

				CCCCCCCCC	UU		UU	TTTTTTTTTTT	IIIIIIIII	LL			00000000	00000000	11			
				CCCCCCCCC	UU		UU	TTTTTTTTTTT	IIIIIIIII	LL			0000000000	0000000000	111			
				CC	UU		UU	TT	II	LL			00	0000	00	0000	1111	
				CC	UU		UU	TT	II	LL			00	00	00	00	00	11
				CC	UU		UU	TT	II	LL			00	00	00	00	00	11
				CC	UU		UU	TT	II	LL			00	00	00	00	00	11
				CC	UU		UU	TT	II	LL			00	00	00	00	00	11
				CC	UU		UU	TT	II	LL			00	00	00	00	00	11
				CC	UU		UU	TT	II	LL			0000	00	0000	00	11	
				CC	UU		UU	TT	II	LL			0000	00	0000	00	11	
				CCCCCCCCC	UUUUUUUUUUUU		TT	IIIIIIIII	LLLLLLLLLLLLL			0000000000	0000000000	1111111111				
				CCCCCCCCC	UUUUUUUUUU		TT	IIIIIIIII	LLLLLLLLLLLLL			00000000	00000000	1111111111				

				JJJJJJJJJ	333333333		11		00000000	11			AAAAAAAAA	
				JJJJJJJJJ	33333333333		111		0000000000	111			AAAAAAAAAAAA	
				JJ	33		33	1111	00	0000	1111		AA	AA
				JJ			33	11	00	00	00	11	AA	AA
				JJ			33	11	00	00	00	11	AA	AA
				JJ	3333		11	00	00	00	11		AAAAAAAAAAAA	
				JJ	3333		11	00	00	00	11		AAAAAAAAAAAA	
				JJ			33	11	00	00	00	11	AA	AA
				JJ			33	11	0000	00	11		AA	AA
				JJ	JJ	33	33	11	000	00	11		AA	AA
				JJJJJJJ	33333333333		1111111111		0000000000	1111111111		AA	AA	
				JJJJJJ	333333333		1111111111		00000000	1111111111		AA	AA	

****A	START	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	START	A****
****A	START	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	START	A****
****A	START	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	START	A****
****A	START	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	START	A****

J E S 2 J O B L O G

16.49.03 JOB 3101 \$HASP373 CUTIL001 STARTED - INIT 1 - CLASS A - SYS BSP1
16.49.03 JOB 3101 IEF403I CUTIL001 - STARTED - TIME=16.49.03
16.49.03 JOB 3101 IEFACTRT - Stepname Procstep Program Retcode
16.49.03 JOB 3101 CUTIL001 ASM IFOX00 RC= 0000
16.49.03 JOB 3101 CUTIL001 LKED IEWL RC= 0000
16.49.03 JOB 3101 DYNABLDL STOPPED
16.49.03 JOB 3101 CUTIL001 DBSTOP IEFPROC DYNABLDL RC= 0000
16.49.03 JOB 3101 DYNABLDL STARTED
16.49.03 JOB 3101 CUTIL001 DBSTART IEFPROC DYNABLDL RC= 0000
16.49.03 JOB 3101 CUTIL001 ADDHELP IEBUPDTE RC= 0000
16.49.03 JOB 3101 CUTIL001 ADDCLST IEBUPDTE RC= 0000
16.49.04 JOB 3101 CUTIL001 ADDCLST IEBUPDTE RC= 0000
16.49.04 JOB 3101 CUTIL001 TESTIT0 STEP01 IKJEFT01 RC= 0000
16.49.07 JOB 3101 CUTIL001 TESTIT1 STEP01 IKJEFT01 RC= 0000
16.49.07 JOB 3101 IEF404I CUTIL001 - ENDED - TIME=16.49.07
16.49.07 JOB 3101 \$HASP395 CUTIL001 ENDED

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

12 APR 20 JOB EXECUTION DATE

9,803 CARDS READ

12,670 SYSOUT PRINT RECORDS

0 SYSOUT PUNCH RECORDS

0.06 MINUTES EXECUTION TIME

1 //CUTIL001 JOB (SYS),'Install CUTIL00 TSO',CLASS=A, JOB 3101

*** ** LENOVO2IN1: SUBMVS submitted
*** ** M:\\$myMainframe\\$submit_work\SUBMIT_CUTIL001_JOB12465.JCL
*** ** from source file (len=57):
*** ** m:\\$myMainframe\\$LARRY_MVS_CNTL\CUTIL00\V1000\\$inst01.JCL
*** RESTART=TESTIT1.STEP01,

// MSGCLASS=A,MSGLEVEL=(1,1)

*** -----*
*** * CUTIL00 for MVS3.8J TSO / Hercules *
*** * * *
*** * Install and Validate CUTIL00 TSO Components *
*** * - Review JOB card per your system requirements *
*** * - Review <--TARGET DDs per your system requirements *
*** * - CUTIL00 program installs to SYS2.CMDLIB *
*** * - CUTIL00 help installs to SYS2.HELP *
*** * - CCUTIL00 help installs to SYS2.HELP *
*** * - CCUTIL00 clist installs to SYS2.CMDPROC *
*** * - CCUTIL0V clist installs to SYS2.CMDPROC *
*** * - Validate CUTIL00 via BATCH TSO using CCUTIL0V *
*** * and CCUTIL00 stored in SYS2.CMDPROC *
*** * * *

*** * NOTE: *
*** * ----- *
*** * Two user-mods, ZP60014 and ZP60038, are REQUIRED *
*** * * *
*** * If the above user-mods are not installed, a system *
*** * abend 806-4 will occur when CUTIL00 is executed. *
*** * Additionally, an error message will display on *
*** * your TSO session screen: *
*** * 'CUTIL00 4098 -Cannot link to IKJCT441' *
*** * * *
*** * For more information, refer to the CUTIL00 program *
*** * documentation in this JCL jobstream. *
*** * * *

*** -----*

2 //ASM EXEC PGM=IFOX00,
// PARM='NODECK,LOAD,RENT,TERM'
3 //SYSGO DD DSN=&&LOADSET,DISP=(MOD,PASS),SPACE=(CYL,(1,1)),
// UNIT=VIO,DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)
4 //SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
5 // DD DSN=SYS1.AMODGEN,DISP=SHR
6 // DD DSN=SYS2.MACLIB,DISP=SHR ** YREG **
7 //SYSTEM DD SYSOUT=*
8 //SYSPRINT DD SYSOUT=*
9 //SYSPUNCH DD DSN=NULLFILE
10 //SYSUT1 DD UNIT=VIO,SPACE=(CYL,(6,1))
11 //SYSUT2 DD UNIT=VIO,SPACE=(CYL,(6,1))
12 //SYSUT3 DD UNIT=VIO,SPACE=(CYL,(6,1))
13 //SYSIN DD *
14 //LKED EXEC PGM=IEWL,PARM='MAP,LIST,LET,RENT,XREF',
// COND=(0,NE,ASM)
15 //SYSLIN DD DSN=&&LOADSET,DISP=(OLD,DELETE)
16 // DD DDNAME=SYSIN
17 //SYSLMOD DD DSN=SYS2.CMDLIB,DISP=SHR <--TARGET
18 //SYSPRINT DD SYSOUT=*
19 //SYSUT1 DD UNIT=VIO,SPACE=(CYL,(5,2))
20 //SYSIN DD *

*** -----*
*** * *
*** * If SYSn.LINKLIB or SYSn.CMDLIB is updated *
*** * *
*** -----*

21 //DBSTOP EXEC DBSTOP,

```

22 //COND.IEFPROC=(0,NE,ASM)
23 XXIEFPROC EXEC PGM=DYNABLDL,REGION=100K,PARM=P
//DBSTART EXEC DBSTART,
//COND.IEFPROC=(0,NE,ASM)
***
*** -----*
*** *
*** * HELP PDS Member Installation (CUTIL00, CCUTIL00)
*** *
*** -----*
24 XXIEFPROC EXEC PGM=DYNABLDL,REGION=100K
25 //ADDHELP EXEC PGM=IEBUPDTE,PARM=NEW,
//COND=(0,NE,ASM)
26 //SYSPRINT DD SYSOUT=*
27 //SYSUT2 DD DSN=SYS2.HELP,DISP=SHR <--TARGET
28 //SYSIN DD DATA,DLM='><'
29 //SYSIN DD * GENERATED STATEMENT
***
*** -----*
*** *
*** * CLIST PDS Member Installation (CCUTIL00)
*** *
*** -----*
30 //ADDCLST EXEC PGM=IEBUPDTE,PARM=NEW,
//COND=(0,NE,ASM)
31 //SYSPRINT DD SYSOUT=*
32 //SYSUT2 DD DSN=SYS2.CMDPROC,DISP=SHR <--TARGET
33 //SYSIN DD DATA,DLM='><'
***
*** -----*
*** *
*** * CLIST PDS Member Installation (CCUTIL0V)
*** *
*** -----*
34 //ADDCLST EXEC PGM=IEBUPDTE,PARM=NEW,
//COND=(0,NE,ASM)
35 //SYSPRINT DD SYSOUT=*
36 //SYSUT2 DD DSN=SYS2.CMDPROC,DISP=SHR <--TARGET
37 //SYSIN DD DATA,DLM='><'
***
*** -----*
*** *
*** * CUTIL00 validation using BATCH TSO and CCLIST00
*** *
*** * NOTE:
*** * -----
*** * Two user-mods, ZP60014 and ZP60038, are REQUIRED
*** *
*** * If the above user-mods are not installed, a system
*** * abend 806-4 will occur when CUTIL00 is executed.
*** * Additionally, an error message will display on
*** * your TSO session screen:
*** * 'CUTIL00 4098 -Cannot link to IKJCT441'
*** *
*** * For more information, refer to the CUTIL00 program
*** * documentation above in this JCL jobstream.
*** *
*** -----*
//BATCHTSO PROC
//STEP01 EXEC PGM=IKJEFT01
//SYSPROC DD DISP=SHR,DSN=SYS2.CMDPROC <--TARGET
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY Command Line Input
//PEND
***
38 //TESTIT0 EXEC BATCHTSO,
//COND.STEP01=(0,NE,ASM)

```

```
39  ++BATCHTSO  PROC
40  ++STEP01   EXEC  PGM=IKJEFT01
41  ++SYSPROC  DD    DISP=SHR,DSN=SYS2.CMDPROC          <--TARGET
42  ++SYSPRINT DD    SYSOUT=*
43  ++SYSTSPRT DD    SYSOUT=*
44  //STEP01.SYSTSIN DD *
    +/SYSTSIN  DD    DUMMY          Command Line Input
45  //TESTIT1  EXEC  BATCHTSO,
    //          COND.STEP01=(0,NE,ASM)
46  ++BATCHTSO  PROC
47  ++STEP01   EXEC  PGM=IKJEFT01
48  ++SYSPROC  DD    DISP=SHR,DSN=SYS2.CMDPROC          <--TARGET
49  ++SYSPRINT DD    SYSOUT=*
50  ++SYSTSPRT DD    SYSOUT=*
51  //STEP01.SYSTSIN DD *
    +/SYSTSIN  DD    DUMMY          Command Line Input
    //
```

```

IEF236I ALLOC. FOR CUTIL001 ASM
IEF237I VIO  ALLOCATED TO SYSGO
IEF237I 148  ALLOCATED TO SYSLIB
IEF237I 248  ALLOCATED TO
IEF237I 148  ALLOCATED TO
IEF237I JES2 ALLOCATED TO SYSTEM
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I DMY  ALLOCATED TO SYSPUNCH
IEF237I VIO  ALLOCATED TO SYSUT1
IEF237I VIO  ALLOCATED TO SYSUT2
IEF237I VIO  ALLOCATED TO SYSUT3
IEF237I JES2 ALLOCATED TO SYSIN
IEF142I CUTIL001 ASM - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20103.T164903.RA000.CUTIL001.LOADSET      PASSED      *-----23
IEF285I   SYS1.MACLIB                                KEPT          *-----41
IEF285I   VOL SER NOS= MVSRES.
IEF285I   SYS1.AMODGEN                                KEPT          *-----29
IEF285I   VOL SER NOS= MVSDLB.
IEF285I   SYS2.MACLIB                                KEPT          *-----0
IEF285I   VOL SER NOS= MVSRES.
IEF285I   JES2.JOB03101.S00109                        SYSOUT
IEF285I   JES2.JOB03101.S00110                        SYSOUT
IEF285I   SYS20103.T164903.RA000.CUTIL001.R0000001    DELETED      *-----306
IEF285I   SYS20103.T164903.RA000.CUTIL001.R0000002    DELETED      *-----129
IEF285I   SYS20103.T164903.RA000.CUTIL001.R0000003    DELETED      *-----82
IEF285I   JES2.JOB03101.SI0101                        SYSIN
IEF373I STEP /ASM      / START 20103.1649
IEF374I STEP /ASM      / STOP 20103.1649 CPU      0MIN 00.48SEC SRB      0MIN 00.03SEC VIRT  768K SYS  412K
*****
*      1. Jobstep of job: CUTIL001      Stepname: ASM      Program name: IFOX00      Executed on 12.04.20 from 16.49.03 to 16.49.03 *
*      elapsed time 24:00:00,57      CPU-Identifier: BSP1      Page-in: 0 *
*      CPU time 00:00:00,51      Virtual Storage used: 768K      Page-out: 0 *
*      corr. CPU: 00:00:00,51      CPU time has been corrected by 1 / 1,0 multiplier *
* * * * *
*      I/O Operation *
*      Number of records read via DD * or DD DATA: 7643 *
*      FFF.....23 148.....41 248.....29 148.....0 DMY.....0 DMY.....0 DMY.....0 FFF.....306 FFF.....129 FFF.....82 *
*      DMY.....0 *
* * * * *
*      Charge for step (w/o SYSOUT): 0,85 *
*****
IEF236I ALLOC. FOR CUTIL001 LKED
IEF237I VIO  ALLOCATED TO SYSLIN
IEF237I JES2 ALLOCATED TO
IEF237I 197  ALLOCATED TO SYSLMOD
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I VIO  ALLOCATED TO SYSUT1
IEF142I CUTIL001 LKED - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20103.T164903.RA000.CUTIL001.LOADSET      DELETED      *-----24
IEF285I   JES2.JOB03101.SI0102                        SYSIN
IEF285I   SYS2.CMDLIB                                KEPT          *-----10
IEF285I   VOL SER NOS= PUB007.
IEF285I   JES2.JOB03101.S00111                        SYSOUT
IEF285I   SYS20103.T164903.RA000.CUTIL001.R0000004    DELETED      *-----0
IEF373I STEP /LKED      / START 20103.1649
IEF374I STEP /LKED      / STOP 20103.1649 CPU      0MIN 00.01SEC SRB      0MIN 00.00SEC VIRT  260K SYS  252K
*****
*      2. Jobstep of job: CUTIL001      Stepname: LKED      Program name: IEWL      Executed on 12.04.20 from 16.49.03 to 16.49.03 *
*      elapsed time 24:00:00,02      CPU-Identifier: BSP1      Page-in: 0 *
*      CPU time 00:00:00,01      Virtual Storage used: 260K      Page-out: 0 *
*      corr. CPU: 00:00:00,01      CPU time has been corrected by 1 / 1,0 multiplier *
* * * * *
*      I/O Operation *

