

```

GGGGGGGGGG  AAAAAAAAAA  WW      WW  AAAAAAAAAA  SSSSSSSSSS  MM      MM  LL
GGGGGGGGGGGG  AAAAAAAAAAAA  WW      WW  AAAAAAAAAAAA  SSSSSSSSSSSS  MMM      MMM  LL
GG      GG  AA      AA  WW      WW  AA      AA  SS      SS  MMMM      MMMM  LL
GG      AA      AA  WW      WW  AA      AA  SS      MM  MM  MM  MM  LL
GG      AA      AA  WW      WW  AA      AA  SSS      MM  MMMM  MM  LL
GG      AAAAAAAAAA  WW      WW  AAAAAAAAAA  SSSSSSSSSS  MM      MM  MM  LL
GG      GGGGG  AAAAAAAAAAAA  WW  WW  WW  AAAAAAAAAAAA  SSSSSSSSSS  MM      MM  LL
GG      GGGGG  AA      AA  WW  WWW  WW  AA      AA      SSS  MM      MM  LL
GG      GG  AA      AA  WW  WW  WW  WW  AA      AA      SS  MM      MM  LL
GG      GG  AA      AA  WWW  WWW  AA      AA  SS      SS  MM      MM  LL
GGGGGGGGGGGG  AA      AA  WWW  WWW  AA      AA  SSSSSSSSSSSS  MM      MM  LLLLLLLLLLLL
GGGGGGGGGGGG  AA      AA  WW      WW  AA      AA  SSSSSSSSSS  MM      MM  LLLLLLLLLLLL

```

```

JJJJJJJJJJ      444      00000000      444      AAAAAAAAAA
JJJJJJJJJJ      4444      0000000000      4444      AAAAAAAAAAAA
JJ      44 44      00      0000      44 44      AA      AA
JJ      44 44      00      00 00      44 44      AA      AA
JJ      44 44      00      00 00      44 44      AA      AA
JJ      444444444444      00 00 00      444444444444      AAAAAAAAAAAA
JJ      444444444444      00 00 00      444444444444      AAAAAAAAAAAA
JJ      44      00 00      00      44      AA      AA
JJ      JJ      44      0000      00      44      AA      AA
JJ      JJ      44      000      00      44      AA      AA
JJJJJJJJ      44      0000000000      44      AA      AA
JJJJJJ      44      00000000      44      AA      AA

```

```

****A  START  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  START  A****
****A  START  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  START  A****
****A  START  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  START  A****
****A  START  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  START  A****

```

J E S 2 J O B L O G

20.14.46 JOB 404 \$HASP373 GAWASML STARTED - INIT 1 - CLASS A - SYS TK4-
20.14.46 JOB 404 IEF403I GAWASML - STARTED - TIME=20.14.46
20.14.47 JOB 404 IEFACTRT - Stepname Procstep Program Retcode
20.14.47 JOB 404 GAWASML GAW8051 ASM IFOX00 RC= 0000
20.14.47 JOB 404 GAWASML GAW8051 LKED IEWL RC= 0000
20.14.47 JOB 404 IEF404I GAWASML - ENDED - TIME=20.14.47
20.14.47 JOB 404 \$HASP395 GAWASML ENDED

----- JES2 JOB STATISTICS -----

31 JUL 19 JOB EXECUTION DATE

961 CARDS READ

1,171 SYSOUT PRINT RECORDS

0 SYSOUT PUNCH RECORDS

0.01 MINUTES EXECUTION TIME

1	//GAWASML JOB , 'G. WASSINK', MSGCLASS=A, NOTIFY=HERC01,	JOB 404
	// USER=HERC01, PASSWORD= GENERATED BY GDL	
2	//GAW8051 EXEC ASMFCL, PARM.ASM='LOAD,NODECK,LIST,NOXREF'	00000209
3	XXASMFCL PROC MAC='SYS1.MACLIB', MAC1='SYS1.MACLIB',	00000100
	XX MAC2='SYS1.MACLIB', MAC3='SYS1.MACLIB', SOUT='*'	00000200
4	XXASM EXEC PGM=IFOX00, PARM=OBJ, REGION=128K	00000300
5	XKSYSLIB DD DSN=&MAC, DISP=SHR	00000400
6	XX DD DSN=&MAC1, DISP=SHR	00000500
7	XX DD DSN=&MAC2, DISP=SHR	00000600
8	XX DD DSN=&MAC3, DISP=SHR	00000700
9	XKSYSUT1 DD DSN=&&SYSUT1, UNIT=SYSSQ, SPACE=(1700, (600, 100)),	00000800
	XX SEP=(SYSLIB)	00000900
10	XKSYSUT2 DD DSN=&&SYSUT2, UNIT=SYSSQ, SPACE=(1700, (300, 50)),	00001000
	XX SEP=(SYSLIB, SYSUT1)	00001100
11	XKSYSUT3 DD DSN=&&SYSUT3, UNIT=SYSSQ, SPACE=(1700, (300, 50))	00001200
12	XKSYSPRINT DD SYSOUT=&SOUT, DCB=BLKSIZE=1089	00001300
13	XKSYSPUNCH DD SYSOUT=B	00001400
14	XKSYSGO DD DSN=&&OBJSET, UNIT=SYSSQ, SPACE=(80, (200, 50)),	00001500
	XX DISP=(MOD, PASS)	00001600
15	//ASM.SYSIN DD *	00000309
16	XXLKED EXEC PGM=IEWL, PARM=(XREF, LET, LIST, NCAL), REGION=128K,	00001700
	XX COND=(8, LT, ASM)	00001800
17	XKSYSLIN DD DSN=&&OBJSET, DISP=(OLD, DELETE)	00001900
18	XX DD DDNAME=SYSIN	00002000
19	//LKED.SYSLMOD DD DSN=HERC01.TEST.LOADLIB, DISP=OLD	00095709
	X/SYSLMOD DD DSN=&&GOSET(GO), UNIT=SYSDA, SPACE=(1024, (50, 20, 1)),	00002100
	XX DISP=(MOD, PASS)	00002200
20	XKSYSUT1 DD DSN=&&SYSUT1, UNIT=(SYSDA, SEP=(SYSLIN, SYSLMOD)),	00002300
	XX SPACE=(1024, (50, 20))	00002400
21	XKSYSPRINT DD SYSOUT=&SOUT	00002500
22	//LKED.SYSIN DD *	00095809

STMT NO. MESSAGE

5 IEF653I SUBSTITUTION JCL - DSN=SYS1.MACLIB,DISP=SHR
 6 IEF653I SUBSTITUTION JCL - DSN=SYS1.MACLIB,DISP=SHR
 7 IEF653I SUBSTITUTION JCL - DSN=SYS1.MACLIB,DISP=SHR
 8 IEF653I SUBSTITUTION JCL - DSN=SYS1.MACLIB,DISP=SHR
 12 IEF653I SUBSTITUTION JCL - SYSOUT=*,DCB=BLKSIZE=1089
 21 IEF653I SUBSTITUTION JCL - SYSOUT=*

IEF236I ALLOC. FOR GAWASML ASM GAW8051

IEF237I 148 ALLOCATED TO SYSLIB
 IEF237I 148 ALLOCATED TO
 IEF237I 148 ALLOCATED TO
 IEF237I 148 ALLOCATED TO
 IEF237I 190 ALLOCATED TO SYSUT1
 IEF237I 140 ALLOCATED TO SYSUT2
 IEF237I 170 ALLOCATED TO SYSUT3
 IEF237I JES2 ALLOCATED TO SYSPRINT
 IEF237I JES2 ALLOCATED TO SYSPUNCH
 IEF237I 180 ALLOCATED TO SYSGO
 IEF237I JES2 ALLOCATED TO SYSIN

IEF142I GAWASML ASM GAW8051 - STEP WAS EXECUTED - COND CODE 0000

IEF285I SYS1.MACLIB KEPT *-----11
 IEF285I VOL SER NOS= MVSRES.
 IEF285I SYS1.MACLIB KEPT *-----0
 IEF285I VOL SER NOS= MVSRES.
 IEF285I SYS1.MACLIB KEPT *-----0
 IEF285I VOL SER NOS= MVSRES.
 IEF285I SYS1.MACLIB KEPT *-----0
 IEF285I VOL SER NOS= MVSRES.
 IEF285I SYS19212.T201446.RA000.GAWASML.SYSUT1 DELETED *-----80
 IEF285I VOL SER NOS= WORK03.
 IEF285I SYS19212.T201446.RA000.GAWASML.SYSUT2 DELETED *-----28
 IEF285I VOL SER NOS= WORK00.
 IEF285I SYS19212.T201446.RA000.GAWASML.SYSUT3 DELETED *-----16
 IEF285I VOL SER NOS= WORK01.
 IEF285I JES2.JOB00404.S00103 SYSOUT
 IEF285I JES2.JOB00404.S00104 SYSOUT
 IEF285I SYS19212.T201446.RA000.GAWASML.OBJSET PASSED *-----39
 IEF285I VOL SER NOS= WORK02.
 IEF285I JES2.JOB00404.SI0101 SYSIN

IEF373I STEP /ASM / START 19212.2014

IEF374I STEP /ASM / STOP 19212.2014 CPU 0MIN 00.39SEC SRB 0MIN 00.10SEC VIRT 128K SYS 196K

* 1. Jobstep of job: GAWASML Stepname: ASM Program name: IFOX00 Executed on 31.07.19 from 20.14.46 to 20.14.47 *
 * elapsed time 00:00:00,59 CPU-Identifier: TK4- Page-in: 0 *
 * CPU time 00:00:00,49 Virtual Storage used: 128K Page-out: 0 *
 * corr. CPU: 00:00:00,49 CPU time has been corrected by 1 / 1,0 multiplier *
 *

* I/O Operation *

* Number of records read via DD * or DD DATA: 952 *
 * 148.....11 148.....0 148.....0 148.....0 190.....80 140.....28 170.....16 DMY.....0 DMY.....0 180.....39 *
 * DMY.....0 *
 *

