004 04000000  *.............................*
117EE0  0A000000 00000000 02020034 00000043  00000000 00000541 0000050D 0000050D  *.............................*
117F00  00002B40 02000000 3A058000 00000028  00008002 40000000 00000000 03010034  *.............................*
117F20  00000048 00000000 00000000 00000645  00000000 00002D40 02000000 68058000  *.............................*
117F40  00000050 00000000 40000000 00000000  02020034 0000004B 00000000 00000000  *.............................*
117F60  00000645 000005DD 00002E40 02000000  6805C000 00000050 00000000 40000000  *.............................*
117F80  00000000 02010034 0000004D 00000000  00000000 00000000 00000000 00002F40  *.............................*
117FA0  02000000 B8058000 00000005 00008000  40000000 00000000 12000034 00000050  *.............................*
117FC0  00000000 00000000 000006AD 00000000  00003200 00000100 C8070010 00000100  *.............................*
117FE0  00000000 00000000 00000000 12000034  00000054 00000000 00000000 000006E1  *.........................H.....*
118000  00000000 00003800 00000101 36071032  00000200 02000000 00080000 00000000  *.............................*
118020  12000034 00000057 00000000 00000000  00000715 00000000 00003C00 00000101  *.............................*
118040  7C070030 00000300 00000000 00080000  00000000 13000034 00000060 00000000  *.............................*
118060  00000000 00000749 000006E1 00003D00  00000101 7C070010 00000400 00000000  *.............................*
118080  00000000 00000000 13000034 00000069  00000000 00000000 0000077D 000006E1  *.............................*
1180A0  00004400 00000101 EC070010 00000500  00000000 00000000 00000000 13000034  *.............................*
1180C0  00000064 00000000 00000000 000007B1  000006E1 00004700 00000102 50071030  *.............................*
1180E0  00000600 03000000 00280000 00000000  12000034 0000005B 00000000 00000000  *.............................*
118100  00000000 00000000 00004A00 00000102  56071030 00000700 04000000 00080000  *.............................*
```

```
118120   00000000 14000084 0000006B 00000000   00000000 00000000 00000000 00000000   *................................*
118140   00000000 00010000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118160   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 118180 SAME AS ABOVE