```

```
*      Number of records read via DD * or DD DATA:      1      *
*      FFF.....24 DMY.....0 197.....10 DMY.....0 FFF.....0      *
*
*
*      Charge for step (w/o SYSOUT):      0,01      *
*****
IEF236I ALLOC. FOR CUTIL001 IEFPROC DBSTOP
DYNABLDL STOPPED
IEF142I CUTIL001 IEFPROC DBSTOP - STEP WAS EXECUTED - COND CODE 0000
IEF373I STEP /IEFPROC / START 20103.1649
IEF374I STEP /IEFPROC / STOP 20103.1649 CPU      0MIN 00.00SEC SRB      0MIN 00.00SEC VIRT      16K SYS      156K
IEF236I ALLOC. FOR CUTIL001 IEFPROC DBSTART
DYNABLDL STARTED
IEF142I CUTIL001 IEFPROC DBSTART - STEP WAS EXECUTED - COND CODE 0000
IEF373I STEP /IEFPROC / START 20103.1649
IEF374I STEP /IEFPROC / STOP 20103.1649 CPU      0MIN 00.00SEC SRB      0MIN 00.00SEC VIRT      16K SYS      156K
IEF236I ALLOC. FOR CUTIL001 ADDHELP
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 240 ALLOCATED TO SYSUT2
IEF237I JES2 ALLOCATED TO SYSIN
IEF237I JES2 ALLOCATED TO SYSIN
IEF142I CUTIL001 ADDHELP - STEP WAS EXECUTED - COND CODE 0000
IEF285I JES2.JOB03101.S00112      SYSOUT
IEF285I SYS2.HELP      KEPT      *-----4
IEF285I VOL SER NOS= PUB000.
IEF285I JES2.JOB03101.SI0103      SYSIN
IEF285I JES2.JOB03101.SI0104      SYSIN
IEF373I STEP /ADDHELP / START 20103.1649
IEF374I STEP /ADDHELP / STOP 20103.1649 CPU      0MIN 00.01SEC SRB      0MIN 00.00SEC VIRT      48K SYS      184K
*****
*      5. Jobstep of job: CUTIL001      Stepname: ADDHELP      Program name: IEBUPDTE      Executed on 12.04.20 from 16.49.03 to 16.49.03      *
*      elapsed time 24:00:00,01      CPU-Identifier: BSP1      Page-in:      0      *
*      CPU time 00:00:00,01      Virtual Storage used: 48K      Page-out:      0      *
*      corr. CPU: 00:00:00,01      CPU time has been corrected by 1 / 1,0 multiplier      *
*
*      I/O Operation      *
*      Number of records read via DD * or DD DATA:      219      *
*      DMY.....0 240.....4 DMY.....0 DMY.....0      *
*
*      Charge for step (w/o SYSOUT):      0,01      *
*****
IEF236I ALLOC. FOR CUTIL001 ADDCLST
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 195 ALLOCATED TO SYSUT2
IEF237I JES2 ALLOCATED TO SYSIN
IEF142I CUTIL001 ADDCLST - STEP WAS EXECUTED - COND CODE 0000
IEF285I JES2.JOB03101.S00113      SYSOUT
IEF285I SYS2.CMDPROC      KEPT      *-----4
IEF285I VOL SER NOS= PUB005.
IEF285I JES2.JOB03101.SI0105      SYSIN
IEF373I STEP /ADDCLST / START 20103.1649
IEF374I STEP /ADDCLST / STOP 20103.1649 CPU      0MIN 00.01SEC SRB      0MIN 00.00SEC VIRT      48K SYS      184K
*****
*      6. Jobstep of job: CUTIL001      Stepname: ADDCLST      Program name: IEBUPDTE      Executed on 12.04.20 from 16.49.03 to 16.49.03      *
*      elapsed time 24:00:00,01      CPU-Identifier: BSP1      Page-in:      0      *
*      CPU time 00:00:00,01      Virtual Storage used: 48K      Page-out:      0      *
*      corr. CPU: 00:00:00,01      CPU time has been corrected by 1 / 1,0 multiplier      *
*
*      I/O Operation      *
*      Number of records read via DD * or DD DATA:      167      *
*      DMY.....0 195.....4 DMY.....0      *
*
*      Charge for step (w/o SYSOUT):      0,01      *
*****
IEF236I ALLOC. FOR CUTIL001 ADDCLST
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 195 ALLOCATED TO SYSUT2
```

```

IEF237I JES2 ALLOCATED TO SYSIN
IEF142I CUTIL001 ADDCLST - STEP WAS EXECUTED - COND CODE 0000
IEF285I JES2.JOB03101.S00114 SYSOUT
IEF285I SYS2.CMDPROC KEPT *-----10
IEF285I VOL SER NOS= PUB005.
IEF285I JES2.JOB03101.SI0106 SYSIN
IEF373I STEP /ADDCLST / START 20103.1649
IEF374I STEP /ADDCLST / STOP 20103.1649 CPU 0MIN 00.02SEC SRB 0MIN 00.00SEC VIRT 48K SYS 184K
*****
* 7. Jobstep of job: CUTIL001 Stepname: ADDCLST Program name: IEBUPDTE Executed on 12.04.20 from 16.49.03 to 16.49.04 *
* elapsed time 24:00:00,03 CPU-Identifier: BSP1 Page-in: 0 *
* CPU time 00:00:00,02 Virtual Storage used: 48K Page-out: 0 *
* corr. CPU: 00:00:00,02 CPU time has been corrected by 1 / 1,0 multiplier *
* *
* I/O Operation *
* Number of records read via DD * or DD DATA: 1549 *
* DMY.....0 195.....10 DMY.....0 *
* *
* Charge for step (w/o SYSOUT): 0,03 *
*****
IEF236I ALLOC. FOR CUTIL001 STEP01 TESTIT0
IEF237I 195 ALLOCATED TO SYSPROC
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSTSPRT
IEF237I JES2 ALLOCATED TO SYSTSIN
IEF142I CUTIL001 STEP01 TESTIT0 - STEP WAS EXECUTED - COND CODE 0000
IEF285I SYS2.CMDPROC KEPT *-----9
IEF285I VOL SER NOS= PUB005.
IEF285I JES2.JOB03101.S00115 SYSOUT
IEF285I JES2.JOB03101.S00116 SYSOUT
IEF285I JES2.JOB03101.SI0107 SYSIN
IEF373I STEP /STEP01 / START 20103.1649
IEF374I STEP /STEP01 / STOP 20103.1649 CPU 0MIN 00.65SEC SRB 0MIN 00.00SEC VIRT 152K SYS 244K
*****
* 8. Jobstep of job: CUTIL001 Stepname: STEP01 Program name: IKJEFT01 Executed on 12.04.20 from 16.49.04 to 16.49.04 *
* elapsed time 24:00:00,67 CPU-Identifier: BSP1 Page-in: 0 *
* CPU time 00:00:00,65 Virtual Storage used: 152K Page-out: 0 *
* corr. CPU: 00:00:00,65 CPU time has been corrected by 1 / 1,0 multiplier *
* *
* I/O Operation *
* Number of records read via DD * or DD DATA: 1 *
* 195.....9 DMY.....0 DMY.....0 DMY.....0 *
* *
* Charge for step (w/o SYSOUT): 1,08 *
*****
IEF236I ALLOC. FOR CUTIL001 STEP01 TESTIT1
IEF237I 195 ALLOCATED TO SYSPROC
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSTSPRT
IEF237I JES2 ALLOCATED TO SYSTSIN
IEF142I CUTIL001 STEP01 TESTIT1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I SYS2.CMDPROC KEPT *-----207
IEF285I VOL SER NOS= PUB005.
IEF285I JES2.JOB03101.S00117 SYSOUT
IEF285I JES2.JOB03101.S00118 SYSOUT
IEF285I JES2.JOB03101.SI0108 SYSIN
IEF373I STEP /STEP01 / START 20103.1649
IEF374I STEP /STEP01 / STOP 20103.1649 CPU 0MIN 02.27SEC SRB 0MIN 00.03SEC VIRT 100K SYS 248K
*****
* 9. Jobstep of job: CUTIL001 Stepname: STEP01 Program name: IKJEFT01 Executed on 12.04.20 from 16.49.04 to 16.49.07 *
* elapsed time 24:00:02,34 CPU-Identifier: BSP1 Page-in: 0 *
* CPU time 00:00:02,30 Virtual Storage used: 100K Page-out: 0 *
* corr. CPU: 00:00:02,30 CPU time has been corrected by 1 / 1,0 multiplier *
* *
* I/O Operation *
* Number of records read via DD * or DD DATA: 82 *

```

195.....207 DMY.....0 DMY.....0 DMY.....0

***** Charge for step (w/o SYSOUT): 3,83 *****

IEF375I JOB /CUTIL001/ START 20103.1649
IEF376I JOB /CUTIL001/ STOP 20103.1649 CPU 0MIN 03.45SEC SRB 0MIN 00.06SEC

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID	ASM 0201 16.49 04/12/20
CUTIL00	SD	0001	000000	002F52		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

				2	* * * * *	
--	--	--	--	---	-----------	--

				3	*	
--	--	--	--	---	---	--

				4	*	
--	--	--	--	---	---	--

				4	=====	
--	--	--	--	---	-------	--

				5	*	
--	--	--	--	---	---	--

				6	* CCCC UU UU TTTTTTTT IIIIIIII LL 0000 0000	
--	--	--	--	---	---	--

				7	* CC CC UU UU TT II LL 00 00 00 00	
--	--	--	--	---	------------------------------------	--

				8	* CC UU UU TT II LL 00 00 00 00	
--	--	--	--	---	---------------------------------	--

				9	* CC UU UU TT II LL 00 00 00 00	
--	--	--	--	---	---------------------------------	--

				10	* CC UU UU TT II LL 00 00 00 00	
--	--	--	--	----	---------------------------------	--

				11	* CC CC UUU UUU TT II LL 00 00 00 00	
--	--	--	--	----	--------------------------------------	--

				12	* CCCC UUUU TT IIIIIIII LLLLLLLL 0000 0000	
--	--	--	--	----	--	--

				13	*	
--	--	--	--	----	---	--

				14	*	
--	--	--	--	----	---	--

				14	=====	
--	--	--	--	----	-------	--

				15	* 12345678--12345678--12345678--12345678--12345678--12345678--12345678	
--	--	--	--	----	--	--

				16	*	
--	--	--	--	----	---	--

				17	* Program: CUTIL00	
--	--	--	--	----	--------------------	--

				18	*	
--	--	--	--	----	---	--

				19	* Author: Larry Belmontes Jr.	
--	--	--	--	----	-------------------------------	--

				20	* https://ShareABitofIT.net/CUTIL00-for-MVS-3-8J	
--	--	--	--	----	---	--

				21	* Copyright (C) 2020 Larry Belmontes, Jr.	
--	--	--	--	----	---	--

				22	*	
--	--	--	--	----	---	--

				23	* Disclaimer: <DSCLAIMR>	
--	--	--	--	----	--------------------------	--

				24	*	
--	--	--	--	----	---	--

				24	=====	
--	--	--	--	----	-------	--

				25	*	
--	--	--	--	----	---	--

				26	* No guarantee; No warranty; Install / Use at your own risk.	
--	--	--	--	----	--	--