Charge for step (w/o SYSOUT): 0,81 *

IEF236I ALLOC. FOR GAWASML LKED GAW8051

IEF237I 180 ALLOCATED TO SYSLIN
 IEF237I JES2 ALLOCATED TO
 IEF237I 241 ALLOCATED TO SYSLMOD
 IEF237I 240 ALLOCATED TO SYS00002
 IEF237I 140 ALLOCATED TO SYSUT1
 IEF237I JES2 ALLOCATED TO SYSPRINT

```

IEF142I GAWASML LKED GAW8051 - STEP WAS EXECUTED - COND CODE 0000
IEF285I  SYS19212.T201446.RA000.GAWASML.OBJSET      DELETED      *-----40
IEF285I  VOL SER NOS= WORK02.
IEF285I  JES2.JOB00404.SI0102                      SYSIN
IEF285I  HERC01.TEST.LOADLIB                       KEPT            *-----9
IEF285I  VOL SER NOS= PUB010.
IEF285I  SYS1.UCAT.TSO                             KEPT            *-----0
IEF285I  VOL SER NOS= PUB000.
IEF285I  SYS19212.T201446.RA000.GAWASML.SYSUT1    DELETED      *-----0
IEF285I  VOL SER NOS= WORK00.
IEF285I  JES2.JOB00404.S00105                      SYSOUT
IEF373I STEP /LKED / START 19212.2014
IEF374I STEP /LKED / STOP 19212.2014 CPU   OMIN 00.07SEC SRB   OMIN 00.03SEC VIRT 128K SYS 208K
*****
* 2. Jobstep of job: GAWASML Stepname: LKED Program name: IEWL Executed on 31.07.19 from 20.14.47 to 20.14.47 *
* elapsed time 00:00:00,20 CPU-Identifier: TK4- Page-in: 0 *
* CPU time 00:00:00,10 Virtual Storage used: 128K Page-out: 0 *
* corr. CPU: 00:00:00,10 CPU time has been corrected by 1 / 1,0 multiplier *
* *
* I/O Operation *
* Number of records read via DD * or DD DATA: 1 *
* 180.....40 DMY.....0 241.....9 240.....0 140.....0 DMY.....0 *
* *
* Charge for step (w/o SYSOUT): 0,16 *
*****
IEF375I JOB /GAWASML / START 19212.2014
IEF376I JOB /GAWASML / STOP 19212.2014 CPU   OMIN 00.46SEC SRB   OMIN 00.13SEC

```

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID
	PC	0001	000000	000000	
GAW8051	SD	0002	000000	002998	

ASM 0201 20.14 07/31/19

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
2	*				*****	0000500
3	*				GAW8051 (DE-)COMPRESS A GIVEN STRING, COMPRESSING STRINGS	0000600
4	*				OF THREE OR MORE CONSECUTIVE EQUAL CHARACTERS	0000700
5	*					0000800
6	*				CALL FROM PL/1: CALL GAW8051 (OPER, RC, STRING)	0000900
7	*				WHERE: OPER : CHAR (4)	0001000
8	*				RC : CHAR (2) WHERE 00 = OK	0001100
9	*				02 = NOT OK	0001200
10	*				99 = OPER NOT OK	0001300
11	*				STRING: FIXED BIN (15) CONTAINS LENGTH	0001400
12	*				CHAR(8192)	0001500
13	*					0001600
14	*				PGMR: GERARD A. WASSINK	0001700
15	*					0001800
16	*					0001908
17	*				PARAMETER HANDLING COURTESY OF JAY MOSELEY	0002008
18	*				ORIGINALLY WRITTEN DECEMBER 2001	0002108
19	*				SEE HTTP://WWW.JAYMOSELEY.COM/HERCULES	0002208
20	*					0002308
21	*				*****	0002400
22	*					0002512
23	*					0002612
24	*					0002712
25	*				-----	0002812
26	*				GNU LICENSE CONDITIONS	0002912
27	*				-----	0003012
28	*				THIS PROGRAM IS FREE SOFTWARE; YOU CAN REDISTRIBUTE IT AND/OR MODIFY	0003112
29	*				IT UNDER THE TERMS OF THE GNU GENERAL PUBLIC LICENSE AS PUBLISHED BY	0003212
30	*				THE FREE SOFTWARE FOUNDATION; EITHER VERSION 2 OF THE LICENSE, OR	0003312
31	*				(AT YOUR OPTION) ANY LATER VERSION.	0003412
32	*					0003512
33	*				THIS PROGRAM IS DISTRIBUTED IN THE HOPE THAT IT WILL BE USEFUL,	0003612
34	*				BUT WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF	0003712
35	*				MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SEE THE	0003812
36	*				GNU GENERAL PUBLIC LICENSE FOR MORE DETAILS.	0003912
37	*					0004012
38	*				YOU SHOULD HAVE RECEIVED A COPY OF THE GNU GENERAL PUBLIC LICENSE	0004112
39	*				ALONG WITH THIS PROGRAM; IF NOT, WRITE TO THE FREE SOFTWARE	0004212
40	*				FOUNDATION, INC.,	0004312
41	*				51 FRANKLIN STREET, FIFTH FLOOR, BOSTON, MA 02110-1301 USA.	0004412
42	*					0004512
43	*				-----	0004612
44	*					0004700
45	*					0004812
46	*					0004912
47	*				***** SYMBOLIC NAMES FOR USED REGISTERS *****	0005000
48	*					0005100
00001	R1			49	EQU 1 PARAMETER LIST ADDRESS ON ENTRY	0005200
00002	R2			50	EQU 2)	0005300
00003	R3			51	EQU 3)	0005400
00004	R4			52	EQU 4)	0005500
00005	R5			53	EQU 5) FREE FOR ROUTINE USAGE	0005600
00006	R6			54	EQU 6)	0005700
00007	R7			55	EQU 7)	0005800
00008	R8			56	EQU 8)	0005900

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 20.14 07/31/19	
			00009	57 R9	EQU	9	BASE ADDR OPERATION / BUFFER	00006000
			0000A	58 R10	EQU	10	BASE ADDR RETURN CODE	00006100
			0000B	59 R11	EQU	11	BASE ADDR STRING / RECORD	00006200
			0000C	60 R12	EQU	12	BASE REGISTER FOR THIS PROGRAM	00006300
			0000D	61 R13	EQU	13	SAVE AREA ADDRESS	00006400
			0000E	62 R14	EQU	14	RETURN ADDRESS	00006500
			0000F	63 R15	EQU	15	INITIAL VALUE FOR BASE REGISTER	00006600

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				65	* -----	00006800
				66	* MAIN LINE OPERATIONS	00006906
				67	* -----	00007000
000000				68	GAW8051 CSECT ,	00007106
000000	90EC D00C	0000C		69	STM R14,R12,12(R13)	00007200
				70	*	00007300
000004	18CF			71	LR R12,R15 ESTABLISH	00007405
		00000		72	USING GAW8051,R12 ADDRESSABILITY	00007500
				73	*	00007600
000006	4120 C8C0	008C0		74	LA R2,SAVEAREA LINK	00007700
00000A	50D0 C8C4	008C4		75	ST R13,SAVEAREA+4 SAVEAREA	00007805
00000E	5020 D008	00008		76	ST R2,8(,R13) TO PREVIOUS	00007906
000012	41D0 C8C0	008C0		77	LA R13,SAVEAREA ADDRESS OUR OWN	00008000
				78	*	00008100
				79	* -----*	00008208
				80	* THREE PARAMETERS SHOULD BE PASSED FROM THE CALLER. THE ADDRESS OF *	00008308
				81	* THE LAST PARAMETER PASSED WILL HAVE ITS HIGH ORDER BIT SET ON. *	00008408
				82	* IF NOT CALLED WITH THREE PARAMETERS, ISSUE MESSAGE AND ABEND. *	00008508
				83	* -----*	00008608
000016	9180 1000	00000		84	TM 0(R1),X'80' ONE PARAMETER PASSED?	00008708
00001A	4710 C02E	0002E		85	BO PERROR IF YES, ERROR	00008808
00001E	9180 1004	00004		86	TM 4(R1),X'80' TWO PARAMETERS PASSED?	00008908
000022	4710 C02E	0002E		87	BO PERROR IF YES, ERROR	00009008
000026	9180 1008	00008		88	TM 8(R1),X'80' THREE PARAMETERS PASSED?	00009108
00002A	4710 C078	00078		89	BO ALLGOOD IF YES, CONTINUE	00009208
				90	*	00009308
00002E				91	PERROR DS OH	00009408
				92	WTO 'GAW8051: INSUFFICIENT PARAMETERS RECEIVED', ROUTCDE=11,DESC=7	00009508
				93+	CNOP 0,4	09800002
000030	4510 C066	00066		94+	BAL 1,IHB0001A BRANCH AROUND MESSAGE	09850002
000034	002D			95+	DC AL2(45) TEXT LENGTH	13200002
000036	8000			96+	DC B'1000000000000000' MCS FLAGS	13250002
000038	C7C1E6F8F0F5F17A			97+	DC C'GAW8051: INSUFFICIENT PARAMETERS RECEIVED'	13350002
000061	0200			98+	DC B'0000001000000000' DESCRIPTOR CODES	13450002
000063	0020			99+	DC B'0000000000100000' ROUTING CODES	13500002
000066				100+	IHB0001A DS OH	15850002
000066	0A23			101+	SVC 35	15950002
				102	ABEND 712,DUMP	00009708
000068				103+	DS OH	00400002
000068	4110 02C8	002C8		104+	LA 1,712 LOAD PARAMETER REG 1	01900002
00006C	4100 0080	00080		105+	LA 0,128(0,0) PICK UP DUMP/STEP/DUMPOPTS YM1995	01800002
000070	8900 0018	00018		106+	SLL 0,24(0) SHIFT TO HIGH ORDER	01850002
000074	1610			107+	OR 1,0 OR IN WITH COMPCODE	01900002
000076	0A0D			108+	SVC 13 LINK TO ABEND ROUTINE	02050002
				109	*	00009808
000078				110	ALLGOOD DS OH	00009908
000078	989B 1000	00000		111	LM R9,R11,0(R1) LOAD PARAMTER ADRESSES IN R9-R11	00010008
00007C	5890 9000	00000		112	L R9,0(,R9) ADDRESS PRMOPER	00010108
		00000		113	USING PRMOPER,R9	00010200
000080	58A0 A000	00000		114	L R10,0(,R10) ADDRESS PRMRCOD	00010308
		00000		115	USING PRMRCOD,R10	00010400
000084	58B0 B000	00000		116	L R11,0(,R11) ADDRESS PRMRECD	00010508
		00000		117	USING PRMRECD,R11	00010600
				118	*	00010700

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 20.14 07/31/19
000088	D201	A000	C980	00000	00980	119	MVC RETCODE,=C'00'	INITIALIZE RETURN CODE 00010800
00008E	47F0	C0A4		000A4		120	B SELECT	ASSUME ALL WILL GO WELL 00010911
						121	*	00011000
000092						122	BACK DS 0H	00011100
000092	58DD	0004		00004		123	L R13,4(R13)	RESTORE R13 00011200
000096	98EC	D00C		0000C		124	LM R14,R12,12(R13)	RESTORE CALLER'S REGISTERS 00011300
00009A	9601	D00F		0000F		125	OI 15(R13),X'01'	00011405
00009E	41F0	0000		00000		126	LA R15,0(0,0)	RETURN CODE ZERO 00011505
0000A2	07FE					127	BR R14	00011600
						128	*	00011700