1181A0   00000000 0018616F 000007E5 00000000   0806E7E8 E9E2E4C2 00000749 00000000   *...........V......XYZSUB........*
1181C0   2018F2F2 F060D7D9 D6C3C5E2 E260E6D9   C9E3C560 D9C5C3D6 D9C40000 00000000   *..220.PROCESS.WRITE.RECORD......*
1181E0   0000077D 00000000 2010F2F9 F060D7D9   D6C3C5E2 E260C5E7 C9E30000 00000000   *...........290.PROCESS.EXIT......*
118200   00000715 00000000 2017F2F1 F060D7D9   D6C3C5E2 E260D9C5 C1C460D9 C5C3D6D9   *...........210.PROCESS.READ.RECOR*
118220   C4000000 00000000 000007B1 00000000   200FF3F0 F060C3D3 D6E2C560 C6C9D3C5   *D.................300.CLOSE.FILE*
118240   E2000000 00000000 000006E1 00000000   200BF2F0 F060D7D9 D6C3C5E2 E2000000   *S.................200.PROCESS...*
118260   000006AD 00000000 2012F1F0 F060C9D5   C9E3C9C1 D3C9E9C5 60E2E8E2 00000000   *..........100.INITIALIZE.SYS....*
118280   00000679 00000000 200DF0F0 F060D4C1   C9D560D3 D6D6D700 00000645 00000000   *..........000.MAIN.LOOP.........*
1182A0   4006E6E2 60C5D5C4 00000611 00000000   4007C3C1 D9C460C9 D5000000 00000000   * .WS.END........ .CARD.IN.......*
1182C0   000005DD 00000000 4007C4C1 E3C160C9   D5000000 00000000 00000575 00000000   *........ .DATA.IN................*
1182E0   4004D7E3 D6E30000 000005A9 00000017   C006C6C9 D3D3C5D9 0000050D 00000000   * .PTOT...........FILLER........*
118300   4008E3D6 E360D3C9 D5C50000 00000000   00000409 0000003C 4003E3D6 E3000000   * .TOT.LINE.........R.... .TOT...*
118320   000004A5 00000000 4003C1D4 E3000000   00000471 00000000 400AE3D6 E3C1D360   *........ .AMT........... .TOTAL.*
118340   C1D9C5C1 00000000 0000043D 00000000   4003C5D6 C4000000 00000409 00000000   *AREA............ .EOD............*
118360   4008E6E2 60E2E3C1 D9E30000 00000000   000003D5 00000009 400CE2E8 E2E3C5D4   * .WS.START.........N.... .SYSTEM*
118380   60C6D3C1 C7E20000 0000039D 00000000   4008C3C1 D9C460D9 C5C30000 00000000   *.FLAGS........... .CARD.REC......*
1183A0   00000315 00000000 4008C4C9 E2D260D9   C5C30000 00000000 000002C5 00000000   *........ .DISK.REC.........E....*
1183C0   4009C4C9 E2D260C6 C9D3C500 00000000   0000034D 00000000 4009C3C1 D9C460C6   * .DISK.FILE.............. .CARD.F*
1183E0   C9D3C500 00000000 00000241 00000000   0807C5E7 C1D4D7D3 C5000000 00000000   *ILE...............EXAMPLE.......*
118400   0000020D 00000000 4007C3D6 D460D9C5   C7000000 00000000 000001D9 00000000   *........ .COM.REG..........R....*
118420   4008E2C8 C9C6E360 C9D50000 00000000   000001A5 00000000 4009E2C8 C9C6E360   * .SHIFT.IN............. .SHIFT.*
118440   D6E4E300 00000000 00000171 00000000   4005E3C1 D3D3E800 0000013D 00000000   *OUT............. .TALLY........*
118460   400CE2D6 D9E360D4 C5E2E2C1 C7C50000   00000109 00000000 400EE2D6 D9E360D4   * .SORT.MESSAGE........... .SORT.M*
118480   D6C4C560 E2C9E9C5 000000D5 00000000   400EE2D6 D9E360C6 C9D3C560 E2C9E9C5   *ODE.SIZE...N.... .SORT.FILE.SIZE*
118480   000000A1 00000000 400EE2D6 D9E360C3   D6D9C560 E2C9E9C5 0000006D 00000000   *........ .SORT.CORE.SIZE........*
1184C0   400CE2D6 D9E360C3 D6D5E3D9 D6D30000   00000039 00000000 400BE2D6 D9E360D9   * .SORT.CONTROL........... .SORT.R*
1184E0   C5E3E4D9 D5000000 00000005 00000000   400BD9C5 E3E4D9D5 60C3D6C4 C5000000   *ETURN........... .RETURN.CODE...*
118500   00000000 05000002 C5000000 00000000   00000000 50000000 50000000 00000000   *........E.......................*
118520   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118540   00000000 00000000 00000000 00000000   00000000 000000E2 40C6C9D3 C5F0F140   *.......................S FILE01 *
118560   40000000 00000000 00984000 00000000   00000000 50000000 50000000 00000000   * ........ ......................*
118580   00000000 05000003 4D000000 00000000   00000000 50000000 50000000 00000000   *................................*
118580   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118A0   00000000 00000000 00000000 00000000   00000000 000000E2 40C3C1D9 C4C9D540   *.......................S CARDIN *
118C0   00000000 00000000 00000000 00000000   00000000 50000000 50000000 00000000   * ........ ......................*
118E0   40000000 00000000 00004000 00000000   00000037 00000000 00000000 00000000   *...........Z....................*
118600   00000000 01000000 00C00003 E9000400   00000000 00000000 00000000 0000005B   *................................*
118620   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118640   00000000 00000000 00000000 00000000
      LINES 118660-118840 SAME AS ABOVE

118860   00000000 0000002C 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118880   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118A0   00000000 00000029 00000000 00000000   00000000 00000000 0000001A 00000000   *................................*
118C0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 118E0 SAME AS ABOVE

118900   00000000 00000000 00000000 00000000   00000000 00000048 00000000 00000000   *................................*
118920   00000000 0000004D 00000006 00000000   00000000 00000000 00000000 00000000   *................................*
118940   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 118960-118A00 SAME AS ABOVE

118A20   00000000 00000000 00000000 00000000   00000000 00000000 00000060 00000050   *................................*
118A40   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 118A60 SAME AS ABOVE

118A80   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000057   *................................*
118AA0   00000000 00000041 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118AC0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 118AE0-118C40 SAME AS ABOVE