				27	*	
--	--	--	--	----	---	--

				28	* This software is provided "AS IS" and without any expressed	
--	--	--	--	----	---	--

				29	* or implied warranties, including, without limitation, the	
--	--	--	--	----	---	--

				30	* implied warranties of merchantability and fitness for a	
--	--	--	--	----	---	--

				31	* particular purpose.	
--	--	--	--	----	-----------------------	--

				32	*	
--	--	--	--	----	---	--

				33	* The author requests keeping authors name intact in any	
--	--	--	--	----	--	--

				34	* modified versions.	
--	--	--	--	----	----------------------	--

				35	*	
--	--	--	--	----	---	--

				36	* In addition, the author requests submissions regarding any	
--	--	--	--	----	--	--

				37	* code modifications / enhancements and/or associated comments	
--	--	--	--	----	--	--

				38	* for consideration into a subsequent release (giving credit	
--	--	--	--	----	--	--

				39	* to contributor(s)) thus, improving overall functionality	
--	--	--	--	----	--	--

				40	* benefiting the MVS 3.8J hobbyist public domain community.	
--	--	--	--	----	---	--

				41	*	
--	--	--	--	----	---	--

				42	*	
--	--	--	--	----	---	--

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
44 * Overview:
45 * =====
46 *
47 * This program, CUTIL00, performs various functions on symbolic
48 * variables given a function code and variable names containing
49 * such data in a TSO CLIST.
50 *
51 * Given the limited CLIST functions available under MVS 3.8J,
52 * simple functions available in other languages and current OS
53 * versions are enhanced for your CLIST programming convenience.
54 *
55 * For example, centering content or trimming of strings or date
56 * conversion:
57 *
58 * String (before) String (after)
59 * -----
60 * 'center this TEXT ' 'center this TEXT '
61 * 'trim this TEXT area ' 'trim this TEXT area'
62 * '12311987' 'December 31, 1987'
63 *
64 * Invoking this utility is kept to a simple line command using
65 * two to four parameters:
66 *
67 * CUTIL00 FUNCT VAR1 VAR2
68 * where; FUNCT - function being requested
69 * VAR1 - variable name containing source string
70 * VAR2 - variable name for results or parameters,
71 * which may be optional depending of function
72 * Note: Other parameters are available. See Command Syntax.
73 *
74 * This program is written in IFOX00 Assembler (24-bit addressing)
75 * using the MVS 3.8J Tur(n)key 3 Operating System.
76 *
77 * CUTIL00 was developed and tested using Volker Bandke's MVS CD -
78 * MVS38J TK3 with Hercules 3.13 hosted on a Windows 10 Pro machine.
79 *
80 * MVS38J TK4- update 8 (maintained by Juergen Winkelmann) includes
81 * user-mod ZP60014. After applying user-mod ZP60038, CUTIL00 was
82 * installed and successfully validated with a CLIST harness under
83 * TSO.
84 *
85 * Enjoy the added value to CLIST development with CUTIL00!
86 * Larry Belmontes Jr.
87 *
88 *
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

90	*				Prerequisite: User Modifications <PREREQS>	
91	*				=====	
92	*					
93	*				Two user-mods, ZP60014 and ZP60038, are REQUIRED to process	
94	*				CLIST symbolic variables via the IKJCT441 API on MVS 3.8J before	
95	*				using CUTIL00. Otherwise, CUTIL00 is rendered useless!	
96	*					
97	*				More information on the above user-mods can be obtained from	
98	*				the following website:	
99	*				http://www.prycroft6.com.au/vs2mods/	
100	*					
101	*				If the above user-mods are not installed, a system abend	
102	*				806-4 will occur when CUTIL00 is executed. Additionally, an	
103	*				error message will be displayed on your TSO session screen:	
104	*				'CUTIL00 4098 -Cannot link to IKJCT441'	
105	*					
106	*				Check your system to determine if one or both user-mods are	
107	*				required. ZP60038 requires ZP60014.	
108	*					
109	*					

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

111	*				Command Syntax for CUTIL00: <CMDSYNTAX>	
112	*				=====	
113	*					
114	*				TSO TPM:	
115	*				CUTIL00 FUNCT VAR1 VAR2 QUOTE	
116	*					
117	*					
118	*				TSO Called:	
119	*				CALL 'pds(CUTIL00)' 'FUNCT VAR1 VAR2 QUOTE'	
120	*					
121	*				where:	
122	*					
123	*				1) FUNCT required, function to be performed	
124	*					
125	*				2) VAR1 required, name of CLIST variable	
126	*				o is not prefixed with an '&'	
127	*				o variable must exist before invoking CUTIL00	
128	*				o 1-8 bytes in length, must follow CLIST variable	
129	*				naming conventions	
130	*					
131	*				3) VAR2 optional, name of CLIST variable to store results	
132	*				or contain function parameters	
133	*				o is not prefixed with an '&'	
134	*				o 1-8 bytes in length, must follow CLIST variable	
135	*				naming conventions	
136	*				o Storing results	
137	*				- If variable does not exist, it will be created	
138	*				- If variable does exist, it will be updated	
139	*				- If no variable is specifed, use VAR1 variable	
140	*				- If '\$NOVAR2' is specifed, use VAR1 variable	
141	*				- '\$NOVAR2' used as a placeholder if VAR2	
142	*				is not specified with QUOTE option	
143	*				o Function Parameters	
144	*				- variable must exist before invoking CUTIL00	
145	*					
146	*				4) QUOTE optional, quote delimited string values	
147	*				o If 'QUOTE' is specified , string content between	
148	*				first and last quotes comprise VAR1 value.	
149	*				Use this option to include leading or trailing	
150	*				spaces in a string.	
151	*					

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

153	*				Command Syntax for CUTIL00: <CMDSYNTAX> (continued)	
154	*				=====	
155	*				The following variable names provide additional information	
156	*				after successful execution of CUTIL00:	
157	*					
158	*				o &LASTCC contains return code from CUTIL00 after execution	
159	*				o &ERRMSG contains error message text for function executed	
160	*				o &var1L contains length of var1 after execution	
161	*				o &var2L contains length of var2 after execution	
162	*					
163	*				NOTE: The above variables may not be declared by CUTIL00 if	
164	*				CUTIL00 abends or does not execute. Variables include	
165	*				&ERRMSG, &var1L and &var2L. Suggested to initialize	
166	*				to NULLS before invoking CUTIL00. See LTRIM CLIST	
167	*				example below.	
168	*					
169	*				NOTE: - CLIST variable name length is 8 bytes max	
170	*				- CLIST variable data length is 256 bytes max	
171	*				- Result data length is 256 bytes max	
172	*					
173	*					

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

				175 *	Function List:	
--	--	--	--	-------	----------------	--

				176 *	-----	
--	--	--	--	-------	-------	--

				177 *00	ABOUT - Display mySTAMP information	
--	--	--	--	---------	-------------------------------------	--

				178 *		
--	--	--	--	-------	--	--

				179 *01	LTRIM - remove leading spaces	
--	--	--	--	---------	-------------------------------	--

				180 *02	RTRIM - remove trailing spaces	
--	--	--	--	---------	--------------------------------	--

				181 *03	TRIM - remove leading and trailing	
--	--	--	--	---------	------------------------------------	--

				182 *04	TRIM\$ - remove leading, trailing, and duplicate spaces	
--	--	--	--	---------	---	--

				183 *		
--	--	--	--	-------	--	--

				184 *05	INDEX - Alias to FIND	
--	--	--	--	---------	-----------------------	--

				185 *06	INDEXB - Alias to FINDL	
--	--	--	--	---------	-------------------------	--

				186 *		
--	--	--	--	-------	--	--

				187 *07	ISNUM - Test for all numeric (0-9)	
--	--	--	--	---------	------------------------------------	--

				188 *08	ISALPHA - Test for all alphabetic (A-Z,a-z)	
--	--	--	--	---------	---	--

				189 *09	ISLOWER - Test for all alphabetic Lower (a-z)	
--	--	--	--	---------	---	--

				190 *10	ISUPPER - Test for all alphabetic Upper (A-Z)	
--	--	--	--	---------	---	--

				191 *11	ISBLANK - Test for all blank (whitespace)	
--	--	--	--	---------	---	--

				192 *12	ISDSN - Test for valid MVS Dataset Name	
--	--	--	--	---------	---	--

				193 *		
--	--	--	--	-------	--	--

				194 *13	ECHO - Display string value	
--	--	--	--	---------	-----------------------------	--

				195 *14	ECHOQ - Display string value surrounded by single quotes	
--	--	--	--	---------	--	--

				196 *		
--	--	--	--	-------	--	--

				197 *15	REVRS - Translate to reverse A-Z > Z-A, 0-9 > 9-0	
--	--	--	--	---------	---	--

				198 *		
--	--	--	--	-------	--	--

				199 *16	UPPERC - Translate to uppercase	
--	--	--	--	---------	---------------------------------	--

				200 *17	LOWERC - Translate to lowercase	
--	--	--	--	---------	---------------------------------	--

				201 *		
--	--	--	--	-------	--	--

				202 *18	COUNT - Count occurrences of a search string	
--	--	--	--	---------	--	--

				203 *18	FINDALL - Alias COUNT	
--	--	--	--	---------	-----------------------	--

				204 *19	FIND - Find starting position of FIRST search string	
--	--	--	--	---------	--	--

				205 *20	FINDL - Find starting position of LAST search string	
--	--	--	--	---------	--	--

				206 *		
--	--	--	--	-------	--	--

				207 *21	CENTER - Center content	
--	--	--	--	---------	-------------------------	--

				208 *22	LJUST - Left justify content	
--	--	--	--	---------	------------------------------	--

				209 *23	RJUST - Right justify content	
--	--	--	--	---------	-------------------------------	--

				210 *		
--	--	--	--	-------	--	--

				211 *24	ZFILL - Zero fill after content is right justified	
--	--	--	--	---------	--	--

				212 *25	WORDS - Number of words (contiguous non-blank groups)	
--	--	--	--	---------	---	--

				213 *26	GEN# - Generate a 3 digit number (000-999)	
--	--	--	--	---------	--	--

				214 *		
--	--	--	--	-------	--	--

				215 *27	DD2DSN - Dataset name associated with DD name	
--	--	--	--	---------	---	--

				216 *28	JOBINFO - Job Information	
--	--	--	--	---------	---------------------------	--

				217 *		
--	--	--	--	-------	--	--

				218 *29	DAYSMM - Number of days in MM	
--	--	--	--	---------	-------------------------------	--

				219 *30	DAYSYY - Number of days in CCYY (365 or 366)	
--	--	--	--	---------	--	--

				220 *31	ISLEAP - Test CCYY for leap year	
--	--	--	--	---------	----------------------------------	--

				221 *		
--	--	--	--	-------	--	--

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

223 * Function List: (continued)

224 * -----

225 *32 CYJ-D8 - Date Conversion CCYYJJJ to monthname DD, CCYY

226 *33 CYJ-DAY - Date Conversion CCYYJJJ to day name

227 *34 CYJ-DOW - Date Conversion CCYYJJJ to day of week number

228 *35 CYJ-mdcy - Date Conversion CCYYJJJ to MMDDCCYY (any order)

229 *

230 *36 JCY-D8 - Date Conversion JJJCCYY to monthname DD, CCYY

231 *37 JCY-DAY - Date Conversion JJJCCYY to day name

232 *38 JCY-DOW - Date Conversion JJJCCYY to day of week number

233 *39 JCY-mdcy - Date Conversion JJJCCYY to MMDDCCYY (any order)

234 *

235 *40 MDCY-D8 - Date Conversion MMDDCCYY to monthname DD, CCYY

236 *41 MDCY-DAY - Date Conversion MMDDCCYY to day name

237 *42 MDCY-DOW - Date Conversion MMDDCCYY to day of week number

238 *43 MDCY-cyj - Date Conversion MMDDCCYY to CCYYJJJ (any order)

239 *

240 *44 DMCY-D8 - Date Conversion DDMMCCYY to monthname DD, CCYY

241 *45 DMCY-DAY - Date Conversion DDMMCCYY to day name

242 *46 DMCY-DOW - Date Conversion DDMMCCYY to day of week number

243 *47 DMCY-cyj - Date Conversion DDMMCCYY to CCYYJJJ (any order)

244 *

245 *48 CYMD-D8 - Date Conversion CCYYMMDD to monthname DD, CCYY

246 *49 CYMD-DAY - Date Conversion CCYYMMDD to day name

247 *50 CYMD-DOW - Date Conversion CCYYMMDD to day of week number

248 *51 CYMD-cyj - Date Conversion CCYYMMDD to CCYYJJJ (any order)

249 *

250 *52 CYDM-D8 - Date Conversion CCYYDDMM to monthname DD, CCYY

251 *53 CYDM-DAY - Date Conversion CCYYDDMM to day name

252 *54 CYDM-DOW - Date Conversion CCYYDDMM to day of week number

253 *55 CYDM-cyj - Date Conversion CCYYDDMM to CCYYJJJ (any order)