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				130	* -----	00011900
				131	* SELECT OPERATION	SELECT 00012000
				132	* -----	00012100
0000A4				133	SELECT DS 0H	00012200
				134	*	00012300
				135	* --- CHECK THE SPECIFIED LENGTH ---	00012400
				136	*	00012500
0000A4 1777				137	XR R7,R7 ZERO OUT R7	00012600
0000A6 4870 B000	00000			138	LH R7,RECLN LOAD RECORD LENGTH	00012700
0000AA 5970 C994	00994			139	C R7,BUFMAX GREATER THAN MAXIMUM?	00012800
0000AE 47D0 C0BC	000BC			140	BNH SEL005 NO, CONTINUE	00012900
				141	*	00013000
0000B2 D201 A000 C982	00000 00982			142	MVC RETCODE,=C'02' NOT_OK	00013100
0000B8 47F0 C092	00092			143	B BACK AND GO BACK	00013200
				144	*	00013300
0000BC				145	SEL005 DS 0H	00013400
				146	*	00013500
				147	* --- SAVE THE CONTENTS OF THE SPECIFIED RECORD IN THE BUFFER ---	00013600
				148	*	00013700
0000BC 9069 C0F0	000F0			149	STM R6,R9,SEL005 STORE REGISTERS	00013800
0000C0 4160 C998	00998			150	LA R6,BUFFER PREPARE	00013900
				151	*	R7 CONTAINS LENGTH 00014000
0000C4 418B 0002	00002			152	LA R8,2(R11) AND	00014100
0000C8 1897				153	LR R9,R7 EXECUTE	00014200
0000CA 0E68				154	MVCL R6,R8 MOVE CHARACTERS LONG	00014300
0000CC 9869 C0F0	000F0			155	LM R6,R9,SEL005 RESTORE REGISTERS	00014400
				156	*	00014500
				157	* --- SELECT THE OPERATION TO PERFORM ---	00014600
				158	*	00014700
0000D0 D503 9000 C978	00000 00978			159	CLC OPERAT,=C'COMP' COMPRESS REQUESTED?	00014800
0000D6 4780 C138	00138			160	BE COMPRESS	00014900
0000DA D503 9000 C97C	00000 0097C			161	CLC OPERAT,=C'DECO' DECOMPRESS REQUESTED?	00015000
0000E0 4780 C768	00768			162	BE DECOMP	00015100
				163	*	00015200
				164	DROP R9 NO USE FOR THIS BASE ANYMORE	00015300
				165	*	00015400
0000E4 D201 A000 C984	00000 00984			166	MVC RETCODE,=C'99' NEITHER, RETURN ERROR	00015500
0000EA 47F0 C092	00092			167	B BACK	00015600
				168	*	00015700
0000F0				169	SEL005 DS 18F LOCAL SAVE AREA	00015800
				170	*	00015900

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				172 *	-----	00016100
				173 *	COMPRESSION ROUTINE	COMPRESS 00016200
				174 *	-----	00016300
				175 *		00016400
				176 *	THIS ROUTINE WILL PERFORM COMPRESSION OF THREE TYPES OF FIELDS:	00016500
				177 *		00016600
				178 *	1) SPACE SUPPRESSION	00016700
				179 *	2) REPLICATING NON-SPACE & NON-DIGIT CHARACTERS	00016800
				180 *	3) CONVERTING ZONED DATA TO BCD	00016900
				181 *		00017000
				182 *	TO BE ABLE TO PROCESS ANY RECORD, EVERY FIELD IN THE COMPRESSED	00017100
				183 *	RECORD IS PRECEDED BY A CONTROL CODE. THIS CONTROL CODE IS ONE BYTE	00017200
				184 *	LONG AND IS USED AS FOLLOWS:	00017300
				185 *		00017400
				186 *	BIT: 76543210	00017500
				187 *	CE.....	00017600
				188 *		00017700
				189 *	BIT 7 INDICATES COMPRESSION '1': NEXT AREA CONTAINS COMPRESSED DATA	00017800
				190 *	'0': NEXT AREA CONTAINS UNCOMPRESSED DATA	00017900
				191 *		00018000
				192 *	BIT 6, WHEN '1', INDICATES AN EXTRA LENGTH BYTE IS PROVIDED	00018100
				193 *		00018200
				194 *		00018300
				195 *		00018400
				196 *	WHEN BIT 7 IS '1', THE LAYOUT OF THE REST OF THE CONTROL CODE IS:	00018500
				197 *		00018600
				198 *	BIT: 76543210	00018700
				199 *	..TLLLLL	00018800
				200 *		00018900
				201 *	WHERE TT STANDS FOR COMPRESSION TYPE:	00019000
				202 *	'10' - REPLICATE CHARACTER COMPRESSION	00019100
				203 *	'01' - SPACE SUPPRESSION	00019200
				204 *	'00' - ZONED TO BCD COMPRESSION	00019300
				205 *		00019400
				206 *	AND LLLL SPECIFIES THE LENGTH OF THE COMPRESSED AREA	00019500
				207 *		00019600
				208 *		00019700
				209 *		00019800
				210 *	WHEN BIT 7 IS '0', THE LAYOUT OF THE REST OF THE CONTROL CODE	00019900
				211 *	IS AS FOLLOWS:	00020000
				212 *		00020100
				213 *	BIT 5 TO 0 SPECIFUY THE LENGHT OF THE UN-COMPRESSED AREA	00020200
				214 *		00020300
				215 *		00020400
				216 *	WHEN BIT 6 WAS '1', THE NEXT BYTE CONTAINS AN EXTRA 8 BITS OF	00020500
				217 *	LENGTH INFORMATION; THE TOTAL LENGTH OF THE FIELD IS CALCULATED	00020600
				218 *	BY MULTIPLYING THE LENGTH FORM THE FIRST CONTROL BYTE BY 256 AND	00020700
				219 *	ADDING THE VALUE FROM THE SECOND BYTE.	00020800
				220 *		00020900
				221 *		00021000
				222 *	IN CASE OF COMPRESSION, WHEN THE COMPRESSION TYPE IS '10',	00021100
				223 *	REPLICATING CHARACTERS, THE NEXT (SECOND OR THIRD) BYTE CONTAINS	00021200
				224 *	THE REPLICATING CHARACTER	00021300
				225 *		00021400
				226 *	-----	00021500

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				227 *		00021600
				228 *		00021700
				229 *		00021800
				230 *	-----	00021900
				231 *	COMPRESSION PROCESS:	00022000
				232 *		00022100
				233 *	1) INITIALIZE POINTERS	00022200
				234 *		00022300
				235 *	2) FIND POINTERS TO FIRST FIELD OF EACH COMPRESSION TYPE	00022400
				236 *		00022500
				237 *	3) DETERMINE WHICH TYPE-POINTER IS SMALLEST	00022600
				238 *		00022700
				239 *	4) WHEN APPROPRIATE, TAKE OVER THE AREA PRECEDING THE SMALLEST	00022800
				240 *	POINTER, INDICATING IT'S UNCOMPRESSED DATA	00022900
				241 *		00023000
				242 *	5) PERFORM THE COMPRESSION OF THE SPECIFIED TYPE	00023100
				243 *		00023200
				244 *	6) FIND THE NEXT AREA OF THE SAME TYPE	00023300
				245 *		00023400
				246 *	7) IF NOT END OF RECORD, LOOP AROUND TO 3)	00023500
				247 *		00023600
				248 *	8) GIVE BACK CONTROL TO CALLING PROGRAM	00023700
				249 *		00023800
				250 *	-----	00023900

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
000138				252	COMPRESS DS 0H	00024100
				253	*	00024200
				254	* --- INITIALIZE VARIABLES & REGISTERS	00024300
				255	*	00024400
000138	4190 C998	00998		256	LA R9,BUFFER BUFFER BASE ADDRESS	00024500
00013C	1879			257	LR R7,R9 BUFFER START	00024600
00013E	4860 B000	00000		258	LH R6,RECLEN PLUS RECORD LENGTH	00024700
000142	1A76			259	AR R7,R6 GIVE BUFFER END + 1	00024800
000144	9240 7000	00000		260	MVI 0(R7),C' ' MOVE IN ONE SPACE PAST END	00024900
000148	0670			261	BCTR R7,0 MINUS ONE EQUALS BUFFER END	00025000
00014A	4130 C998	00998		262	LA R3,BUFFER POINTER TO BUFFER	00025100
00014E	1883			263	LR R8,R3 POINTER IN BUFFER	00025200
000150	4150 B002	00002		264	LA R5,RECORD POINTER TO RECORD	00025300
				265	*	00025400
000154	5030 C964	00964		266	ST R3,T1 STORE T1	00025500
000158	5080 C968	00968		267	ST R8,T2 STORE T2	00025600
00015C	5050 C96C	0096C		268	ST R5,T3 STORE T3	00025700
000160	5070 C970	00970		269	ST R7,EOR STORE EOR	00025800
				270	*	00025900
				271	* --- FIND THE FIRST OCCURRENCE OF EACH POSSIBLE COMPRESSION TYPE ---	00026000
				272	*	00026100
000164	45E0 C574	00574		273	BAL R14,NEXTZON FIND NEXT ZONED AREA	00026200
000168	45E0 C638	00638		274	BAL R14,NEXTSPA FIND NEXT SPACE AREA	00026301
00016C	45E0 C6BC	006BC		275	BAL R14,NEXTREP FIND NEXT REPLICATE AREA	00026400
				276	*	00026500
000170				277	CMP010 DS 0H	00026600
000170	D503 C95C C958 0095C 00958			278	CLC NXTSPA,NXTZON SPACES > ZONED?	00026700
000176	4720 C188 00188			279	BH CMP012 YES	00026800
00017A	D503 C95C C960 0095C 00960			280	CLC NXTSPA,NXTREP SPACES > REPL?	00026900
000180	4720 C188 00188			281	BH CMP012 YES	00027000
000184	47F0 C196 00196			282	B CMP100 DO SPACES	00027100
				283	*	00027200
000188				284	CMP012 DS 0H	00027300
000188	D503 C960 C958 00960 00958			285	CLC NXTREP,NXTZON REPL <= ZONED?	00027400
00018E	47D0 C236 00236			286	BNH CMP200 YES, DO REPL'S	00027500
000192	47F0 C300 00300			287	B CMP300 NO, DO ZONED	00027600
				288	*	00027700