118C60   00000000 00000000 00000000 00000000   00000000 00000000 0000003A 00000000   *................................*
118C80   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
118CA0   00000000 00000000 00000000 00000003   00000000 00000000 00000000 00000000   *................................*
118CC0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 118CE0-118D60 SAME AS ABOVE

118D80   00000000 00000000 00000000 00000000   00000000 00000000 0000006B 00000000   *................................*
118DA0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 118DC0-118FE0 SAME AS ABOVE

119000   00000000 00000000 00000000 0000000F   00000000 00000000 00000000 00000000   *................................*
119020   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 119040-119100 SAME AS ABOVE

119120   0000004B 00000035 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119140   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119160   00000023 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119180   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000045   *................................*
1191A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
1191C0   00000000 00000000 00000054 00000069   00000000 00000000 00000000 00000000   *................................*
1191E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119200   00000000 00000000 00000000 00000000   0000003E 00000000 00000000 00000000   *................................*
119220   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 119240-119360 SAME AS ABOVE

119380   00000000 00000000 00000000 00000000   00000020 00000000 00000000 00000000   *................................*
1193A0   00000000 00000000 00000064 00000000   00000000 00000000 00000000 00000000   *................................*
1193C0   00000000 00000000 00000000 00000000   00000012 00000000 00000000 00000000   *................................*
1193E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 119400 SAME AS ABOVE

119420   00000000 0000000C 00000000 0000002F   00000000 00000000 00000015 00000032   *................................*
119440   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119460   00000000 00000000 00000000 00000026   00000000 00000000 00000000 0000001D   *................................*
119480   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 1194A0-119520 SAME AS ABOVE

119540   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000043   *................................*
119560   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINES 119580-1195E0 SAME AS ABOVE
```

The TGT for EXAMPLE begins at 119610, with the 72 byte Save Area.

```
119600   00000000 00000000 00000000 00000000   00108001 00114FB0 001170F8 501177FE   *...........................8....*
119620   00115750 00119754 001197C0 501177F4   0011492C 00119830 401177C8 008F6FF8   *...............4..........H...8*
119640   001143C0 00114580 00119AE0 00117374   00117588 00117360 C3F2E3C7 E34EF4F8   *.......................C2TGT.48*
119660   02000000 6D110220 00157CC8 0011492C   00000000 00000002 000000BE 00000000   *...........H...................*
119680   00000000 00158688 00000000 00000000   00000000 00000000 40400000 00000000   *.......................  ......*
```

```
1196A0   00000004 00119E66 E2E8E2D6 E4E34040   C9C7E9E2 D9E3C3C4 00000000 00000000   *........SYSOUT  IGZSRTCD........*
1196C0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
1196E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000001   *................................*
```

Address of Start of COBOL Program EXAMPLE

```
119700   00119780 00000000 001173F5 00119758   001172E0 00117378 00119788 00117370   *...........5....................*
```

Working Storage Address

```
119720   001197A8 00119AE0 00000000 40404040   40404040 40404040 40404040 40404040   *............                    *
```

Base Locator for Working Storage EXAMPLE

```
119740   00119BA0 00119AE0 00000000 001143C0   00114580 00158700 501177F4 00117878   *...........................4....*
119760   07FE07FE 00000000 00000000 00001FFF   07FE0000 00000000 00000000 00000000   *................................*
119780   00000000 00000000 0011761C 00117734   001176AE 00117844 00117734 0011780E   *................................*
119A0    40404040 0011761C 00119830 00119990   C0000000 00000000 00000000 00119A90   *    ........................... *
```

Temporary Storage 2