254 *

255 *56 FILL - Fill VAR1 nnn times with delimiter

256 *

257 *57 LSTRIP - Remove leading delimiter

258 *58 RSTRIP - Remove trailing delimiter

259 *59 STRIP - Remove leading and trailing delimiter

260 *60 STRIP\$ - Remove all delimiter

261 *

262 *61 CONCAT - Concatenate two strings

263 *62 UNSTR - Unstring into separate variables

264 *

265 *63 REPLACE - Replace substring in string

266 *64 VEXIST - Test if CLIST Variable exists

267 *65 TDSN - Generate temporary DSN nodes

268 *66 NOW - Current system date and time

269 *67 PAD - PAD string with delimiter to given length

270 *

271 *

272 *

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
274 * -----
275 * | Function | VAR1 | VAR2 | Description |
276 * +-----+-----+-----+
277 *00| ABOUT   | -   | -   | Display program stamp information. |
278 * |         |     |     |                                     |
279 * -----
280 * CUTIL00 ABOUT                      /* ABOUT      */
281 * SET RC=&LASTCC                      /* RETURN CODE */
282 *
283 * CUTIL00 MVS3.8J Vv.r.mm 02202020 mm/dd/yy @ hh.mm
284 * |         |     |     |         |
285 * |         |     |     |         |
286 * +-----+-----+-----+
287 * |         |         |         |
288 * |         |         |         |
289 * |         |         |         |
290 * |         |         |         |
291 * |         |         |         |
292 * |         |         |         |
293 * |         |         |         |
294 * Results:  RC      =0
295 *           ERRMSG  ='SUCCESSFUL REQUEST'
296 *
297 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

299 * Functions:

300 *	-----
301 *	Function VAR1 VAR2 Description
302 *	+-----+-----+-----+
303 *01	LTRIM req opt Remove leading spaces from variable
304 *	VAR1.
305 *	
306 *	If VAR2 is omitted, results will be
307 *	placed into VAR1.
308 *	-----

309 * Statement Line:

310 *
311 * CUTIL00 LTRIM var1 var2 /* LTRIM */

312 *
313 * o var1 must be declared before invoking utility
314 * o var2 contains results, if specified.
315 * Otherwise, var1 contains results.
316 * o &LASTCC contains return code from CUTIL00
317 * o &ERRMSG contains error message information from CUTIL00
318 * o &var1L contains length of var1
319 * o &var2L contains length of var2, if var2 was specified
320 *

321 *
322 * Example CLIST:
323 * 0...0...1...1...2...2...3...3...4...4...5...5...6...6...
324 * 1...5...0...5...0...5...0...5...0...5...0...5...
325 * SET MYDATA=&STR(leading spaces) /* INIT MYDATA */
326 * SET MYRESULT= /* INIT MYRESULT */
327 * SET MYDATAL = /* INIT MYDATAL */
328 * SET MYRESULTL= /* INIT MYRESULTL */
329 * SET ERRMSG = /* INIT ERRMSG */
330 * CUTIL00 LTRIM MYDATA MYRESULT /* LTRIM */
331 * SET RC=&LASTCC /* RETURN CODE */
332 * WRITE CUTIL00 RC=&RC - &ERRMSG
333 * WRITE MYDATAL =&MYDATAL MYDATA='&MYDATA'
334 * WRITE MYRESULTL =&MYRESULTL MYRESULT='&MYRESULT'
335 * CUTIL00 ECHOQ MYRESULT
336 * /* */

337 *
338 * Results: RC =0
339 * ERRMSG ='SUCCESSFUL REQUEST'
340 * MYDATA =' LEADING SPACES'
341 * MYDATAL =17
342 * MYRESULT ='LEADING SPACES'
343 * MYRESULTL=14
344 *

345 * NOTE: MVS3.8J CLIST processor converts all characters to
346 * UPPERCASE! While CUTIL00 can convert a string to
347 * lowercase, the string will remain lowercase until
348 * referenced in a CLIST.
349 *

350 * req - VAR name is required
351 * opt - VAR name is optional
352 * no - VAR name is not required