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				290	* -----	00027900
				291	* SPACE COMPRESSION	SPACES 00028000
				292	* -----	00028100
000196				293	CMP100 DS 0H	00028200
000196	5880 C95C	0095C		294	L R8,NXTSPA ADDRESS OF NEXT SPACE AREA	00028300
00019A	45E0 C494	00494		295	BAL R14,TAKEOVER TAKE OVER UNCOMPRESSED DATA	00028400
				296	*	00028500
00019E	5830 C964	00964		297	L R3,T1 LOAD CHANGED T1	00028600
0001A2	5850 C96C	0096C		298	L R5,T3 AND T3	00028700
				299	*	00028800
0001A6	5980 C970	00970		300	C R8,EOR PAST EOR?	00028900
0001AA	4720 C434	00434		301	BH CMP900 YES, END	00029000
				302	*	00029100
				303	* --- DETERMINE LENGTH OF SPACE AREA WITH T2 ---	00029200
				304	*	00029300
0001AE				305	CMP102 DS 0H	00029400
0001AE	9540 8000	00000		306	CLI 0(R8),C' ' IS IT A SPACE?	00029501
0001B2	4770 C1C2	001C2		307	BNE CMP105 NO, END OF SPACE AREA	00029600
0001B6	4188 0001	00001		308	LA R8,1(R8) INCREMENT T2	00029700
0001BA	5980 C970	00970		309	C R8,EOR PAST EOR?	00029800
0001BE	47D0 C1AE	001AE		310	BNH CMP102 NO, NEXT	00029900
				311	*	00030000
				312	* --- R8 POINTS TO EOR + 1 OR TO FIRST NON-SPACE CHARACTER ---	00030100
				313	*	00030200
0001C2				314	CMP105 DS 0H	00030300
0001C2	9200 C953	00953		315	MVI CONTROL,X'00' ZERO OUT CONTROL CODE	00030400
0001C6	1848			316	LR R4,R8 LENGTH EQUALS	00030500
0001C8	1B43			317	SR R4,R3 AREA END -/- AREA START	00030600
0001CA	4040 C950	00950		318	STH R4,ARLEN REMEMBER AREA LENGTH	00030700
0001CE	5940 C988	00988		319	C R4,MAXL4 EXTRA LENGTH BYTE NEEDED?	00030800
0001D2	47D0 C206	00206		320	BNH CMP107 NO, FITS IN FOUR BITS	00030900
				321	*	00031000
				322	* --- CREATE EXTRA LENGTH BYTE ---	00031100
				323	*	00031200
0001D6	4240 C954	00954		324	STC R4,CONTROL2 STORE LENGTH IN 2ND CONTROL BYTE	00031300
0001DA	8840 0008	00008		325	SRL R4,8(0) SHIFT LENGTH TO THE RIGHT	00031400
0001DE	4240 C953	00953		326	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00031500
0001E2	940F C953	00953		327	NI CONTROL,X'0F' ZERO OUT HIGH HALF	00031600
0001E6	9680 C953	00953		328	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00031700
0001EA	9640 C953	00953		329	OI CONTROL,EXTLEN INDICATE EXTRA LENGTH BYTE	00031800
0001EE	9610 C953	00953		330	OI CONTROL,SPACES INDICATE SPACE COMPRESSION	00031900
0001F2	D200 5000 C953	00000 00953		331	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00032000
0001F8	D200 5001 C954	00001 00954		332	MVC 1(1,R5),CONTROL2 STORE CONTROL2 IN RECORD	00032100
0001FE	4150 5002	00002		333	LA R5,2(,R5) BUMP UP R5 PAST CONTROL BYTES	00032200
000202	47F0 C220	00220		334	B CMP190 END OF COMPRESSION	00032300
				335	*	00032400
000206				336	CMP107 DS 0H	00032500
000206	4240 C953	00953		337	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00032600
00020A	940F C953	00953		338	NI CONTROL,X'0F' ZERO OUT HIGH HALF	00032700
00020E	9680 C953	00953		339	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00032800
000212	9610 C953	00953		340	OI CONTROL,SPACES INDICATE SPACE COMPRESSION	00032900
000216	D200 5000 C953	00000 00953		341	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00033000
00021C	4150 5001	00001		342	LA R5,1(,R5) BUMP UP R5 PAST CONTROL BYTE	00033100
				343	*	00033200
000220				344	CMP190 DS 0H	00033300

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 20.14 07/31/19	
000220	1838			345	LR	R3,R8	T1 = T2	00033400
				346	*			00033500
000222	5030 C964	00964		347	ST	R3,T1	STORE T1	00033600
000226	5080 C968	00968		348	ST	R8,T2	STORE T2	00033700
00022A	5050 C96C	0096C		349	ST	R5,T3	STORE T3	00033800
				350	*			00033900
00022E	45E0 C638	00638		351	BAL	R14,NEXTSPA	FIND NEXT SPACE AREA	00034001
				352	*			00034100
000232	47F0 C434	00434		353	B	CMP900	BACK TO CALLER	00034200
				354	*			00034300

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				356	* -----	00034500
				357	* REPLICATE DATA COMPRESSION	REPL'S 00034600
				358	* -----	00034700
000236				359	CMP200 DS 0H	00034800
000236	5880 C960	00960		360	L R8,NXTREP ADDRESS OF NEXT REPL AREA	00034900
00023A	45E0 C494	00494		361	BAL R14,TAKEOVER JUST TAKE OVER AS UNCOMPRESSED	00035000
				362	*	00035100
00023E	5830 C964	00964		363	L R3,T1 LOAD CHANGED T1	00035200
000242	5850 C96C	0096C		364	L R5,T3 AND T3	00035300
				365	*	00035400
000246	5980 C970	00970		366	C R8,EOR PAST EOR?	00035500
00024A	4720 C434	00434		367	BH CMP900 YEP, END	00035600
				368	*	00035700
				369	* --- DETERMINE LENGTH OF REPLICATE AREA WITH T2 ---	00035800
				370	*	00035900
00024E	D200 C952 8000	00952 00000		371	MVC CHAR,0(R8) WHAT'S THE CHARACTER?	00036000
000254	4180 8001	00001		372	LA R8,1(,R8) INCREMENT R8	00036100
000258	5980 C970	00970		373	C R8,EOR PAST EOR?	00036200
00025C	4720 C276	00276		374	BH CMP205 YEP, SKIP	00036300
				375	*	00036400
000260				376	CMP202 DS 0H	00036500
000260	D500 C952 8000	00952 00000		377	CLC CHAR,0(R8) STILL SAME CHARACTER?	00036600
000266	4770 C276	00276		378	BNE CMP205 NOPE, END OF REPL AREA	00036700
00026A	4188 0001	00001		379	LA R8,1(R8) INCREMENT T2	00036800
00026E	5980 C970	00970		380	C R8,EOR PAST EOR?	00036900
000272	47D0 C260	00260		381	BNH CMP202 NO, REPEAT	00037000
				382	*	00037100
000276				383	CMP205 DS 0H	00037200
000276	9200 C953	00953		384	MVI CONTROL,X'00' ZERO OUT CONTROL CODE	00037300
00027A	1848			385	LR R4,R8 LENGTH EQUALS	00037400
00027C	1B43			386	SR R4,R3 AREA END -/- AREA START	00037500
00027E	4040 C950	00950		387	STH R4,ARLEN REMEMBER AREA LENGTH	00037600
000282	5940 C988	00988		388	C R4,MAXL4 EXTRA LENGTH BYTE NEEDED?	00037700
000286	47D0 C2C0	002C0		389	BNH CMP207 NO, FITS IN FOUR BITS	00037800
				390	*	00037900
				391	* --- CREATE EXTRA LENGTH BYTE ---	00038000
				392	*	00038100
00028A	4240 C954	00954		393	STC R4,CONTROL2 STORE LENGTH IN 2ND CONTROL BYTE	00038200
00028E	8840 0008	00008		394	SRL R4,8(0) SHIFT LENGTH TO THE RIGHT	00038300
000292	4240 C953	00953		395	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00038400
000296	940F C953	00953		396	NI CONTROL,X'0F' ZERO OUT HIGH HALF	00038500
00029A	9680 C953	00953		397	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00038600
00029E	9640 C953	00953		398	OI CONTROL,EXTLEN INDICATE EXTRA LENGTH BYTE	00038700
0002A2	9620 C953	00953		399	OI CONTROL,REPLIC INDICATE REPL CHAR COMPRESSION	00038800
0002A6	D200 5000 C953	00000 00953		400	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00038900
0002AC	D200 5001 C954	00001 00954		401	MVC 1(1,R5),CONTROL2 STORE CONTROL2 IN RECORD	00039000
0002B2	D200 5002 C952	00002 00952		402	MVC 2(1,R5),CHAR STORE REPLICATING CHARACTER	00039100
0002B8	4150 5003	00003		403	LA R5,3(,R5) BUMP UP R5 PAST CONTROL BYTES	00039200
0002BC	47F0 C2DC	002DC		404	B CMP290 END OF COMPRESSION	00039300
				405	*	00039400
0002C0				406	CMP207 DS 0H	00039500
0002C0	4240 C953	00953		407	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00039600
0002C4	9680 C953	00953		408	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00039700
0002C8	9620 C953	00953		409	OI CONTROL,REPLIC INDICATE REPL CHAR COMPRESSION	00039800
0002CC	D200 5000 C953	00000 00953		410	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00039900

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 20.14 07/31/19
0002D2	D200	5001 C952	00001	00952	411	MVC	1(1,R5),CHAR	STORE REPLICATING CHARACTER 00040000
0002D8	4150	5002	00002		412	LA	R5,2(,R5)	BUMP UP R5 PAST CONTROL BYTE 00040100
					413	*		00040200
0002DC					414	CMP290	DS 0H	00040300
0002DC	1838				415	LR	R3,R8	T1 = T2 00040400
					416	*		00040500
0002DE	5030	C964	00964		417	ST	R3,T1	STORE T1 00040600
0002E2	5080	C968	00968		418	ST	R8,T2	STORE T2 00040700
0002E6	5050	C96C	0096C		419	ST	R5,T3	STORE T3 00040800
					420	*		00040900
0002EA	D503	C960 C958	00960	00958	421	CLC	NXTREP,NXTZON	REPL'S EQUAL ZONED? 00041000
0002F0	4770	C2F8	002F8		422	BNE	CMP292	NO, SKIP 00041100
0002F4	45E0	C574	00574		423	BAL	R14,NEXTZON	YES, FIND NEXT ZONED AREA 00041200
					424	*		00041300
0002F8					425	CMP292	DS 0H	00041400
0002F8	45E0	C6BC	006BC		426	BAL	R14,NEXTREP	FIND NEXT REPL AREA 00041501
					427	*		00041600
0002FC	47F0	C434	00434		428	B	CMP900	BACK TO CALLER 00041700
					429	*		00041800