```
1197C0   00119AF0 80119AF2 00000000 00000000   00000000 00000000 00000000 00000000   *...0...2........................*
1197E0   00000000 011143B8 00004000 00114560   46000001 90157ED4 00CC0048 00928548   *.......... ............M........*
119800   92BA899A 00000000 0014BB76 08090050   00000000 00114410 00114410 00114410   *................................*
119820   00000050 00000000 00000000 00000000   C6C3C200 01020000 FFFFFFFF FFFFFFFF   *................FCB.............*
119840   FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF   FFFFFFFF 00000000 00000000 00000000   *................................*
119860   00000000 00000000 00000000 8012EB48   0014B80E 8012EB48 8012EB48 00000000   *................................*
119880   00000000 8012EB48 0014B80E 00000000   00000000 00000000 40000000 00000000   *................................*
1198A0   00400010 08000000 00119700 00000000   00000000 00000000 00000000 00000000   *. ..............................*
1198C0   00000000 00000000 00000000 00000000   00000000 00117478 00000000 00000000   *................................*
1198E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119900   00000000 00000000 00000050 00000000   00000000 00000000 00000000 00000000   *................................*
119920   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.......... ....................*
119940   00000000 05114578 00504000 00114860   4614BA54 80157ED4 00B84800 00928600   *.......... ............M........*
119960   12BA898E 00F85DC0 0014BB76 08090050   00000000 00114710 001145D0 00114580   *.....8.........................*
119980   00000050 00000000 00000000 00000000   C6C3C200 02020000 FFFFFFFF FFFFFFFF   *................FCB.............*
119A0    FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF   FFFFFFFF 00000000 00000000 00000000   *................................*
119C0    00000000 00000000 00000000 0014B7BE   8012EB48 8012EB48 8012EB48 00000000   *................................*
119E0    00000000 8012EB48 00000000 00000000   00000000 00000000 80000000 00000000   *................................*
119A00   00000010 04000000 00119930 00000000   00000000 00000000 00000000 00000000   *................................*
119A20   00000000 00000000 00000000 00000000   00000000 00117518 00158690 00000000   *................................*
119A40   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
119A60   00000000 00000054 00000050 00000000   00000000 00000000 00000000 00000000   *................................*
119A80   00000000 00000000 00000000 00114580   00000000 00000000 00000000 00000000   *................................*
119AA0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
      LINE 119AC0 SAME AS ABOVE
```

Working Storage for EXAMPLE

amt tot