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
354 * -----
355 * | Function | VAR1 | VAR2 | Description |
356 * +-----+-----+-----+
357 *02| RTRIM    | req  | opt  | Remove trailing spaces from variable
358 *      |          |      |      | VAR1.
359 *      |          |      |      |
360 *      |          |      |      | If VAR2 is omitted, results will be
361 *      |          |      |      | placed into VAR1.
362 * -----
363 * SET MYDATA=&STR(trailing spaces )          /* INIT MYDATA */
364 * SET MYRESULT=                             /* INIT MYRESULT */
365 * SET MYDATA =                             /* INIT MYDATA */
366 * SET MYRESULTL=                           /* INIT MYRESULTL */
367 * SET ERRMSG =                             /* INIT ERRMSG */
368 * CUTIL00 RTRIM MYDATA MYRESULT             /* RTRIM */
369 * SET RC=&LASTCC                             /* RETURN CODE */
370 *
371 * Results: RC =0
372 *           ERRMSG ='SUCCESSFUL REQUEST'
373 *           MYDATA ='TRAILING SPACES'
374 *           MYDATA =17
375 *           MYRESULT ='TRAILING SPACES'
376 *           MYRESULTL=15
377 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
379 * -----
380 * | Function | VAR1 | VAR2 | Description |
381 * +-----+-----+-----+
382 *03| TRIM      | req  | opt  | Remove leading and trailing spaces from
383 *      |          |      |      | variable VAR1.
384 *
385 *      |          |      |      | If VAR2 is omitted, results will be
386 *      |          |      |      | placed into VAR1.
387 * -----
388 * SET MYDATA=&STR( trim  my  data )          /* INIT MYDATA */
389 * SET MYRESULT=                             /* INIT MYRESULT */
390 * SET MYDATAL =                             /* INIT MYDATAL */
391 * SET MYRESULTL=                           /* INIT MYRESULTL */
392 * SET ERRMSG =                             /* INIT ERRMSG */
393 * CUTIL00 TRIM MYDATA MYRESULT              /* TRIM */
394 * SET RC=&LASTCC                            /* RETURN CODE */
395 *
396 *      Results:  RC          =0
397 *                ERRMSG     ='SUCCESSFUL REQUEST'
398 *                MYDATA     =' TRIM  MY  DATA '
399 *                MYDATAL    =17
400 *                MYRESULT   ='TRIM  MY  DATA'
401 *                MYRESULTL  =15
402 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
404 * -----
405 * | Function | VAR1 | VAR2 | Description |
406 * +-----+-----+-----+
407 *04| TRIM$    | req  | opt  | Remove leading, trailing and duplicate
408 *    |         |      |      | spaces from variable VAR1.
409 *    |         |      |      |
410 *    |         |      |      | If VAR2 is omitted, results will be
411 *    |         |      |      | placed into VAR1.
412 * -----
413 * SET MYDATA=&STR( trim  my  data )          /* INIT MYDATA */
414 * SET MYRESULT=                             /* INIT MYRESULT */
415 * SET MYDATAL =                             /* INIT MYDATAL */
416 * SET MYRESULTL=                           /* INIT MYRESULTL */
417 * SET ERRMSG =                             /* INIT ERRMSG */
418 * CUTIL00 TRIM$ MYDATA MYRESULT             /* TRIM$ */
419 * SET RC=&LASTCC                             /* RETURN CODE */
420 *
421 *      Results:  RC          =0
422 *                ERRMSG     ='SUCCESSFUL REQUEST'
423 *                MYDATA     =' TRIM  MY  DATA '
424 *                MYDATAL    =17
425 *                MYRESULT   ='TRIM MY DATA'
426 *                MYRESULTL  =12
427 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
429 * -----
430 * | Function | VAR1 | VAR2 | Description |
431 * +-----+-----+-----+
432 *05| INDEX   | req  | opt  | ALIAS to FIND, see FIND function. |
433 * | (alias) |      |      |      |
434 * -----
435 * SET MYDATA=&STR(my, string, ok? ) /* INIT MYDATA */
436 * SET DLMTR=&STR(",") /* INIT DLMTR */
437 * SET MYDATA1 = /* INIT MYDATA1 */
438 * SET DLMTRL = /* INIT DLMTRL */
439 * SET ERRMSG = /* INIT ERRMSG */
440 * CUTIL00 INDEX MYDATA DLMTR /* INDEX */
441 * SET RC=&LASTCC /* RETURN CODE */
442 *
443 * Results: RC =3
444 * ERRMSG = 'SUCCESSFUL REQUEST'
445 * MYDATA = 'MY, STRING, OK?'
446 * MYDATA1 =18
447 * DLMTR = '"', "'
448 * DLMTRL =3
449 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
451 * -----
452 * | Function | VAR1 | VAR2 | Description |
453 * +-----+-----+-----+
454 *06| INDEXB   | req  | opt  | ALIAS to FINDL, see FINDL function. |
455 * | (alias)   |      |      |
456 * -----
457 * SET MYDATA=&STR(my, string, ok? ) /* INIT MYDATA */
458 * SET DLMTR=&STR(",") /* INIT DLMTR */
459 * SET MYDATAL = /* INIT MYDATAL */
460 * SET DLMTRL = /* INIT DLMTRL */
461 * SET ERRMSG = /* INIT ERRMSG */
462 * CUTIL00 INDEXB MYDATA DLMTR /* INDEXB */
463 * SET RC=&LASTCC /* RETURN CODE */
464 *
465 * Results: RC =11
466 * ERRMSG = 'SUCCESSFUL REQUEST '
467 * MYDATA = 'MY, STRING, OK? '
468 * MYDATAL =18
469 * DLMTR = '"', "'
470 * DLMTRL =3
471 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
473 * -----
474 * | Function | VAR1 | VAR2 | Description |
475 * +-----+ +-----+ +-----+ +-----+
476 *07| ISNUM    | req   | no    | Test variable VAR1 for all numeric
477 *    |         |       |       | values (0-9) returning a true-false
478 *    |         |       |       | indication.
479 *    |         |       |       |
480 *    |         |       |       | RC=4000  &ERRMSG='FALSE'
481 *    |         |       |       | RC=4001  &ERRMSG='TRUE'
482 * -----
483 * SET MYDATA=&STR(0123456789)                /* INIT MYDATA */
484 * SET MYRESULT=                             /* INIT MYRESULT */
485 * SET MYDATA1 =                             /* INIT MYDATA1 */
486 * SET MYRESULT1=                             /* INIT MYRESULT1 */
487 * SET ERRMSG =                               /* INIT ERRMSG */
488 * CUTIL00 ISNUM MYDATA                       /* ISNUM */
489 * SET RC=&LASTCC                             /* RETURN CODE */
490 *
491 *      Results:  RC          =4001
492 *                ERRMSG     ='TRUE'
493 *                MYDATA      ='0123456789'
494 *                MYDATA1     =10
495 *                MYRESULT    =' '
496 *                MYRESULT1   =0
497 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
499 * -----
500 * | Function | VAR1 | VAR2 | Description |
501 * +-----+-----+-----+
502 *08 | ISALPHA | req | no | Test variable VAR1 for all alphabetic
503 * | | | | values (A-Z,a-z) returning a true-false
504 * | | | | indication.
505 * | | | |
506 * | | | | RC=4000 &ERRMSG='FALSE'
507 * | | | | RC=4001 &ERRMSG='TRUE'
508 * -----
509 * SET MYDATA=&STR(abCDEfghiJ) /* INIT MYDATA */
510 * SET MYRESULT= /* INIT MYRESULT */
511 * SET MYDATAL = /* INIT MYDATAL */
512 * SET MYRESULTL= /* INIT MYRESULTL */
513 * SET ERRMSG = /* INIT ERRMSG */
514 * CUTIL00 ISALPHA MYDATA /* ISALPHA */
515 * SET RC=&LASTCC /* RETURN CODE */
516 *
517 * Results: RC =4001
518 * ERRMSG ='TRUE'
519 * MYDATA ='ABCDEFGHIJ'
520 * MYDATAL =10
521 * MYRESULT =' '
522 * MYRESULTL=0
523 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
525 * -----
526 * | Function | VAR1 | VAR2 | Description |
527 * +-----+ +-----+ +-----+ +-----+
528 *09 | ISLOWER | req | no | Test variable VAR1 for all alphabetic
529 * | | | | lowercase (A-Z,a-z) returning a
530 * | | | | true-false indication.
531 * | | | |
532 * | | | | RC=4000 &ERRMSG='FALSE'
533 * | | | | RC=4001 &ERRMSG='TRUE'
534 * -----
535 * SET MYDATA=&STR(abcdefghij) /* INIT MYDATA */
536 * SET MYRESULT= /* INIT MYRESULT */
537 * SET MYDATAL = /* INIT MYDATAL */
538 * SET MYRESULTL= /* INIT MYRESULTL */
539 * SET ERRMSG = /* INIT ERRMSG */
540 * CUTIL00 ISALPHA MYDATA /* ISALPHA */
541 * SET RC=&LASTCC /* RETURN CODE */
542 *
543 * Results: RC =4000
544 * ERRMSG ='FALSE'
545 * MYDATA ='ABCDEFGHIJ'
546 * MYDATAL =10
547 * MYRESULT =' '
548 * MYRESULTL=0
549 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
551 * -----
552 * | Function | VAR1 | VAR2 | Description |
553 * +-----+ +-----+ +-----+ +-----+
554 *10 | ISUPPER  | req  | no   | Test variable VAR1 for all alphabetic
555 *   |         |      |      | uppercase (A-Z) returning a true-false
556 *   |         |      |      | indication.
557 *   |         |      |      |
558 *   |         |      |      | RC=4000  &ERRMSG='FALSE'
559 *   |         |      |      | RC=4001  &ERRMSG='TRUE'
560 * -----
561 * SET MYDATA=&STR(ABCDEFGH IJ)                /* INIT MYDATA */
562 * SET MYRESULT=                                /* INIT MYRESULT */
563 * SET MYDATAL =                                /* INIT MYDATAL */
564 * SET MYRESULTL=                              /* INIT MYRESULTL */
565 * SET ERRMSG =                                /* INIT ERRMSG */
566 * CUTIL00 ISUPPER MYDATA                      /* ISUPPER */
567 * SET RC=&LASTCC                              /* RETURN CODE */
568 *
569 *      Results:  RC          =4001
570 *                ERRMSG     ='TRUE'
571 *                MYDATA     ='ABCDEFGH IJ'
572 *                MYDATAL    =10
573 *                MYRESULT   =' '
574 *                MYRESULTL  =0
575 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
577 * -----
578 * | Function | VAR1 | VAR2 | Description |
579 * +-----+ +-----+ +-----+ +-----+
580 *11| ISBLANK  | req  | no   | Test variable VAR1 for all blank
581 *      |      |      |      | (spaces) returning a true-false
582 *      |      |      |      | indication.
583 *
584 *      |      |      |      | RC=4000  &ERRMSG='FALSE'
585 *      |      |      |      | RC=4001  &ERRMSG='TRUE'
586 *
587 * -----
588 * SET MYDATA=&STR(          ) /* INIT MYDATA */
589 * SET MYRESULT=             /* INIT MYRESULT */
590 * SET MYDATAL =             /* INIT MYDATAL */
591 * SET MYRESULTL=           /* INIT MYRESULTL */
592 * SET ERRMSG =             /* INIT ERRMSG */
593 * CUTIL00 ISBLANK MYDATA    /* ISBLANK */
594 * SET RC=&LASTCC           /* RETURN CODE */
595 *
596 *      Results:  RC      =4001
597 *                ERRMSG  ='TRUE'
598 *                MYDATA  =' '
599 *                MYDATAL  =9
600 *                MYRESULT =' '
601 *                MYRESULTL=0
602 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
604 * -----
605 * | Function | VAR1 | VAR2 | Description |
606 * +-----+ +-----+ +-----+ +-----+
607 *12 | ISDSN    | req  | no   | Test variable VAR1 for valid MVS
608 *    |         |      |      | dataset name returning a true-false
609 *    |         |      |      | indication.
610 *
611 *
612 *    |         |      |      | 0   5   1   1   2   2   3   3   4   4
613 *    |         |      |      | 1...5...0...5...0...5...0...5...0...4
614 *    |         |      |      | xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
615 *
616 *    MVS Dataset Naming standard:
617 *    - DSN is 1 to 44 bytes
618 *    - DSN can contain up to 22 segments
619 *    - DSN segment delimiter is a period (.)
620 *    - Each segment is 1-8 bytes
621 *    - Byte 1-1 in segment is A-Z #@$
622 *    - Byte 2-8 in segment is A-Z #@$ 0-9
623 *    - Case-sensitive, upper case
624 *
625 *
626 *    RC=4000  &ERRMSG='FALSE'
627 *    RC=4001  &ERRMSG='TRUE'
628 *
629 * -----
630 * SET MYDATA=&STR(A.A1234567)                /* INIT MYDATA */
631 * SET MYRESULT=                             /* INIT MYRESULT */
632 * SET MYDATAL =                             /* INIT MYDATAL */
633 * SET MYRESULTL=                           /* INIT MYRESULTL */
634 * SET ERRMSG =                             /* INIT ERRMSG */
635 * CUTIL00 ISDSN MYDATA                      /* ISDSN */
636 * SET RC=&LASTCC                             /* RETURN CODE */
637 *
638 * Results:  RC          =4001
639 *           ERRMSG     ='TRUE'
640 *           MYDATA     ='A.A1234567'
641 *           MYDATAL    =10
642 *           MYRESULT   =' '
643 *           MYRESULTL  =0
644 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
646 * -----
647 * | Function | VAR1 | VAR2 | Description |
648 * +-----+-----+-----+
649 *13| ECHO      | req  | no   | Display variable VAR1 value on screen. |
650 * |          |      |      |          |
651 * -----
652 * SET MYDATA=&STR(my data to display.)          /* INIT MYDATA */
653 * SET MYRESULT=                                /* INIT MYRESULT */
654 * SET MYDATAAL =                               /* INIT MYDATAAL */
655 * SET MYRESULTL=                               /* INIT MYRESULTL */
656 * SET ERRMSG =                                /* INIT ERRMSG */
657 * CUTIL00 ECHO MYDATA                          /* ECHO */
658 * SET RC=&LASTCC                               /* RETURN CODE */
659 *
660 *      Results:  RC          =0
661 *                ERRMSG     ='SUCCESSFUL REQUEST'
662 *                MYDATA      ='MY DATA TO DISPLAY.'
663 *                MYDATAAL    =19
664 *                MYRESULT    =' '
665 *                MYRESULTL=0
666 *                CUTIL00 ECHO of MYRESULT =MY DATA TO DISPLAY.
667 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
669 * -----
670 * | Function | VAR1 | VAR2 | Description |
671 * +-----+-----+-----+-----+
672 *14| ECHOQ    | req  | no   | Display variable VAR1 value on screen |
673 * |          |      |      | with content in quotes.              |
674 * |          |      |      |                                     |
675 * -----
676 * SET MYDATA=&STR(my data to display.)      /* INIT MYDATA */