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				431 *	-----	00042000
				432 *	ZONED DATA COMPRESSION	ZONED 00042100
				433 *	-----	00042200
000300				434	CMP300 DS 0H	00042300
000300	5880 C958	00958		435	L R8,NXTZON ADDRESS OF NEXT ZONED AREA	00042400
000304	45E0 C494	00494		436	BAL R14,TAKEOVER JUST TAKE OVER AS UNCOMPRESSED	00042500
				437 *		00042600
000308	5830 C964	00964		438	L R3,T1 LOAD CHANGED T1	00042700
00030C	5850 C96C	0096C		439	L R5,T3 AND T3	00042800
				440 *		00042900
000310	5980 C970	00970		441	C R8,EOR PAST EOR?	00043000
000314	4720 C434	00434		442	BH CMP900 YEP, END	00043100
				443 *		00043200
				444 *	--- DETERMINE LENGTH OF REPLICATE AREA WITH T2 ---	00043300
				445 *		00043400
000318				446	CMP302 DS 0H	00043500
000318	95F0 8000	00000		447	CLI 0(R8),C'0' LESS THAN '0' ?	00043600
00031C	4740 C334	00334		448	BL CMP305 END OF ZONED AREA	00043700
000320	95F9 8000	00000		449	CLI 0(R8),C'9' GREATER THAN '9' ?	00043800
000324	4720 C334	00334		450	BH CMP305 END OF ZONED AREA	00043900
000328	4188 0001	00001		451	LA R8,1(R8) INCREMENT T2	00044000
00032C	5980 C970	00970		452	C R8,EOR PAST EOR?	00044100
000330	47D0 C318	00318		453	BNH CMP302 NO, REPEAT	00044200
				454 *		00044300
				455 *	--- R8 POINTS TO EOR + 1 OR TO FIRST NON-ZONED CHARACTER	00044400
				456 *		00044500
000334				457	CMP305 DS 0H	00044600
000334	9200 C953	00953		458	MVI CONTROL,X'00' ZERO OUT CONTROL CODE	00044700
000338	1848			459	LR R4,R8 LENGTH EQUALS	00044800
00033A	1B43			460	SR R4,R3 AREA END -/- AREA START	00044900
00033C	5940 C988	00988		461	C R4,MAXL4 EXTRA LENGTH BYTE NEEDED?	00045000
000340	47D0 C374	00374		462	BNH CMP307 NO, FITS IN FOUR BITS	00045100
				463 *		00045200
				464 *	--- CREATE EXTRA LENGTH BYTE ---	00045300
				465 *		00045400
000344	4240 C954	00954		466	STC R4,CONTROL2 STORE LENGTH IN 2ND CONTROL BYTE	00045500
000348	8840 0008	00008		467	SRL R4,8(0) SHIFT LENGTH TO THE RIGHT	00045600
00034C	4240 C953	00953		468	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00045700
000350	940F C953	00953		469	NI CONTROL,X'0F' ZERO OUT HIGH HALF	00045800
000354	9680 C953	00953		470	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00045900
000358	9640 C953	00953		471	OI CONTROL,EXTLEN INDICATE EXTRA LENGTH BYTE	00046000
00035C	9600 C953	00953		472	OI CONTROL,ZONED INDICATE ZONED CHAR COMPRESSION	00046100
000360	D200 5000 C953	00000 00953		473	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00046200
000366	D200 5001 C954	00001 00954		474	MVC 1(1,R5),CONTROL2 STORE CONTROL2 IN RECORD	00046300
00036C	4150 5002	00002		475	LA R5,2(,R5) BUMP UP R5 PAST CONTROL BYTES	00046400
000370	47F0 C38A	0038A		476	B CMP310 DO COMPRESSION	00046500
				477 *		00046600
000374				478	CMP307 DS 0H	00046700
000374	4240 C953	00953		479	STC R4,CONTROL STORE LENGTH IN 1ST CONTROL BYTE	00046800
000378	9680 C953	00953		480	OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE	00046900
00037C	9600 C953	00953		481	OI CONTROL,ZONED INDICATE ZONED CHAR COMPRESSION	00047000
000380	D200 5000 C953	00000 00953		482	MVC 0(1,R5),CONTROL STORE CONTROL IN RECORD	00047100
000386	4150 5001	00001		483	LA R5,1(,R5) BUMP UP R5 PAST CONTROL BYTE	00047200
				484 *		00047300
00038A				485	CMP310 DS 0H	00047400

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19	
00038A	1863			486	LR R6,R3	WALK THROUGH WITH R6	00047500
				487	*		00047600
				488	* --- MAKE IT BCD ---		00047700
				489	*		00047800
00038C				490	CMP320 DS 0H		00047900
				491	*		00048000
				492	* --- GET FIRST HALF CHARACTER ---		00048100
				493	*		00048200
00038C	1722			494	XR R2,R2	ZERO OUT R2	00048300
00038E	D200 C952 6000 00952 00000			495	MVC CHAR,0(R6)	GET CHAR FROM BUFFER	00048400
000394	940F C952 00952			496	NI CHAR,X'0F'	THROW AWAY ZONE HALF	00048500
000398	4320 C952 00952			497	IC R2,CHAR	LOAD DIGIT IN R2	00048600
00039C	8920 0004 00004			498	SLL R2,4(0)	SHIFT LEFT 4	00048700
0003A0	4220 5000 00000			499	STC R2,0(,R5)	STORE CHARACTER IN RECORD	00048800
0003A4	4150 5001 00001			500	LA R5,1(,R5)	BUMP UP R5	00048900
				501	*		00049000
0003A8	4160 6001 00001			502	LA R6,1(,R6)	INCREMENT R6	00049100
0003AC	1968			503	CR R6,R8	R6 >= R8 ?	00049200
0003AE	47B0 C3D6 003D6			504	BNL CMP390	YES, SKIP	00049300
				505	*		00049400
				506	* --- IF APPROPRIATE, GET SECOND HALF OF CHARACTER ---		00049500
				507	*		00049600
0003B2	D200 C952 6000 00952 00000			508	MVC CHAR,0(R6)	GET CHAR FROM BUFFER	00049700
0003B8	940F C952 00952			509	NI CHAR,X'0F'	THROW AWAY ZONE HALF	00049800
0003BC	4370 C952 00952			510	IC R7,CHAR	LOAD DIGIT IN R2	00049900
0003C0	1627			511	OR R2,R7	OR'EM TOGETHER	00050000
0003C2	0650			512	BCTR R5,0	TEMP DECR R5	00050100
0003C4	4220 5000 00000			513	STC R2,0(,R5)	STORE IT AGAIN	00050200
0003C8	4150 5001 00001			514	LA R5,1(,R5)	BUMP R5 BACK UP	00050300
				515	*		00050400
0003CC	4160 6001 00001			516	LA R6,1(,R6)	INCREMENT R6	00050501
0003D0	1968			517	CR R6,R8	R6 < R8 ?	00050600
0003D2	4740 C38C 0038C			518	BL CMP320	YES, REPEAT	00050700
				519	*		00050800
				520	* --- END OF ZONED COMPRESSION ---		00050900
				521	*		00051000
0003D6				522	CMP390 DS 0H		00051100
0003D6	1838			523	LR R3,R8	T1 = T2	00051200
				524	*		00051300
0003D8	5030 C964 00964			525	ST R3,T1	STORE T1	00051400
0003DC	5080 C968 00968			526	ST R8,T2	STORE T2	00051500
0003E0	5050 C96C 0096C			527	ST R5,T3	STORE T3	00051600
				528	*		00051700
0003E4	45E0 C574 00574			529	BAL R14,NEXTZON	YES, FIND NEXT ZONED AREA	00051800
				530	*		00051900
0003E8	47F0 C434 00434			531	B CMP900	BACK TO CALLER	00052000
				532	*		00052100
0003EC				533	CMP3SAV DS 18F	LOCAL SAVE AREA	00052200

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				535 *	-----	00052400
				536 *	COMPRESSION ROUTINE ENDS	CMP900 00052500
				537 *	-----	00052600
000434				538	CMP900 DS 0H	00052700
000434	5930 C970	00970		539	C R3,EOR PAST END ?	00052800
000438	47D0 C170	00170		540	BNH CMP010 NO, AGAIN	00052900
				541 *		00053000
00043C	4130 B002	00002		542	LA R3,RECORD DETERMINE	00053100
000440	1B53			543	SR R5,R3 NEW LENGTH	00053200
000442	4050 B000	00000		544	STH R5,RECLN AND STORE IT	00053300
				545 *		00053400
000446	47F0 C092	00092		546	B BACK BACK TO CALLER	00053500
				547 *		00053600
00044C				548	CMPSAV DS 18F LOCAL SAVE AREA	00053700

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				550	* -----	00053900
				551	* TAKE OVER AN AREA OF UNCOMPRESSED CHARACTERS	TAKEOVER 00054000
				552	* -----	00054100
000494				553	TAKEOVER DS 0H	00054200
000494	90EC D00C	0000C		554	STM R14,R12,12(R13)	STORE REGISTERS 00054300
000498	50D0 C530	00530		555	ST R13,TKOSAV+4	INCL R13 00054400
00049C	41D0 C52C	0052C		556	LA R13,TKOSAV	ADDRESS LOCAL SAVEAREA 00054500
				557	*	00054600
0004A0	5830 C964	00964		558	L R3,T1	LOAD T1 00054700
0004A4	5850 C96C	0096C		559	L R5,T3	LOAD T3 00054800
				560	*	00054900
0004A8	1938			561	CR R3,R8	UNCOMPRESSED AREA TO PROCESS ? 00055000
0004AA	4780 C520	00520		562	BE TKO900	NOPE, SKIP 00055100
				563	*	00055200
0004AE	9200 C953	00953		564	MVI CONTROL,X'00'	ZERO OUT CONTROL CODE 00055300
0004B2	1848			565	LR R4,R8	LENGTH EQUALS 00055400
0004B4	1B43			566	SR R4,R3	AREA END -/- AREA START 00055500
0004B6	4040 C950	00950		567	STH R4,ARLEN	REMEMBER AREA LENGTH 00055600
0004BA	5940 C98C	0098C		568	C R4,MAXL6	EXTRA LENGTH BYTE NEEDED? 00055700
0004BE	47D0 C4EA	004EA		569	BNH TKO050	NO, FITS IN SIX BITS 00055800
				570	*	00055900
				571	* --- CREATE EXTRA LENGTH BYTE ---	00056000
				572	*	00056100
0004C2	4240 C954	00954		573	STC R4,CONTROL2	STORE LENGTH IN 2ND CONTROL BYTE 00056200
0004C6	8840 0008	00008		574	SRL R4,8(0)	SHIFT LENGTH TO THE RIGHT 00056300
0004CA	4240 C953	00953		575	STC R4,CONTROL	STORE LENGTH IN 1ST CONTROL BYTE 00056400
0004CE	943F C953	00953		576	NI CONTROL,X'3F'	ZERO OUT HIGH 2 BITS 00056500
0004D2	9640 C953	00953		577	OI CONTROL,EXTLEN	INDICATE EXTRA LENGTH BYTE 00056600
0004D6	D200 5000 C953	00000 00953		578	MVC 0(1,R5),CONTROL	STORE CONTROL IN RECORD 00056700
0004DC	D200 5001 C954	00001 00954		579	MVC 1(1,R5),CONTROL2	STORE CONTROL2 IN RECORD 00056800
0004E2	4150 5002	00002		580	LA R5,2(,R5)	BUMP UP R5 PAST CONTROL BYTES 00056900
0004E6	47F0 C4FC	004FC		581	B TKO100	DO COMPRESSION 00057000
				582	*	00057100
0004EA				583	TKO050 DS 0H	00057200
0004EA	4240 C953	00953		584	STC R4,CONTROL	STORE LENGTH IN 1ST CONTROL BYTE 00057300
0004EE	943F C953	00953		585	NI CONTROL,X'3F'	ZERO OUT HIGH 2 BITS 00057400
0004F2	D200 5000 C953	00000 00953		586	MVC 0(1,R5),CONTROL	STORE CONTROL IN RECORD 00057500
0004F8	4150 5001	00001		587	LA R5,1(,R5)	BUMP UP R5 PAST CONTROL BYTE 00057600
				588	*	00057700
0004FC				589	TKO100 DS 0H	00057800
0004FC	9069 C908	00908		590	STM R6,R9,TEMPSAV	SAVE REGISTERS 00057900
000500	1865			591	LR R6,R5	PREPARE MVCL, DEST ADDRESS 00058000
000502	4870 C950	00950		592	LH R7,ARLEN	LOAD LENGTH 00058100
000506	1847			593	LR R4,R7	SAVE LENGTH 00058200
000508	1883			594	LR R8,R3	SOURCE ADDRESS 00058300
00050A	1897			595	LR R9,R7	LOAD LENGTH 00058400
00050C	0E68			596	MVCL R6,R8	MOVE AREA FROM BUFFER TO RECORD 00058500
00050E	9869 C908	00908		597	LM R6,R9,TEMPSAV	RESTORE REGISTERS 00058600
				598	*	00058700
000512	1838			599	LR R3,R8	T1 = T2 00058800
000514	4154 5000	00000		600	LA R5,0(R4,R5)	T3 = T3 + LENGTH 00058900
				601	*	00059000
000518	5030 C964	00964		602	ST R3,T1	STORE T1 00059100
00051C	5050 C96C	0096C		603	ST R5,T3	STORE T3 00059200
				604	*	00059300