```
119AE0   E6E260E2 E3C1D9E3 D5D64000 00000000   001F0000 4F000000 40404040 40E3D6E3   *WS.STARTNO .............      TOT*
119B00   C1D340D9 C5C3D6D9 C4E240D7 D9D6C3C5   E2E2C5C4 407E0000 00004040 E3C8C9E2   *AL RECORDS PROCESSED .....  THIS*
119B20   40E5C1D3 E4C540C1 C3C3E4D4 E4D3C1E3   C5C440C9 D540E2E4 C2E7E8E9 40404040   * VALUE ACCUMULATED IN SUBXYZ    *
119B40   40400000 00000000 D9C5C3D6 D9C440D5   E4D4C2C5 D940F440 40404040 40404040   *  ......RECORD NUMBER 4         *
119B60   40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040   *                                *
119B80   40404040 40404040 40404040 40404040   40404040 40404040 E6E260C5 D5C40000   *                        WS.END..*
119BA0   00000000 00000000 00000000 00000000   00000000 00000000 E2E8E2D6 E4E34040   *                       SYSOUT   *
119BC0   00000000 00000000 0E000000 00000000   0F000000 00000000 47F0F02C 2740C9C7   *.........................00.. IG*
119BE0   E9C5C2E2 E3404200 C3F2F34B F0F0F961   F2F961F8 F840F2F0 4BF5F040 4040D600   *ZEBST ..C23.009.29.88 20.50   O.*
119C00   00000000 18CF1841 D503D000 C41B4770   C04A1F11 5850D05C 58F051C0 05EF47F0   *........N...D..............O.....O*
119C20   C062D503 D000C41F 4770C062 41100004   5850A044 58F051C0 05EF0501 D000C413   *..N...D..............O....N...O.*
119C40   4770C0B0 9101D003 47E0C0B0 91B8D002   4780C12A 9120D002 4780C08C 5880A040   *.................A..............*
119C60   47F0C12A 9180D002 4780C0A0 5850D058   58805040 47F0C12A 9108D002 47E0C0AC   *.OA................ .OA........*
119C80   58807078 47F0C12A D503D000 C4174770   C0C65850 D1885880 50B047F0 C12A5890   *.....0A.N...D....F..J......0A...*
119CA0   40009140 902E4770 C0EE5800 40081850   91405054 47E0C0EE 18505850 50585880   *  ..... ..... ..... ..... .....*
119CC0   504047F0 C12A0700 4100C0F8 47F0C100   C9C7E9C3 E3C3D640 1B110A08 4180002C   *. .0A......8.0A.IGZCTCO ........*
119CE0   1E809180 800C4710 C11C5890 80181299   4780C12A 4500C128 C9C7E9C3 E3C3D640   *........A........A...A.IGZCTCO *
119D00   0A091F00 19804780 C1445850 80181950   4780C144 58B0505C 47F0C146 1FBB185D   *.......A........A......0A.....*
119D20   1F001980 4780C176 5890808C 19904780   C1769101 800C47E0 C1761819 18211BD1   *......A........A.......A......J*
119D40   D7302048 204858F0 B1C047F0 C1C44100   04004510 C17E0A0A 18211BD1 07302048   *P......0...0AD......A......JP...*
119D60   2048128B 4770C1C0 4100C198 47F0C1A0   C9C7E9C5 C9D5C940 1B110A08 18F09680   *......A...A..0A.IGZEINI .....0..*
119D80   207895D6 F0264780 C1BCD203 208FC408   45E0C360 47F0C1C4 58F0B1C0 5050D004   *...00...A.K...D...C..0AD.0......*
119DA0   5800D004 185050D0 5008D203 D000C413   9610D002 D203D008 C3F841A0 20485800   *..........K...D.....K...C8......*
119DC0   C3FC5000 A0285850 C4005050 A02C5040   A0044100 C28E5000 A0085020 A00C9601   *C........D...... ....B..........*
119DE0   A0000510 121147B0 C21A4150 20489680   50004110 204805EF 18919180 207847E0   *........B.......................*
119E00   C2384500 C236C9C7 E9C5C9D5 C9400A09   5850D004 50509004 50905008 1F885080   *B...B.IGZEINI ..................*
119E20   900858A0 90585880 A0405920 808C4780   C2661812 41000400 41101000 0A0A9120   *......... ......B...............*
119E40   905447E0 C27A5850 90585810 503047F0   C2825850 90045810 50181809 58C04004   *....B..........0B..........R.. .*
119E60   58E04014 07FE90EC D00C18CF 41E0028E   1FCE185D 182D5E20 D0085020 50081BD2   *.. ............................K*
119E80   D203D000 C4135050 D004D203 D008C3F8   18419180 400047E0 C3180700 4100C2CC   *K...D.....K...C8.... ...C.....B.*
119EA0   47F0C2D4 C9C7E9C5 D7C3D340 1B110A08   18F095D6 F0264780 C2ECD203 D08FC40C   *.0BMIGZEPCL .....0.00...B.K...D.*
119EC0   45E0C360 4110D093 D708D093 D0934160   D0939681 60005800 40045000 600405EF   *..C.....P.......................*
119EE0   4500C314 C9C7E9C5 D7C3D340 0A0918D5   58204008 5800400C 50002010 91204000   *..C.IGZEPCL ...N.. ... ........ .*
119F00   47E0C334 41500001 47F0C336 1F559140   400047E0 C34C5810 40104100 04004110   *..C.......OC....  ...C... .......*
119F20   10000A0A 18D21255 4780C35A 98ECD00C   0B0E98EC D00C07FE D203D07C C3F4D203   *.....K....C.............K...C4K.*
119F40   D080C3EC D203D084 C3F0D202 D08CC410   4100D08C 5000D088 4110D07C 45F0C398   *..C.K...C0K...D.............0C.*
119F60   00119F68 00000000 C9C7E9C5 E6E3D640   0A0607FE D7C1E3C3 C840C1D9 C5C14060   *.........IGZEWTO ....PATCH AREA .*
119F80   40C9C7E9 C5C2E2E3 4040F8F8 4BF2F7F3   C3B8C3BA C3BCC3BE C3C0C3C2 C3C4C3C6   * IGZEBST  88.273C.C.C.C.C.CBCDCF*
119FA0   C3C8C3CA C3CCC3CE C3D0C3D2 C3D4C3D6   C3D8C3DA C3DCC3DE C3E0C3E2 C3E4C3E6   *CHC.C.C.C.CKCMCOCQC.C.C.C.CSCUCW*
119FC0   C3E80000 00000001 00000007 0000000B   000000A0 00000000 00000000 00000000   *CY..............................*
119FE0   C5C9D5C9 C5D7C3D3 C9C7E900 10000100   30000000 10800100 10240100 00000000   *EINIEPCLIGZ.....................*
LPA/JPA MODULE    IGZCPAC
```

This DUMP contains a plethora of material, most of which is not relevant to the singular ABEND. However, when you become accustomed to navigating through the sheer volume of data and are able to extract the key item, the dump becomes a valuable source of information. It is your challenge to see how the key piece interacts with the rest of the program so that further ABENDs can be avoided.