677 * SET MYRESULT=                             /* INIT MYRESULT */
678 * SET MYDATAL =                             /* INIT MYDATAL */
679 * SET MYRESULTL=                             /* INIT MYRESULTL */
680 * SET ERRMSG =                             /* INIT ERRMSG */
681 * CUTIL00 ECHOQ MYDATA                      /* ECHOQ */
682 * SET RC=&LASTCC                            /* RETURN CODE */
683 *
684 * Results: RC =0
685 *           ERRMSG ='SUCCESSFUL REQUEST'
686 *           MYDATA ='MY DATA TO DISPLAY.'
687 *           MYDATAL =19
688 *           MYRESULT =' '
689 *           MYRESULTL=0
690 *           CUTIL00 ECHOQ of MYRESULT ='MY DATA TO DISPLAY.'
691 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
693 * -----
694 * | Function | VAR1 | VAR2 | Description |
695 * +-----+ +-----+ +-----+
696 *15| REVRS    | req  | opt  | Reverse letters and numbers in variable
697 *      |      |      |      | VAR1.
698 *
699 *      |      |      |      | A-Z translate to Z-A
700 *      |      |      |      | a-z translate to z-a
701 *      |      |      |      | 0-9 translate to 9-0
702 *      |      |      |      | Others translate to blank (space)
703 *
704 *      |      |      |      | If VAR2 is omitted, results will be
705 *      |      |      |      | placed into VAR1.
706 * -----
707 * SET MYDATA=&STR(ABCDEFGHJKLM96%QRS)          /* INIT MYDATA */
708 * SET MYRESULT=                                /* INIT MYRESULT */
709 * SET MYDATAL =                                /* INIT MYDATAL */
710 * SET MYRESULTL=                              /* INIT MYRESULTL */
711 * SET ERRMSG =                                /* INIT ERRMSG */
712 * CUTIL00 REVRS MYDATA MYRESULT                /* REVRS */
713 * SET RC=&LASTCC                                /* RETURN CODE */
714 *
715 * Results: RC =0
716 *           ERRMSG ='SUCCESSFUL REQUEST'
717 *           MYDATA ='ABCDEFGHJKLM96%QRS'
718 *           MYDATAL =19
719 *           MYRESULT ='ZYXWVUTSRQPON03 JIH'
720 *           MYRESULTL=19
721 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
723 * -----
724 * | Function | VAR1 | VAR2 | Description |
725 * +-----+ +-----+ +-----+ +-----+
726 *16| UPPERC   | req   | opt   | Translate content to uppercase in
727 *          |       |       |       | variable VAR1.
728 *
729 *          |       |       |       | If VAR2 is omitted, results will be
730 *          |       |       |       | placed into VAR1.
731 * -----
732 * SET MYDATA=&STR(abCdeFgHIJklmnopqrS)          /* INIT MYDATA */
733 * SET MYRESULT=                                /* INIT MYRESULT */
734 * SET MYDATA1 =                                /* INIT MYDATA1 */
735 * SET MYRESULT1=                               /* INIT MYRESULT1 */
736 * SET ERRMSG =                                /* INIT ERRMSG */
737 * CUTIL00 UPPERC MYDATA                        /* UPPERC */
738 * SET RC=&LASTCC                               /* RETURN CODE */
739 * CUTIL00 ECHOQ MYRESULT                       /* ECHOQ */
740 *
741 * Results: RC =0
742 *          ERRMSG ='SUCCESSFUL REQUEST'
743 *          MYDATA ='ABCDEFGHJKLMNOPQRS'
744 *          MYDATA1 =19
745 *          MYRESULT =' '
746 *          MYRESULT1=0
747 *          CUTIL00 ECHOQ of MYRESULT ='ABCDEFGHJKLMNOPQRS'
748 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
750 * -----
751 * | Function | VAR1 | VAR2 | Description |
752 * +-----+-----+-----+
753 *17| LOWERC   | req   | opt   | Translate content to lowercase in
754 *   |         |       |       | variable VAR1.
755 *   |         |       |       |
756 *   |         |       |       | If VAR2 is omitted, results will be
757 *   |         |       |       | placed into VAR1.
758 * -----
759 * SET MYDATA=&STR(abCdeFgHIJklmnopqrS)      /* INIT MYDATA */
760 * SET MYRESULT=                             /* INIT MYRESULT */
761 * SET MYDATAL =                             /* INIT MYDATAL */
762 * SET MYRESULTL=                             /* INIT MYRESULTL */
763 * SET ERRMSG =                             /* INIT ERRMSG */
764 * CUTIL00 LOWERC MYDATA MYRESULT            /* LOWERC */
765 * SET RC=&LASTCC                             /* RETURN CODE */
766 * CUTIL00 ECHOQ MYRESULT                    /* ECHOQ */
767 *
768 * Results: RC =0
769 *           ERRMSG ='SUCCESSFUL REQUEST'
770 *           MYDATA ='ABCDEFGHijklmnopQRS'
771 *           MYDATAL =19
772 *           MYRESULT ='ABCDEFGHijklmnopQRS'
773 *           MYRESULTL=19
774 *           CUTIL00 ECHOQ of MYRESULT ='ABCDEFGHijklmnopQRS'
775 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
777 * -----
778 * | Function | VAR1 | VAR2 | Description |
779 * +-----+-----+-----+-----+
780 *18 | COUNT    | req  | req  | Count occurrences of a search string
781 *   |         |      |      | in variable VAR1.
782 *18 | FINDALL   |      |      |
783 *   | (alias)  |      |      | RC=occurance count
784 *   |         |      |      | occurance count of 0 = NOT FOUND
785 *
786 *   |         |      |      | VAR2 contains Search Params -----
787 *   |         |      |      | Format: "sssssss",bbb,eee
788 *
789 *   |         |      |      | s - search string in double quotes
790 *   |         |      |      | b - optional, begining search position
791 *   |         |      |      | 1-3 numeric digits, default=1
792 *   |         |      |      | default=1
793 *   |         |      |      | e - optional, ending search position
794 *   |         |      |      | 1-3 numeric digits, default=end
795 *   |         |      |      | Params separated by commas (,)
796 *
797 *   |         |      |      | Note: If VAR2 not specified, defaults
798 *   |         |      |      | are applied including search
799 *   |         |      |      | string of one BLANK (" ")
800 * -----
801 * SET MYDATA=&STR(My name is Dino, big Dino.) /* INIT MYDATA */
802 * SET SRCHPRMS = &STR("Dino") /* INIT SRCHPRMS */
803 * SET MYDATAL = /* INIT MYDATAL */
804 * SET SRCHPRMSL= /* INIT SRCHPRMSL */
805 * SET ERRMSG = /* INIT ERRMSG */
806 * CUTIL00 COUNT MYDATA SRCHPRMS /* COUNT */
807 * SET RC=&LASTCC /* RETURN CODE */
808 *
809 * Results: RC =2
810 *          ERRMSG ='SUCCESSFUL REQUEST'
811 *          MYDATA ='MY NAME IS DINO, BIG DINO.'
812 *          MYDATAL =26
813 *          SRCHPRMS =' "DINO" '
814 *          SRCHPRMSL=6
815 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
817 * -----
818 * | Function | VAR1 | VAR2 | Description |
819 * +-----+-----+-----+
820 *19 FIND      req   req   Find starting position of FIRST search
821 *              string occurrence in variable VAR1.
822 *05 INDEX
823 *   (alias)
824 *              RC=position, relative to 1
825 *              position of 0 = NOT FOUND
826 *              VAR2 contains Search Params -----
827 *              Format: "sssssss",bbb,eee
828 *
829 *              s - search string in double quotes
830 *              b - optional, beginning search position
831 *                  1-3 numeric digits, default=1
832 *                  default=1
833 *              e - optional, ending search position
834 *                  1-3 numeric digits, default=end
835 *              Params separated by commas (,):
836 * -----
837 * SET MYDATA=&STR(My name is Dino, big Dino.) /* INIT MYDATA */
838 * SET SRCHPRMS = &STR("Dino") /* INIT SRCHPRMS */
839 * SET MYDATAL = /* INIT MYDATAL */
840 * SET SRCHPRMSL= /* INIT SRCHPRMSL */
841 * SET ERRMSG = /* INIT ERRMSG */
842 * CUTIL00 FIND MYDATA SRCHPRMS /* FIND */
843 * SET RC=&LASTCC /* RETURN CODE */
844 *
845 * Results: RC =12
846 *           ERRMSG ='SUCCESSFUL REQUEST'
847 *           MYDATA ='MY NAME IS DINO, BIG DINO.'
848 *           MYDATAL =26
849 *           SRCHPRMS =' "DINO" '
850 *           SRCHPRMSL=6
851 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
853 * -----
854 * | Function | VAR1 | VAR2 | Description |
855 * +-----+ +-----+ +-----+
856 *20 FINDL      req    req    Find starting position of LAST search
857 *              string occurrence in variable VAR1.
858 *06 INDEXB
859 *      (alias)      RC=position, relative to 1
860 *                  position of 0 = NOT FOUND
861 *
862 *              VAR2 contains Search Params -----
863 *              Format: "sssssss",bbb,eee
864 *
865 *              s - search string in double quotes
866 *              b - optional, beginning search position
867 *                  1-3 numeric digits, default=1
868 *                  default=1
869 *              e - optional, ending search position
870 *                  1-3 numeric digits, default=end
871 *              Params separated by commas (,):
872 * -----
873 * SET MYDATA=&STR(My name is Dino, big Dino.)      /* INIT MYDATA */
874 * SET SRCHPRMS = &STR("Dino")                      /* INIT SRCHPRMS */
875 * SET MYDATAL =                                     /* INIT MYDATAL */
876 * SET SRCHPRMSL=                                    /* INIT SRCHPRMSL */
877 * SET ERRMSG =                                       /* INIT ERRMSG */
878 * CUTIL00 FINDL MYDATA SRCHPRMS                     /* FINDL */
879 * SET RC=&LASTCC                                     /* RETURN CODE */
880 *
881 *      Results:  RC          =22
882 *                ERRMSG     ='SUCCESSFUL REQUEST'
883 *                MYDATA     ='MY NAME IS DINO, BIG DINO.'
884 *                MYDATAL    =26
885 *                SRCHPRMS   =' "DINO" '
886 *                SRCHPRMSL  =6
887 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
889 * -----
890 * | Function | VAR1 | VAR2 | Description |
891 * +-----+-----+-----+
892 *21| CENTER   | req  | opt  | Center content in variable VAR1.
893 *
894 * |          |      |      | If VAR2 is omitted, results will be
895 * |          |      |      | placed into VAR1.
896 * -----
897 * SET MYDATA=&STR(Center this T E X T ... ) /* INIT MYDATA */
898 * SET MYRESULT=                               /* INIT MYRESULT */
899 * SET MYDATAL =                               /* INIT MYDATAL */
900 * SET MYRESULTL=                             /* INIT MYRESULTL */
901 * SET ERRMSG =                               /* INIT ERRMSG */
902 * CUTIL00 CENTER MYDATA MYRESULT             /* CENTER */
903 * SET RC=&LASTCC                             /* RETURN CODE */
904 *
905 * Results: RC =0
906 *           ERRMSG ='SUCCESSFUL REQUEST'
907 *           MYDATA ='CENTER THIS T E X T ...'
908 *           MYDATAL =30
909 *           MYRESULT =' CENTER THIS T E X T ...'
910 *           MYRESULTL=30
911 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
913 * -----
914 * | Function | VAR1 | VAR2 | Description |
915 * +-----+-----+-----+
916 *22| LJUST   | req  | opt  | Left justify content in variable VAR1. |
917 *
918 * |         |      |      | If VAR2 is omitted, results will be |
919 * |         |      |      | placed into VAR1.                   |
920 * -----
921 * SET MYDATA=&STR( JUSTIFY this text ) /* INIT MYDATA */
922 * SET MYRESULT= /* INIT MYRESULT */
923 * SET MYDATAL = /* INIT MYDATAL */
924 * SET MYRESULTL= /* INIT MYRESULTL */
925 * SET ERRMSG = /* INIT ERRMSG */
926 * CUTIL00 LJUST MYDATA MYRESULT /* LJUST */
927 * SET RC=&LASTCC /* RETURN CODE */
928 *
929 * Results: RC =0
930 * ERRMSG ='SUCCESSFUL REQUEST '
931 * MYDATA =' JUSTIFY THIS TEXT '
932 * MYDATAL =22
933 * MYRESULT ='JUSTIFY THIS TEXT '
934 * MYRESULTL=22
935 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
937 * -----
938 * | Function | VAR1 | VAR2 | Description |
939 * +-----+-----+-----+
940 *23| RJUST   | req  | opt  | Right justify content in variable VAR1. |
941 * |         |      |      | If VAR2 is omitted, results will be |
942 * |         |      |      | placed into VAR1. |
943 * -----
944 *
945 * SET MYDATA=&STR( JUSTIFY this text ) /* INIT MYDATA */
946 * SET MYRESULT= /* INIT MYRESULT */
947 * SET MYDATAL = /* INIT MYDATAL */
948 * SET MYRESULTL= /* INIT MYRESULTL */
949 * SET ERRMSG = /* INIT ERRMSG */
950 * CUTIL00 RJUST MYDATA MYRESULT /* RJUST */
951 * SET RC=&LASTCC /* RETURN CODE */
952 * /*
953 *
954 * Results: RC =0
955 * ERRMSG ='SUCCESSFUL REQUEST '
956 * MYDATA =' JUSTIFY THIS TEXT '
957 * MYDATAL =22
958 * MYRESULT =' JUSTIFY THIS TEXT'
959 * MYRESULTL=22
960 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
962 * -----
963 * | Function | VAR1 | VAR2 | Description |
964 * +-----+-----+-----+
965 *24| ZFILL    | req   | opt   | Zero fill after content is right
966 *      |          |       |       | justified in variable VAR1.
967 *      |          |       |       |
968 *      |          |       |       | If VAR2 is omitted, results will be
969 *      |          |       |       | placed into VAR1.
970 * -----
971 * SET MYDATA=&STR(12    )          /* INIT MYDATA */
972 * SET MYRESULT=                  /* INIT MYRESULT */
973 * SET MYDATA1 =                  /* INIT MYDATA1 */
974 * SET MYRESULT1=                 /* INIT MYRESULT1 */
975 * SET ERRMSG =                   /* INIT ERRMSG */
976 * CUTIL00 ZFILL MYDATA MYRESULT  /* ZFILL */
977 * SET RC=&LASTCC                 /* RETURN CODE */
978 *
979 *      Results: RC          =0
980 *                ERRMSG    ='SUCCESSFUL REQUEST'
981 *                MYDATA     ='12'
982 *                MYDATA1    =5
983 *                MYRESULT   ='00012'
984 *                MYRESULT1  =5
985 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```