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 20.14 07/31/19
000520				605	TKO900	DS 0H	00059400
000520	58D0 C530	00530		606		L R13,TKOSAV+4 RESTORE R13	00059500
000524	98EC D00C	0000C		607		LM R14,R12,12(R13) RESTORE REGISTERS	00059600
000528	07FE			608		BR R14 BACK TO CALLER	00059700
				609	*		00059800
00052C				610	TKOSAV	DS 18F LOCAL SAVE AREA	00059900
				611	*		00060000

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				613	* -----	00060200
				614	* FIND NEXT AREA WITH FOUR OR MORE ZONED CHARACTERS	NEXTZON 00060300
				615	* -----	00060400
000574				616	NEXTZON DS OH	00060500
000574	90EC D00C	0000C		617	STM R14,R12,12(R13) STORE REGISTERS	00060600
000578	50D0 C5F4	005F4		618	ST R13,NXTZSAV+4 INCL R13	00060700
00057C	41D0 C5F0	005F0		619	LA R13,NXTZSAV ADDRESS LOCAL SAVEAREA	00060800
				620	*	00060900
000580	5840 C970	00970		621	L R4,EOR BUFFER END ADDRESS	00061000
000584	0640			622	BCTR R4,0 MINUS THREE	00061100
000586	0640			623	BCTR R4,0 EQUALS	00061200
000588	0640			624	BCTR R4,0 LIMIT	00061300
				625	*	00061400
00058A				626	NZO100 DS OH	00061500
00058A	95F0 8000	00000		627	CLI 0(R8),C'0' ZONED CHARACTER ?	00061600
00058E	4740 C5CE	005CE		628	BL NZO110 NOPE, NEXT	00061700
000592	95F9 8000	00000		629	CLI 0(R8),C'9' ZONED CHARACTER ?	00061800
000596	4720 C5CE	005CE		630	BH NZO110 NOPE, NEXT	00061900
				631	*	00062000
00059A	95F0 8001	00001		632	CLI 1(R8),C'0' ZONED CHARACTER ?	00062100
00059E	4740 C5CE	005CE		633	BL NZO110 NOPE, NEXT	00062200
0005A2	95F9 8001	00001		634	CLI 1(R8),C'9' ZONED CHARACTER ?	00062300
0005A6	4720 C5CE	005CE		635	BH NZO110 NOPE, NEXT	00062400
				636	*	00062500
0005AA	95F0 8002	00002		637	CLI 2(R8),C'0' ZONED CHARACTER ?	00062600
0005AE	4740 C5CE	005CE		638	BL NZO110 NOPE, NEXT	00062700
0005B2	95F9 8002	00002		639	CLI 2(R8),C'9' ZONED CHARACTER ?	00062800
0005B6	4720 C5CE	005CE		640	BH NZO110 NOPE, NEXT	00062900
				641	*	00063000
0005BA	95F0 8003	00003		642	CLI 3(R8),C'0' ZONED CHARACTER ?	00063100
0005BE	4740 C5CE	005CE		643	BL NZO110 NOPE, NEXT	00063200
0005C2	95F9 8003	00003		644	CLI 3(R8),C'9' ZONED CHARACTER ?	00063300
0005C6	4720 C5CE	005CE		645	BH NZO110 NOPE, NEXT	00063400
				646	*	00063500
				647	* --- FOUR ZONED CHARACTERS FOUND ---	00063600
				648	*	00063700
0005CA	47F0 C5E0	005E0		649	B NZO900 YEP, GO BACK	00063800
				650	*	00063900
0005CE				651	NZO110 DS OH	00064000
0005CE	4188 0001	00001		652	LA R8,1(R8) INCREMENT R8	00064100
0005D2	1984			653	CR R8,R4 LIMIT REACHED?	00064200
0005D4	47D0 C58A	0058A		654	BNH NZO100 NO, AGAIN	00064300
				655	*	00064400
0005D8	5880 C970	00970		656	L R8,EOR POINT PAST	00064500
0005DC	4188 0001	00001		657	LA R8,1(R8) BUFFER END	00064600
				658	*	00064700
0005E0				659	NZO900 DS OH	00064800
0005E0	5080 C958	00958		660	ST R8,NXTZON REMEMBER NEXT ZONED AREA	00064900
				661	*	00065000
0005E4	58D0 C5F4	005F4		662	L R13,NXTZSAV+4 RESTORE R13	00065100
0005E8	98EC D00C	0000C		663	LM R14,R12,12(R13) RESTORE REGISTERS	00065200
0005EC	07FE			664	BR R14 BACK TO CALLER	00065300
				665	*	00065400
0005F0				666	NXTZSAV DS 18F LOCAL SAVE AREA	00065500

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				668 *	-----	00065700
				669 *	FIND NEXT AREA WITH TWO OR MORE SPACES	NEXTSPA 00065800
				670 *	-----	00065900
000638				671	NEXTSPA DS 0H	00066000
000638	90EC D00C	0000C		672	STM R14,R12,12(R13) STORE REGISTERS	00066100
00063C	50D0 C678	00678		673	ST R13,NXTSSAV+4 INCL R13	00066200
000640	41D0 C674	00674		674	LA R13,NXTSSAV ADDRESS LOCAL SAVEAREA	00066300
				675 *		00066400
000644	5840 C970	00970		676	L R4,EOR BUFFER END ADDRESS	00066500
000648	0640			677	BCTR R4,0 MINUS 1 IS LIMIT	00066600
				678 *		00066700
00064A				679	NSP001 DS 0H	00066800
00064A	D501 8000 C986	00000 00986		680	CLC 0(2,R8),=C' ' TWO SPACES ?	00066901
000650	4780 C666	00666		681	BE NSP800 YEP, FOUND	00067000
000654	4188 0001	00001		682	LA R8,1(R8) INCREMENT R8	00067100
000658	1984			683	CR R8,R4 LIMIT REACHED ?	00067200
00065A	47D0 C64A	0064A		684	BNH NSP001 NOPE, REPEAT	00067300
				685 *		00067400
00065E	5880 C970	00970		686	L R8,EOR YEP, POINT PAST	00067500
000662	4188 0001	00001		687	LA R8,1(R8) BUFFER END	00067600
				688 *		00067700
000666				689	NSP800 DS 0H	00067800
000666	5080 C95C	0095C		690	ST R8,NXTSPA REMEMBER NEXT SPACE AREA	00067900
				691 *		00068000
00066A				692	NSP900 DS 0H	00068100
00066A	58D0 C678	00678		693	L R13,NXTSSAV+4 RESTORE R13	00068200
00066E	98EC D00C	0000C		694	LM R14,R12,12(R13) RESTORE REGISTERS	00068300
000672	07FE			695	BR R14 BACK TO CALLER	00068400
				696 *		00068500
000674				697	NXTSSAV DS 18F LOCAL SAVE AREA	00068600