987 * -----
988 * | Function | VAR1 | VAR2 | Description |
989 * +-----+-----+-----+
990 *25| WORDS    | req  | no   | Number of words (contiguous non-blank
991 *   |         |      |      | groups) in variable VAR1.
992 *   |         |      |      |
993 *   |         |      |      | RC=count
994 * -----
995 * SET MYDATA=&STR( one two three33 four., five ) /* INIT MYDATA */
996 * SET MYRESULT=                                  /* INIT MYRESULT */
997 * SET MYDATAL =                                  /* INIT MYDATAL */
998 * SET MYRESULTL=                                /* INIT MYRESULTL */
999 * SET ERRMSG =                                   /* INIT ERRMSG */
1000 * CUTIL00 WORDS MYDATA                          /* WORDS */
1001 * SET RC=&LASTCC                                /* RETURN CODE */
1002 *
1003 * Results: RC =5
1004 *          ERRMSG ='SUCCESSFUL REQUEST '
1005 *          MYDATA =' ONE TWO THREE33 FOUR., FIVE '
1006 *          MYDATAL =30
1007 *          MYRESULT =' '
1008 *          MYRESULTL=0
1009 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1011 * -----
1012 * | Function | VAR1 | VAR2 | Description |
1013 * +-----+-----+-----+
1014 *26| GEN#      | req  | no   | Generate a 3-digit number and return |
1015 * |          |      |      | in variable VAR1.                    |
1016 * -----
1017 * SET MYDATA=0                      /* INIT MYDATA */
1018 * SET MYRESULT=                    /* INIT MYRESULT */
1019 * SET MYDATAL =                    /* INIT MYDATAL */
1020 * SET MYRESULTL=                  /* INIT MYRESULTL */
1021 * SET ERRMSG =                    /* INIT ERRMSG */
1022 * CUTIL00 GEN# MYDATA              /* GEN# */
1023 * SET RC=&LASTCC                  /* RETURN CODE */
1024 *
1025 *      Results:  RC          =0
1026 *                ERRMSG     ='SUCCESSFUL REQUEST'
1027 *                MYDATA     ='515'
1028 *                MYDATAL    =3
1029 *                MYRESULT   =' '
1030 *                MYRESULTL  =0
1031 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1033 * -----
1034 * | Function | VAR1 | VAR2 | Description |
1035 * +-----+ +-----+ +-----+ +-----+
1036 *27| DD2DSN   | req  | req  | Dataset name associated with DD name. |
1037 *
1038 * |          |      |      | RC=8 DD not found |
1039 *
1040 * |          |      |      | VAR1 contains DD name |
1041 * |          |      |      | note: Case-sensitive, upper case |
1042 * |          |      |      | VAR2 contains Dataset name as result |
1043 * -----
1044 * SET MYDATA=&STR(TEMPJCL)                /* INIT MYDATA */
1045 * SET MYRESULT=                          /* INIT MYRESULT */
1046 * SET MYDATAL =                          /* INIT MYDATAL */
1047 * SET MYRESULTL=                        /* INIT MYRESULTL */
1048 * SET ERRMSG =                          /* INIT ERRMSG */
1049 * CUTIL00 DD2DSN MYDATA MYRESULT          /* DD2DSN */
1050 * SET RC=&LASTCC                          /* RETURN CODE */
1051 *
1052 *      Results:  RC          =0
1053 *                ERRMSG     ='SUCCESSFUL REQUEST'
1054 *                MYDATA     ='SYSPROC'
1055 *                MYDATAL    =7
1056 *                MYRESULT   ='SYS2.CMDPROC'
1057 *                MYRESULTL=12
1058 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1060 * -----
1061 * | Function | VAR1 | VAR2 | Description |
1062 * +-----+ +-----+ +-----+ +-----+
1063 *28| JOBINFO | req  | no   | Return JOB INFORMATION in variable
1064 *      |      |      |      | VAR1 as follows:
1065 *      |      |      |      |
1066 *      |      |      |      | VAR1 Pos      Batch TSO      Online TSO
1067 *      |      |      |      | -----
1068 *      |      |      |      | byte 01-08     JOB NAME      TSO Userid
1069 *      |      |      |      | byte 09-16     JOB PROC      TSO PROC
1070 *      |      |      |      | byte 17-24     JOB STEP      TSO LOGON PROC
1071 * -----
1072 * SET MYDATA=                                /* INIT MYDATA */
1073 * SET MYRESULT=                              /* INIT MYRESULT */
1074 * SET MYDATAAL =                            /* INIT MYDATAAL */
1075 * SET MYRESULTL=                          /* INIT MYRESULTL */
1076 * SET ERRMSG =                             /* INIT ERRMSG */
1077 * CUTIL00 JOBINFO MYDATA                    /* JOBINFO */
1078 * SET RC=&LASTCC                           /* RETURN CODE */
1079 *
1080 *      Results:  RC          =0
1081 *                ERRMSG     ='SUCCESSFUL REQUEST'
1082 *                MYDATA      ='HERC01 IKJACCNTTSOLOGON'
1083 *                MYDATAAL    =24
1084 *                MYRESULT    =' '
1085 *                MYRESULTL   =0
1086 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1088 * -----
1089 * | Function | VAR1 | VAR2 | Description |
1090 * +-----+-----+-----+
1091 *29| DAYSMM   | req  | no   | Number of days in month (MMCCYY).
1092 * |          |      |      |
1093 * |          |      |      | RC=Number of days (1-31)
1094 * |          |      |      |
1095 * |          |      |      | VAR1 contains month in the
1096 * |          |      |      | format of MMCCYY.
1097 * |          |      |      |
1098 * |          |      |      | Leap year is considered.
1099 * |          |      |      |
1100 * |          |      |      | For example, if MM is FEB and CCYY is
1101 * |          |      |      | a leap year, number of days are 29,
1102 * |          |      |      | else, 28.
1103 * -----
1104 * SET MYDATA=&STR(022016)                /* INIT MYDATA */
1105 * SET MYRESULT=                          /* INIT MYRESULT */
1106 * SET MYDATAL =                          /* INIT MYDATAL */
1107 * SET MYRESULTL=                        /* INIT MYRESULTL */
1108 * SET ERRMSG =                          /* INIT ERRMSG */
1109 * CUTIL00 DAYSMM MYDATA                  /* DAYSMM */
1110 * SET RC=&LASTCC                          /* RETURN CODE */
1111 *
1112 *      Results:  RC          =29
1113 *                ERRMSG     ='SUCCESSFUL REQUEST'
1114 *                MYDATA     ='022016'
1115 *                MYDATAL    =6
1116 *                MYRESULT   =' '
1117 *                MYRESULTL=
1118 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1120 * -----
1121 * | Function | VAR1 | VAR2 | Description |
1122 * +-----+ +-----+ +-----+ +-----+
1123 *30 | DAYSYY   | req   | no    | Number of days in year (CCYY).
1124 * |          |       |       | RC=Number of days (365 or 366)
1125 * |          |       |       |
1126 * |          |       |       | VAR1 contains centry and year in the
1127 * |          |       |       | format of CCYY.
1128 * |          |       |       |
1129 * |          |       |       | Leap year is considered.
1130 * |          |       |       |
1131 * |          |       |       | For example, if CCYY is a leap year,
1132 * |          |       |       | number of days is 366, else 365.
1133 * |          |       |       |
1134 * -----
1135 * SET MYDATA=&STR(2020)                /* INIT MYDATA */
1136 * SET MYRESULT=                      /* INIT MYRESULT */
1137 * SET MYDATA1 =                      /* INIT MYDATA1 */
1138 * SET MYRESULT1=                     /* INIT MYRESULT1 */
1139 * SET ERRMSG =                      /* INIT ERRMSG */
1140 * CUTIL00 DAYSYY MYDATA              /* DAYSYY */
1141 * SET RC=&LASTCC                     /* RETURN CODE */
1142 *
1143 * Results: RC =366
1144 *          ERRMSG ='SUCCESSFUL REQUEST'
1145 *          MYDATA ='2020'
1146 *          MYDATA1 =4
1147 *          MYRESULT =' '
1148 *          MYRESULT1=
1149 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1151 * -----
1152 * | Function | VAR1 | VAR2 | Description |
1153 * +-----+ +-----+ +-----+ +-----+
1154 *31| ISLEAP   | req  | no   | Test for leap year of variable VAR1
1155 *      |      |      |      | in the format (CCYY) returning a
1156 *      |      |      |      | true-false indication.
1157 *
1158 *      |      |      |      | RC=4000  &ERRMSG='FALSE'
1159 *      |      |      |      | RC=4001  &ERRMSG='TRUE'
1160 * -----
1161 * SET MYDATA=&STR(2020)                                /* INIT MYDATA */
1162 * SET MYRESULT=                                         /* INIT MYRESULT */
1163 * SET MYDATAL =                                         /* INIT MYDATAL */
1164 * SET MYRESULTL=                                       /* INIT MYRESULTL */
1165 * SET ERRMSG =                                         /* INIT ERRMSG */
1166 * CUTIL00 ISLEAP MYDATA                                /* ISLEAP */
1167 * SET RC=&LASTCC                                       /* RETURN CODE */
1168 *
1169 *      Results:  RC          =4001
1170 *                ERRMSG     ='TRUE'
1171 *                MYDATA      ='2020'
1172 *                MYDATAL     =4
1173 *                MYRESULT    =' '
1174 *                MYRESULTL=
1175 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1177 * -----
1178 * | Function | VAR1 | VAR2 | Description |
1179 * +-----+-----+-----+
1180 *32| CYJ-D8   | req  | opt  | Convert date in variable VAR1 in
1181 *      |          |      |      | format CCYYJJJ to monthname DD, CCYY
1182 *      |          |      |      | where:
1183 *      |          |      |      | monthname - Full month name (see below)
1184 *      |          |      |      | DD        - 2 digit date (01-31)
1185 *      |          |      |      | CCYY      - 4 digit century and year
1186 *
1187 *      |          |      |      | Monthnames:
1188 *      |          |      |      | 01-January 02-February 03-March
1189 *      |          |      |      | 04-April   05-May      06-June
1190 *      |          |      |      | 07-July    08-August   09-September
1191 *      |          |      |      | 10-October 11-November 12-December
1192 *
1193 *      |          |      |      | If VAR2 is omitted, results will be
1194 *      |          |      |      | placed into VAR1.
1195 * -----
1196 * SET MYDATA=&STR(2020061)                /* INIT MYDATA */
1197 * SET MYRESULT=                          /* INIT MYRESULT */
1198 * SET MYDATAL =                          /* INIT MYDATAL */
1199 * SET MYRESULTL=                        /* INIT MYRESULTL */
1200 * SET ERRMSG =                          /* INIT ERRMSG */
1201 * CUTIL00 CYJ-D8 MYDATA MYRESULT        /* CYJ-D8 */
1202 * SET RC=&LASTCC                        /* RETURN CODE */
1203 *
1204 * Results: RC =0
1205 *          ERRMSG ='SUCCESSFUL REQUEST'
1206 *          MYDATA  ='2020061'
1207 *          MYDATAL  =7
1208 *          MYRESULT ='MARCH 01, 2020'
1209 *          MYRESULTL=14
1210 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1212 * -----
1213 * | Function | VAR1 | VAR2 | Description |
1214 * +-----+ +-----+ +-----+ +-----+
1215 *33| CYJ-DAY | req | opt | Convert date in variable VAR1 in
1216 *          |          |          |          | format CCYYJJJ to weekday name.
1217 *          |          |          |          |
1218 *          |          |          |          | Weekday Name:
1219 *          |          |          |          | Sunday, Monday, Tuesday, Wednesday
1220 *          |          |          |          | Thursday, Friday, Saturday
1221 *          |          |          |          |
1222 *          |          |          |          | If VAR2 is omitted, results will be
1223 *          |          |          |          | placed into VAR1.
1224 * -----
1225 * SET MYDATA=&STR(1981061)                /* INIT MYDATA */
1226 * SET MYRESULT=                          /* INIT MYRESULT */
1227 * SET MYDATA =                          /* INIT MYDATA */
1228 * SET MYRESULTL=                        /* INIT MYRESULTL */
1229 * SET ERRMSG =                          /* INIT ERRMSG */
1230 * CUTIL00 CYJ-DAY MYDATA MYRESULT        /* CYJ-DAY */
1231 * SET RC=&LASTCC                          /* RETURN CODE */
1232 *
1233 *      Results:  RC          =0
1234 *                ERRMSG    ='SUCCESSFUL REQUEST'
1235 *                MYDATA    ='1981061'
1236 *                MYDATA    =7
1237 *                MYRESULT  ='MONDAY'
1238 *                MYRESULTL=6
1239 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1241 * -----
1242 * | Function | VAR1 | VAR2 | Description |
1243 * +-----+ +-----+ +-----+ +-----+
1244 *34| CYJ-DOW  | req  | opt  | Convert date in variable VAR1 in
1245 *          |      |      |      | format CCYYJJJ to weekday number.
1246 *          |      |      |      |
1247 *          |      |      |      | Weekday Number:
1248 *          |      |      |      | 0-6
1249 *          |      |      |      |
1250 *          |      |      |      | Weekday Number to Weekday Name:
1251 *          |      |      |      | 0-Sunday    1-Monday    2-Tuesday
1252 *          |      |      |      | 3-Wednesday 4-Thursday   5-Friday
1253 *          |      |      |      | 6-Saturday
1254 *          |      |      |      |
1255 *          |      |      |      | If VAR2 is omitted, results will be
1256 *          |      |      |      | placed into VAR1.
1257 * -----
1258 * SET MYDATA=&STR(1980061)                /* INIT MYDATA */
1259 * SET MYRESULT=                          /* INIT MYRESULT */
1260 * SET MYDATAL =                          /* INIT MYDATAL */
1261 * SET MYRESULTL=                        /* INIT MYRESULTL */
1262 * SET ERRMSG =                          /* INIT ERRMSG */
1263 * CUTIL00 CYJ-DOW MYDATA MYRESULT        /* CYJ-DOW */
1264 * SET RC=&LASTCC                          /* RETURN CODE */
1265 *
1266 *      Results:  RC          =0
1267 *                ERRMSG     ='SUCCESSFUL REQUEST'
1268 *                MYDATA     ='1980061'
1269 *                MYDATAL    =7
1270 *                MYRESULT   ='6'
1271 *                MYRESULTL  =1
1272 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1274 * -----
1275 * | Function | VAR1 | VAR2 | Description |
1276 * +-----+ +-----+ +-----+ +-----+
1277 *35| CYJ-MDCY*| req  | opt  | Convert date in variable VAR1 in
1278 *   |         |      |      | format CCYYJJJ to MMDDCCYY.
1279 *
1280 *   *MDCY can
1281 *   be in any
1282 *   order
1283 *
1284 *   MM - 2 digit month (01-12)
1285 *   DD - 2 digit date  (01-31)
1286 *   CC - 2 digit century
1287 *   YY - 2 digit year
1288 *
1289 *   If VAR2 is omitted, results will be
1290 *   placed into VAR1.