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				699 *	-----	00068800
				700 *	FIND NEXT AREA WITH THREE OR MORE REPLICATING CHARACTERS	NEXTREP 00068900
				701 *	-----	00069000
0006BC				702	NEXTREP DS OH	00069100
0006BC	90EC D00C	0000C		703	STM R14,R12,12(R13)	STORE REGISTERS 00069200
0006C0	50D0 C724	00724		704	ST R13,NXTRSAV+4	INCL R13 00069300
0006C4	41D0 C720	00720		705	LA R13,NXTRSAV	ADDRESS LOCAL SAVEAREA 00069400
				706 *		00069500
0006C8	5840 C970	00970		707	L R4,EOR	BUFFER END ADDRESS 00069600
0006CC	0640			708	BCTR R4,0	MINUS 2 00069700
0006CE	0640			709	BCTR R4,0	IS LIMIT 00069800
				710 *		00069900
0006D0				711	NRP100 DS OH	00070000
0006D0	9540 8000	00000		712	CLI 0(R8),C' '	SPACE ? 00070100
0006D4	4780 C700	00700		713	BE NRP105	YEP, NEXT 00070200
				714 *		00070300
0006D8	95F0 8000	00000		715	CLI 0(R8),C' 0'	DIGIT ? 00070400
0006DC	4740 C6E8	006E8		716	BL NRP102	NO, PERHAPS OK 00070500
				717 *		00070600
0006E0	95F9 8000	00000		718	CLI 0(R8),C' 9'	DIGIT ? 00070700
0006E4	47D0 C700	00700		719	BNH NRP105	YES, NEXT 00070800
				720 *		00070900
0006E8				721	NRP102 DS OH	00071000
0006E8	D500 8000 8001	00000 00001		722	CLC 0(1,R8),1(R8)	SAME CHARACTER ? 00071100
0006EE	4770 C700	00700		723	BNE NRP105	NOPE, NEXT 00071200
0006F2	D500 8001 8002	00001 00002		724	CLC 1(1,R8),2(R8)	SAME CHARACTER ? 00071300
0006F8	4770 C700	00700		725	BNE NRP105	NOPE, NEXT 00071400
0006FC	47F0 C712	00712		726	B NRP900	YEP, FOUND 00071500
				727 *		00071600
000700				728	NRP105 DS OH	00071700
000700	4188 0001	00001		729	LA R8,1(R8)	INCREMENT R8 00071800
000704	1984			730	CR R8,R4	LIMIT REACHED ? 00071900
000706	47D0 C6D0	006D0		731	BNH NRP100	NOPE, REPEAT 00072000
				732 *		00072100
00070A	5880 C970	00970		733	L R8,EOR	YEP, POINT PAST 00072200
00070E	4188 0001	00001		734	LA R8,1(R8)	BUFFER END 00072300
				735 *		00072400
000712				736	NRP900 DS OH	00072500
000712	5080 C960	00960		737	ST R8,NXTREP	REMEMBER NEXT SPACE AREA 00072600
				738 *		00072700
000716	58D0 C724	00724		739	L R13,NXTRSAV+4	RESTORE R13 00072800
00071A	98EC D00C	0000C		740	LM R14,R12,12(R13)	RESTORE REGISTERS 00072900
00071E	07FE			741	BR R14	BACK TO CALLER 00073000
				742 *		00073100
000720				743	NXTRSAV DS 18F	LOCAL SAVE AREA 00073200

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				745 *	-----	00073400
				746 *	DE-COMPRESSION ROUTINE	DECOMP 00073500
				747 *	-----	00073600
000768				748	DECOMP DS 0H	00073700
				749 *		00073800
				750 *	--- INITIALIZE REGISTERS & VARIABLES ---	00073900
				751 *		00074000
000768	4190 C998	00998		752	LA R9,BUFFER	BUFFER BASE ADDRESS 00074100
00076C	1879			753	LR R7,R9	BUFFER START 00074200
00076E	4860 B000	00000		754	LH R6,RECLN	PLUS RECORD LENGTH 00074300
000772	1A76			755	AR R7,R6	EQUALS BUFFER END PLUS 1 00074400
000774	9240 7000	00000		756	MVI 0(R7),C' '	MOVE SPACE AFTER LAST POSITION 00074500
000778	0670			757	BCTR R7,0	MINUS ONE EQUALS BUFFER END 00074600
00077A	5070 C970	00970		758	ST R7,EOR	STORE EOR 00074700
00077E	4180 C998	00998		759	LA R8,BUFFER	POINTER IN BUFFER (T1) 00074800
000782	4150 B002	00002		760	LA R5,RECORD	POINTER IN RECORD (T3) 00074900
				761 *		00075000
				762 *	--- GET FIRST / NEXT CONTROL CODE ---	00075100
				763 *		00075200
000786				764	DEC010 DS 0H	00075300
000786	D200 C953 8000 00953 00000			765	MVC CONTROL,0(R8)	(T2) CONTAINS CONTROL CODE 00075400
00078C	D200 C952 8000 00952 00000			766	MVC CHAR,0(R8)	WORKING CHARACTER 00075500
000792	4188 0001 00001			767	LA R8,1(R8)	INCREMENT R8 00075600
000796	1777			768	XR R7,R7	ZERO OUT R7 00075700
				769 *		00075800
000798	9180 C953 00953			770	TM CONTROL,PACKED	IS IT A PACKED FIELD ? 00075900
00079C	4780 C872 00872			771	BZ DEC800	NO, GO TAKE OVER UNCOMPRESSED 00076000
				772 *		00076100
				773 *	--- IT'S COMPRESSED, LOAD LENGTH IN R7 ---	00076200
				774 *		00076300
0007A0	940F C952 00952			775	NI CHAR,LENGTH4	ZERO OUT HIGH BITS 00076400
0007A4	4370 C952 00952			776	IC R7,CHAR	INSERT LENGTH IN R7 00076500
				777 *		00076600
0007A8	9140 C953 00953			778	TM CONTROL,EXTLEN	EXTRA LENGTH BYTE ? 00076700
0007AC	4780 C7BC 007BC			779	BZ DEC030	NO, R7 OK 00076800
				780 *		00076900
0007B0	8970 0008 00008			781	SLL R7,8(0)	SHIFT LEFT ONE BYTE 00077000
0007B4	4378 0000 00000			782	IC R7,0(R8)	R7 NOW CONTAINS LENGTH 00077100
0007B8	4188 0001 00001			783	LA R8,1(R8)	INCREMENT R8 00077200
				784 *		00077300
				785 *	--- DETERMINE COMPRESSION TYPE ---	00077400
				786 *		00077500
0007BC				787	DEC030 DS 0H	00077600
0007BC	D200 C952 C953 00952 00953			788	MVC CHAR,CONTROL	DUPLICATE CONTROL CODE 00077700
0007C2	9430 C952 00952			789	NI CHAR,TYPCOD	ISOLATE COMPRESSION TYPE 00077800
				790 *		00077900
0007C6	9520 C952 00952			791	CLI CHAR,REPLIC	REPL DATA ? 00078000
0007CA	4780 C7E8 007E8			792	BE DEC100	YEP, GO DO REPL 00078100
				793 *		00078200
0007CE	9510 C952 00952			794	CLI CHAR,SPACES	SPACE DATA ? 00078300
0007D2	4780 C7F6 007F6			795	BE DEC200	YEP, GO DO SPACES 00078400
				796 *		00078500
0007D6	9500 C952 00952			797	CLI CHAR,ZONED	ZONED DATA ? 00078600
0007DA	4780 C816 00816			798	BE DEC300	YEP, GO DO ZONED 00078700
				799 *		00078800

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
0007DE	D201 A000 C982	00000	00982	800	MVC RETCODE,=C'02'	INDICATE ERROR 00078900
0007E4	47F0 C092	00092		801	B BACK	AND BACK TO CALLER 00079000
				802 *		00079100
				803 *	--- REPLICATE DATA DE-COMPRESSION ---	00079200
				804 *		00079300
0007E8				805	DEC100 DS 0H	00079400
0007E8	D200 C952 8000	00952	00000	806	MVC CHAR,0(R8)	GET CAHARACTER FROM BUFFER 00079500
0007EE	4188 0001	00001		807	LA R8,1(R8)	BUMP UP R8 00079600
0007F2	47F0 C7FA	007FA		808	B DEC210	00079700
				809 *		00079800
				810 *	--- SPACE DECOMPRESSION ---	00079900
				811 *		00080000
0007F6				812	DEC200 DS 0H	00080100
0007F6	9240 C952	00952		813	MVI CHAR,C' '	CHARACTYER IS SPACE 00080200
				814 *		00080300
0007FA				815	DEC210 DS 0H	00080400
0007FA	9069 C908	00908		816	STM R6,R9,TEMPSAV	SAVE REGISTERS 00080500
0007FE	1865			817	LR R6,R5	PREPARE MVCL, DEST ADDRESS 00080600
				818 *		00080700
000800	1847			819	LR R4,R7	SAVE LENGTH 00080800
				820 *		00080900
000802	1799			821	XR R9,R9	ZERO OUT SECOND LENGTH REGISTER 00081000
000804	BF98 C952	00952		822	ICM R9,8,CHAR	AND INSERT PADDING BYTE 00081100
000808	0E68			823	MVCL R6,R8	EXECUTE MOVE LONG 00081200
00080A	9869 C908	00908		824	LM R6,R9,TEMPSAV	RESTORE REGISTERS 00081300
				825 *		00081400
00080E	4154 5000	00000		826	LA R5,0(R4,R5)	T3 = T3 + LENGTH 00081500
				827 *		00081600
000812	47F0 C8AA	008AA		828	B DEC900	GO NEXT 00081700
				829 *		00081800
				830 *	--- ZONED DECOMPRESSION ---	00081900
				831 *		00082000
000816				832	DEC300 DS 0H	00082100
000816	1744			833	XR R4,R4	ZERO OUT REGISTER 4 00082200
				834 *		00082300
000818				835	DEC310 DS 0H	00082400
000818	1722			836	XR R2,R2	ZERO OUT REGISTER 2 00082500
00081A	D200 C952 8000	00952	00000	837	MVC CHAR,0(R8)	GET CHARACTER 00082600
000820	94F0 C952	00952		838	NI CHAR,X'F0'	ISOLATE UPPER HALF 00082700
000824	4320 C952	00952		839	IC R2,CHAR	PUT IN R2 00082800
000828	8820 0004	00004		840	SRL R2,4(0)	SHIFT IT 00082900
00082C	4220 5000	00000		841	STC R2,0(R5)	STORE IN PLACE 00083000
000830	96F0 5000	00000		842	OI 0(R5),X'F0'	RECREATE ZONED NIBBLE 00083100
000834	4155 0001	00001		843	LA R5,1(R5)	INCREMENT R5 00083200
000838	4188 0001	00001		844	LA R8,1(R8)	INCREMENT R8 00083300
				845 *		00083400
00083C	4140 4001	00001		846	LA R4,1(R4)	INCREMENT COUNTER 00083500
000840	1947			847	CR R4,R7	COUNTER >= LENGTH ? 00083600
000842	47B0 C86E	0086E		848	BNL DEC390	YEP, READY 00083700
				849 *		00083800
000846	0680			850	BCTR R8,0	TEMPORARY DECREMENT R8 00083900
000848	D200 C952 8000	00952	00000	851	MVC CHAR,0(R8)	RE-GET CHARACTER 00084000
00084E	940F C952	00952		852	NI CHAR,X'0F'	ISOLATE LOWER HALF NOW 00084100
000852	D200 5000 C952	00000	00952	853	MVC 0(1,R5),CHAR	STORE IN PLACE 00084200
000858	96F0 5000	00000		854	OI 0(R5),X'F0'	RECREATE ZONE NIBBLE 00084300

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
00085C	4150 5001	00001		855	LA R5,1(,R5)	INCREMENT R5 00084400
000860	4180 8001	00001		856	LA R8,1(,R8)	INCREMENT R8 00084500
				857 *		00084600
000864	4140 4001	00001		858	LA R4,1(,R4)	INCREMENT COUNTER 00084700
000868	1947			859	CR R4,R7	COUNTER < LENGTH ? 00084800
00086A	4740 C818	00818		860	BL DEC310	YEP, AGAIN 00084900
				861 *		00085000
00086E				862	DEC390 DS 0H	00085100
00086E	47F0 C8AA	008AA		863	B DEC900	GO NEXT 00085200
				864 *		00085300
				865 *	--- TAKE OVER UNCOMPRESSED DATA ---	00085400
				866 *		00085500
000872				867	DEC800 DS 0H	00085600
				868 *		00085700
				869 *	--- LOAD LENGTH IN R7 ---	00085800
				870 *		00085900
000872	943F C952	00952		871	NI CHAR,LENGTH6	ZERO OUT HIGH BITS 00086000
000876	4370 C952	00952		872	IC R7,CHAR	INSERT LENGTH IN R7 00086100
00087A	9140 C953	00953		873	TM CONTROL,EXTLEN	EXTRA LENGTH BYTE ? 00086200
00087E	4780 C88E	0088E		874	BZ DEC820	NO, EXTRACT SHORT LENGTH 00086300
				875 *		00086400
000882	8970 0008	00008		876	SLL R7,8(0)	SHIFT LEFT ONE BYTE 00086500
000886	4378 0000	00000		877	IC R7,0(R8)	R7 NOW CONTAINS LENGTH 00086600
00088A	4188 0001	00001		878	LA R8,1(R8)	BUMP UP R8 00086700
				879 *		00086800
				880 *	--- TAKE OVER UNCOMPRESSED AREA; MOVE 0(R7,R6),0(R8) ---	00086900
				881 *		00087000
00088E				882	DEC820 DS 0H	00087100
00088E	9069 C908	00908		883	STM R6,R9,TEMPSAV	SAVE REGISTERS 00087200
000892	1865			884	LR R6,R5	PREPARE MVCL, DEST ADDRESS 00087300
				885 *		R7 IS LENGTH 00087400
000894	1847			886	LR R4,R7	SAVE LENGTH 00087500
				887 *		R8 -> SOURCE 00087600
000896	1897			888	LR R9,R7	LOAD LENGTH 00087700
000898	0E68			889	MVCL R6,R8	EXECUTE MOVE LONG 00087800
00089A	9869 C908	00908		890	LM R6,R9,TEMPSAV	RESTORE REGISTERS 00087900
				891 *		00088000
00089E	4154 5000	00000		892	LA R5,0(R4,R5)	T3 = T3 + LENGTH 00088100
0008A2	4184 8000	00000		893	LA R8,0(R4,R8)	T2 = T2 + LENGTH 00088200
				894 *		00088300
0008A6	47F0 C8AA	008AA		895	B DEC900	00088400
				896 *		00088500
				897 *	--- ARE WE READY YET ? ---	00088600
				898 *		00088700
0008AA				899	DEC900 DS 0H	00088800
0008AA	5980 C970	00970		900	C R8,EOR	READY ? 00088900
0008AE	47D0 C786	00786		901	BNH DEC010	NO, PROCEED 00089000
				902 *		00089100
0008B2	4140 B002	00002		903	LA R4,RECORD	RECORD START 00089200
0008B6	1B54			904	SR R5,R4	R5 EQUALS LENGTH 00089300
0008B8	4050 B000	00000		905	STH R5,RECLN	STORE LENGTH 00089400
				906 *		00089500
0008BC	47F0 C092	00092		907	B BACK	BACK TO CALLER 00089600

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				909 *	-----	00089800
				910 *		00089900
				911 *	WORKING STORAGE	00090000
0008C0				912	SAVEAREA DS 18F REGISTER SAVE AREA	00090100
000908				913	TEMPSAV DS 18F TEMP REGISTER SAVE AREA	00090200
				914 *		00090300
000950				915	ARLEN DS H LENGTH OF AREA	00090400
				916 *		00090500
000952				917	CHAR DS CL1 STORAGE FOR CHARACTER	00090600
				918 *		00090700
000953				919	CONTROL DS CL1 STORAGE FOR CONTROL CODE	00090800
		00080		920	PACKED EQU X'80' 1..... COMPRESSED INDICATOR	00090900
		00040		921	EXTLEN EQU X'40' .1..... LENGTH IN NEXT BYTE IF ON	00091000
		00020		922	REPLIC EQU X'20' ..10.... INDICATE REPLICATING CHARS	00091100
		00010		923	SPACES EQU X'10' ..01.... INDICATE SPACE AREA	00091200
		00000		924	ZONED EQU X'00' ..00.... INDICATE ZONED AREA	00091300
		00030		925	TYP COD EQU X'30' ..11.... COMPRESSION TYPE WHEN PACKED	00091400
		0000F		926	LENGTH4 EQU X'0F'1111 LENGTH BITS WHEN PACKED	00091500
		0003F		927	LENGTH6 EQU X'3F' ..111111 LENGTH BITS WHEN NOT PACKED	00091600
				928 *		00091700
000954				929	CONTROL2 DS CL1 2ND CONTROL BYTE	00091800
000955				930	CONTROL3 DS CL1 3RD CONTROL BYTE	00091900
				931 *		00092000
000958				932	NXTZON DS F NEXT ZONED AREA	00092100
00095C				933	NXTSPA DS F NEXT SPACE AREA	00092200
000960				934	NXTREP DS F NEXT REPLICATE AREA	00092300
				935 *		00092400
000964				936	T1 DS F T1 STORAGE	00092500
000968				937	T2 DS F T2 STORAGE	00092600
00096C				938	T3 DS F T3 STORAGE	00092700
000970				939	EOR DS F EOR STORAGE	00092800
				940 *		00092900
000978				941	LTORG	00093000
000978	C3D6D4D7			942	=C'COMP'	
00097C	C4C5C3D6			943	=C'DECO'	
000980	F0F0			944	=C'00'	
000982	F0F2			945	=C'02'	
000984	F9F9			946	=C'99'	
000986	4040			947	=C' '	
				948 *		00093100
000988				949	DS 0F	00093200
000988	0000000F			950	MAXL4 DC XL4'0000000F' MAX LENGTH FOR FOUR BITS	00093300
00098C	0000003F			951	MAXL6 DC XL4'0000003F' MAX LENGTH FOR SIX BITS	00093400
				952 *		00093500
000990	0000000F			953	HEX15 DC F'15' =X'0000000F'	00093600
000994	00002000			954	BUFMAX DC F'8192' MAXIMUM BUFFER LENGTH	00093700
000998				955	BUFFER DS CL8192 BUFFER STORAGE TO HOLD RECORD	00093800
				956 *		00093900

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 20.14 07/31/19
				958 *		00094100
				959 *	-----	00094200
				960 *	DUMMY SECTIONS	DSECTS 00094300
				961 *	-----	00094400
000000				962	PRMOPER DSECT	00094500
000000				963	OPERAT DS CL4	00094600
				964 *		00094700
000000				965	PRMRCOD DSECT	00094800
000000				966	RETCODE DS CL2	00094900
				967 *		00095000
000000				968	PRMRECD DSECT	00095100
000000				969	RECLEN DS H	00095200
000002				970	RECORD DS CL8192	00095300
				971 *		00095400
000000				972	END GAW8051	00095500

ASM 0201 20.14 07/31/19

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

HIGHEST SEVERITY WAS 0

OPTIONS FOR THIS ASSEMBLY

ALIGN, ALOGIC, BUFSIZE(STD), NODECK, ESD, FLAG(0), LINECOUNT(55), LIST, NOMCALL, YFLAG, WORKSIZE(2097152)

NOMLOGIC, NONUMBER, OBJECT, NORENT, RLD, NOSTMT, NOLIBMAC, NOTERMINAL, NOTEST, NOXREF(SHORT)

SYSPARM()

WORK FILE BUFFER SIZE/NUMBER = 8702/ 1

TOTAL RECORDS READ FROM SYSTEM INPUT 952

TOTAL RECORDS READ FROM SYSTEM LIBRARY 1124

TOTAL RECORDS PUNCHED 39

TOTAL RECORDS PRINTED 1024

F64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED XREF,LET,LIST,NCAL

DEFAULT OPTION(S) USED - SIZE=(98304,55296)

I EW0000

NAME GAW8051(R)

00095909

CROSS REFERENCE TABLE

CONTROL SECTION

ENTRY

NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
GAW8051	00	2998								

ENTRY ADDRESS 00

TOTAL LENGTH 2998

***GAW8051 NOW REPLACED IN DATA SET

AUTHORIZATION CODE IS 0.

```

GGGGGGGGGG  AAAAAAAAAA  WW      WW  AAAAAAAAAA  SSSSSSSSSS  MM      MM  LL
GGGGGGGGGGGG  AAAAAAAAAAAA  WW      WW  AAAAAAAAAAAA  SSSSSSSSSSSS  MMM      MMM  LL
GG      GG  AA      AA  WW      WW  AA      AA  SS      SS  MMMM      MMMM  LL
GG      AA      AA  WW      WW  AA      AA  SS      SS  MM  MM  MM  MM  LL
GG      AA      AA  WW      WW  AA      AA  SSS      SSS  MM  MMMM  MM  LL
GG      AAAAAAAAAA  WW      WW  AAAAAAAAAA  SSSSSSSSSS  MM      MM  MM  LL
GG      GGGGG  AAAAAAAAAAAA  WW  WW  WW  AAAAAAAAAAAA  SSSSSSSSSS  MM      MM  LL
GG      GGGGG  AA      AA  WW  WWW  WW  AA      AA      SSS  MM      MM  LL
GG      GG  AA      AA  WW  WW  WW  AA      AA      SS  MM      MM  LL
GG      GG  AA      AA  WWW  WWW  AA      AA  SS      SS  MM      MM  LL
GGGGGGGGGGGG  AA      AA  WWW  WWW  AA      AA  SSSSSSSSSSSS  MM      MM  LLLLLLLLLLLL
GGGGGGGGGGGG  AA      AA  WW      WW  AA      AA  SSSSSSSSSS  MM      MM  LLLLLLLLLLLL

```

```

JJJJJJJJJJ      444      00000000      444      AAAAAAAAAA
JJJJJJJJJJ      4444      0000000000      4444      AAAAAAAAAAAA
JJ      44 44      00      0000      44 44      AA      AA
JJ      44 44      00      00 00      44 44      AA      AA
JJ      44 44      00      00 00      44 44      AA      AA
JJ      444444444444      00 00 00      444444444444      AAAAAAAAAAAA
JJ      444444444444      00 00 00      444444444444      AAAAAAAAAAAA
JJ      44      00 00      00      44      AA      AA
JJ      JJ      44      0000      00      44      AA      AA
JJ      JJ      44      000      00      44      AA      AA
JJJJJJJJ      44      0000000000      44      AA      AA
JJJJJJ      44      00000000      44      AA      AA

```

```

****A  END  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  END  A****
****A  END  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  END  A****
****A  END  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  END  A****
****A  END  JOB  404  GAWASML  G.  WASSINK      ROOM      8.14.47 PM 31 JUL 19  PRINTER1  SYS TK4-  JOB  404  END  A****

```