1291 * -----
1292 * SET MYDATA=&STR(2020061)                /* INIT MYDATA */
1293 * SET MYRESULT=                          /* INIT MYRESULT */
1294 * SET MYDATA1 =                          /* INIT MYDATA1 */
1295 * SET MYRESULT1=                        /* INIT MYRESULT1 */
1296 * SET ERRMSG =                          /* INIT ERRMSG */
1297 * CUTIL00 CYJ-MDCY MYDATA MYRESULT      /* CYJ-MDCY */
1298 * SET RC=&LASTCC                        /* RETURN CODE */
1299 *
1300 * Results: RC =0
1301 *          ERRMSG ='SUCCESSFUL REQUEST'
1302 *          MYDATA ='2020061'
1303 *          MYDATA1 =7
1304 *          MYRESULT ='03012020'
1305 *          MYRESULT1=8
1306 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1308 * -----
1309 * | Function | VAR1 | VAR2 | Description |
1310 * +-----+-----+-----+
1311 *36| JCY-D8   | req  | opt  | Convert date in variable VAR1 in
1312 *      |          |      |      | format JJCCYY to monthname DD, CCYY
1313 *      |          |      |      | where:
1314 *      |          |      |      | monthname - Full month name (see below)
1315 *      |          |      |      | DD        - 2 digit date (01-31)
1316 *      |          |      |      | CCYY      - 4 digit century and year
1317 *
1318 *      |          |      |      | Monthnames:
1319 *      |          |      |      | 01-January 02-February 03-March
1320 *      |          |      |      | 04-April   05-May      06-June
1321 *      |          |      |      | 07-July    08-August   09-September
1322 *      |          |      |      | 10-October 11-November 12-December
1323 *
1324 *      |          |      |      | If VAR2 is omitted, results will be
1325 *      |          |      |      | placed into VAR1.
1326 * -----
1327 * SET MYDATA=&STR(1612017)          /* INIT MYDATA */
1328 * SET MYRESULT=                    /* INIT MYRESULT */
1329 * SET MYDATAL =                    /* INIT MYDATAL */
1330 * SET MYRESULTL=                   /* INIT MYRESULTL */
1331 * SET ERRMSG =                     /* INIT ERRMSG */
1332 * CUTIL00 JCY-D8 MYDATA MYRESULT   /* JCY-D8 */
1333 * SET RC=&LASTCC                    /* RETURN CODE */
1334 *
1335 * Results: RC =0
1336 *          ERRMSG ='SUCCESSFUL REQUEST'
1337 *          MYDATA  ='1612017'
1338 *          MYDATAL =7
1339 *          MYRESULT ='JUNE 10, 2017'
1340 *          MYRESULTL=13
1341 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1343 * -----
1344 * | Function | VAR1 | VAR2 | Description |
1345 * +-----+ +-----+ +-----+ +-----+
1346 *37| JCY-DAY | req | opt | Convert date in variable VAR1 in
1347 * | | | | format JJJCCYY to weekday name.
1348 * | | | |
1349 * | | | | Weekday Name:
1350 * | | | | Sunday, Monday, Tuesday, Wednesday
1351 * | | | | Thursday, Friday, Saturday
1352 * | | | |
1353 * | | | | If VAR2 is omitted, results will be
1354 * | | | | placed into VAR1.
1355 * -----
1356 * SET MYDATA=&STR(0611981) /* INIT MYDATA */
1357 * SET MYRESULT= /* INIT MYRESULT */
1358 * SET MYDATAL = /* INIT MYDATAL */
1359 * SET MYRESULTL= /* INIT MYRESULTL */
1360 * SET ERRMSG = /* INIT ERRMSG */
1361 * CUTIL00 JCY-DAY MYDATA MYRESULT /* JCY-DAY */
1362 * SET RC=&LASTCC /* RETURN CODE */
1363 *
1364 * Results: RC =0
1365 * ERRMSG ='SUCCESSFUL REQUEST'
1366 * MYDATA ='0611981'
1367 * MYDATAL =7
1368 * MYRESULT ='MONDAY'
1369 * MYRESULTL=6
1370 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1372 * -----
1373 * | Function | VAR1 | VAR2 | Description |
1374 * +-----+ +-----+ +-----+ +-----+
1375 *38 | JCY-DOW  | req  | opt  | Convert date in variable VAR1 in
1376 *   |         |      |      | format JJCCYY to weekday number.
1377 *   |         |      |      |
1378 *   |         |      |      | Weekday Number:
1379 *   |         |      |      | 0-6
1380 *   |         |      |      |
1381 *   |         |      |      | Weekday Number to Weekday Name:
1382 *   |         |      |      | 0-Sunday    1-Monday    2-Tuesday
1383 *   |         |      |      | 3-Wednesday 4-Thursday  5-Friday
1384 *   |         |      |      | 6-Saturday
1385 *   |         |      |      |
1386 *   |         |      |      | If VAR2 is omitted, results will be
1387 *   |         |      |      | placed into VAR1.
1388 * -----
1389 * SET MYDATA=&STR(0611981)                /* INIT MYDATA */
1390 * SET MYRESULT=                          /* INIT MYRESULT */
1391 * SET MYDATAL =                          /* INIT MYDATAL */
1392 * SET MYRESULTL=                        /* INIT MYRESULTL */
1393 * SET ERRMSG =                          /* INIT ERRMSG */
1394 * CUTIL00 JCY-DOW MYDATA MYRESULT        /* JCY-DOW */
1395 * SET RC=&LASTCC                        /* RETURN CODE */
1396 *
1397 * Results: RC =0
1398 *          ERRMSG ='SUCCESSFUL REQUEST'
1399 *          MYDATA ='0611981'
1400 *          MYDATAL =7
1401 *          MYRESULT ='1'
1402 *          MYRESULTL=1
1403 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1405 * -----
1406 * | Function | VAR1 | VAR2 | Description |
1407 * +-----+ +-----+ +-----+ +-----+
1408 *39| JCY-MDCY* | req  | opt  | Convert date in variable VAR1 in
1409 *      |      |      |      | format JJJCCYY to MMDDCCYY.
1410 *
1411 *      | *MDCY can |      |      | For example-
1412 *      | be in any  |      |      | JCY-MDY  convert JJJCCYY to MMDDYY
1413 *      | order   |      |      | JCY-CYDM convert JJJCCYY to CCYYDDMM
1414 *
1415 *      |      |      |      | MM - 2 digit month (01-12)
1416 *      |      |      |      | DD - 2 digit date  (01-31)
1417 *      |      |      |      | CC - 2 digit century
1418 *      |      |      |      | YY - 2 digit year
1419 *
1420 *      |      |      |      | If VAR2 is omitted, results will be
1421 *      |      |      |      | placed into VAR1.
1422 * -----
1423 * SET MYDATA=&STR(1612017)                /* INIT MYDATA */
1424 * SET MYRESULT=                          /* INIT MYRESULT */
1425 * SET MYDATA1 =                          /* INIT MYDATA1 */
1426 * SET MYRESULT1=                        /* INIT MYRESULT1 */
1427 * SET ERRMSG =                          /* INIT ERRMSG */
1428 * CUTIL00 JCY-MDCY MYDATA MYRESULT      /* JCY-MDCY */
1429 * SET RC=&LASTCC                        /* RETURN CODE */
1430 *
1431 *      Results:  RC          =0
1432 *                ERRMSG    ='SUCCESSFUL REQUEST'
1433 *                MYDATA    ='1612017'
1434 *                MYDATA1   =7
1435 *                MYRESULT  ='06102017'
1436 *                MYRESULT1=8
1437 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1439 * -----
1440 * | Function | VAR1 | VAR2 | Description |
1441 * +-----+-----+-----+
1442 *40 | MDCY-D8 | req | opt | Convert date in variable VAR1 in
1443 * | | | | format MMDDCCYY to monthname DD, CCYY
1444 * | | | | where:
1445 * | | | | monthname - Full month name (see below)
1446 * | | | | DD - 2 digit date (01-31)
1447 * | | | | CCYY - 4 digit century and year
1448 * | | | |
1449 * | | | | Monthnames:
1450 * | | | | 01-January 02-February 03-March
1451 * | | | | 04-April 05-May 06-June
1452 * | | | | 07-July 08-August 09-September
1453 * | | | | 10-October 11-November 12-December
1454 * | | | |
1455 * | | | | If VAR2 is omitted, results will be
1456 * | | | | placed into VAR1.
1457 * -----
1458 * SET MYDATA=&STR(12101990) /* INIT MYDATA */
1459 * SET MYRESULT= /* INIT MYRESULT */
1460 * SET MYDATAL = /* INIT MYDATAL */
1461 * SET MYRESULTL= /* INIT MYRESULTL */
1462 * SET ERRMSG = /* INIT ERRMSG */
1463 * CUTIL00 MDCY-D8 MYDATA MYRESULT /* MDCY-D8 */
1464 * SET RC=&LASTCC /* RETURN CODE */
1465 *
1466 * Results: RC =0
1467 * ERRMSG ='SUCCESSFUL REQUEST'
1468 * MYDATA ='12101990'
1469 * MYDATAL =8
1470 * MYRESULT ='DECEMBER 10, 1990'
1471 * MYRESULTL=17
1472 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1474 * -----
1475 * | Function | VAR1 | VAR2 | Description |
1476 * +-----+ +-----+ +-----+ +-----+
1477 *41| MDCY-DAY | req  | opt  | Convert date in variable VAR1 in
1478 *      |      |      |      | format MMDDCCYY to weekday name.
1479 *
1480 *      |      |      |      | Weekday Name:
1481 *      |      |      |      | Sunday, Monday, Tuesday, Wednesday
1482 *      |      |      |      | Thursday, Friday, Saturday
1483 *
1484 *      |      |      |      | If VAR2 is omitted, results will be
1485 *      |      |      |      | placed into VAR1.
1486 * -----
1487 * SET MYDATA=&STR(12101990)                /* INIT MYDATA */
1488 * SET MYRESULT=                            /* INIT MYRESULT */
1489 * SET MYDATAL =                            /* INIT MYDATAL */
1490 * SET MYRESULTL=                          /* INIT MYRESULTL */
1491 * SET ERRMSG =                            /* INIT ERRMSG */
1492 * CUTIL00 MDCY-DAY MYDATA MYRESULT        /* MDCY-DAY */
1493 * SET RC=&LASTCC                          /* RETURN CODE */
1494 *
1495 *      Results:  RC          =0
1496 *                ERRMSG    ='SUCCESSFUL REQUEST'
1497 *                MYDATA    ='12101990'
1498 *                MYDATAL   =8
1499 *                MYRESULT  ='MONDAY'
1500 *                MYRESULTL=6
1501 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1503 * -----
1504 * | Function | VAR1 | VAR2 | Description |
1505 * +-----+ +-----+ +-----+ +-----+
1506 *42| MDCY-DOW | req  | opt  | Convert date in variable VAR1 in
1507 *      |      |      |      | format MMDDCCYY to weekday number.
1508 *
1509 *      |      |      |      | Weekday Number:
1510 *      |      |      |      | 0-6
1511 *
1512 *      |      |      |      | Weekday Number to Weekday Name:
1513 *      |      |      |      | 0-Sunday    1-Monday    2-Tuesday
1514 *      |      |      |      | 3-Wednesday 4-Thursday   5-Friday
1515 *      |      |      |      | 6-Saturday
1516 *
1517 *      |      |      |      | If VAR2 is omitted, results will be
1518 *      |      |      |      | placed into VAR1.
1519 * -----
1520 * SET MYDATA=&STR(12101990)                /* INIT MYDATA */
1521 * SET MYRESULT=                          /* INIT MYRESULT */
1522 * SET MYDATAL =                          /* INIT MYDATAL */
1523 * SET MYRESULTL=                        /* INIT MYRESULTL */
1524 * SET ERRMSG =                          /* INIT ERRMSG */
1525 * CUTIL00 MDCY-DOW MYDATA MYRESULT        /* MDCY-DOW */
1526 * SET RC=&LASTCC                          /* RETURN CODE */
1527 *
1528 *      Results: RC          =0
1529 *                ERRMSG    ='SUCCESSFUL REQUEST'
1530 *                MYDATA    ='12101990'
1531 *                MYDATAL   =8
1532 *                MYRESULT  ='1'
1533 *                MYRESULTL=1
1534 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1536 * -----
1537 * | Function | VAR1 | VAR2 | Description |
1538 * +-----+ +-----+ +-----+ +-----+
1539 *43 | MDCY-CYJ* | req  | opt  | Convert date in variable VAR1 in
1540 *   |         |      |      | format MMDDCCYY to CCYYJJJ.
1541 *
1542 *   | *CYJ can |      |      | For example-
1543 *   | be in any |      |      | MDCY-YJ  convert MMDDCCYY to YYJJJ
1544 *   | order    |      |      | MDCY-J   convert MMDDCCYY to JJJ
1545 *
1546 *   |         |      |      | JJJ- 3 digit julian day (1-366)
1547 *   |         |      |      | CC - 2 digit century
1548 *   |         |      |      | YY - 2 digit year
1549 *
1550 *   |         |      |      | If VAR2 is omitted, results will be
1551 *   |         |      |      | placed into VAR1.
1552 * -----
1553 * SET MYDATA=&STR(12101990)                /* INIT MYDATA */
1554 * SET MYRESULT=                          /* INIT MYRESULT */
1555 * SET MYDATAL =                          /* INIT MYDATAL */
1556 * SET MYRESULTL=                        /* INIT MYRESULTL */
1557 * SET ERRMSG =                          /* INIT ERRMSG */
1558 * CUTIL00 MDCY-CYJ MYDATA MYRESULT        /* MDCY-CYJ */
1559 * SET RC=&LASTCC                          /* RETURN CODE */
1560 *
1561 * Results: RC =0
1562 *          ERRMSG ='SUCCESSFUL REQUEST'
1563 *          MYDATA ='12101990'
1564 *          MYDATAL =8
1565 *          MYRESULT ='1990344'
1566 *          MYRESULTL=7
1567 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1569 * -----
1570 * | Function | VAR1 | VAR2 | Description |
1571 * +-----+-----+-----+
1572 *44| DMCY-D8 | req | opt | Convert date in variable VAR1 in
1573 * | | | | format DDMMCCYY to monthname DD, CCYY
1574 * | | | | where:
1575 * | | | | monthname - Full month name (see below)
1576 * | | | | DD - 2 digit date (01-31)
1577 * | | | | CCYY - 4 digit century and year
1578 * | | | |
1579 * | | | | Monthnames:
1580 * | | | | 01-January 02-February 03-March
1581 * | | | | 04-April 05-May 06-June
1582 * | | | | 07-July 08-August 09-September
1583 * | | | | 10-October 11-November 12-December
1584 * | | | |
1585 * | | | | If VAR2 is omitted, results will be
1586 * | | | | placed into VAR1.
1587 * -----
1588 * SET MYDATA=&STR(15071987) /* INIT MYDATA */
1589 * SET MYRESULT= /* INIT MYRESULT */
1590 * SET MYDATAL = /* INIT MYDATAL */
1591 * SET MYRESULTL= /* INIT MYRESULTL */
1592 * SET ERRMSG = /* INIT ERRMSG */
1593 * CUTIL00 DMCY-D8 MYDATA MYRESULT /* DMCY-D8 */
1594 * SET RC=&LASTCC /* RETURN CODE */
1595 *
1596 * Results: RC =0
1597 * ERRMSG ='SUCCESSFUL REQUEST'
1598 * MYDATA ='15071987'
1599 * MYDATAL =8
1600 * MYRESULT ='JULY 15, 1987'
1601 * MYRESULTL=13
1602 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1604 * -----
1605 * | Function | VAR1 | VAR2 | Description |
1606 * +-----+ +-----+ +-----+ +-----+
1607 *45| DMCY-DAY | req  | opt  | Convert date in variable VAR1 in
1608 *      |      |      |      | format DDMMCCYY to weekday name.
1609 *
1610 *      |      |      |      | Weekday Name:
1611 *      |      |      |      | Sunday, Monday, Tuesday, Wednesday
1612 *      |      |      |      | Thursday, Friday, Saturday
1613 *
1614 *      |      |      |      | If VAR2 is omitted, results will be
1615 *      |      |      |      | placed into VAR1.
1616 * -----
1617 * SET MYDATA=&STR(15071987)                /* INIT MYDATA */
1618 * SET MYRESULT=                          /* INIT MYRESULT */
1619 * SET MYDATA1 =                          /* INIT MYDATA1 */
1620 * SET MYRESULT1=                        /* INIT MYRESULT1 */
1621 * SET ERRMSG =                          /* INIT ERRMSG */
1622 * CUTIL00 DMCY-DAY MYDATA MYRESULT        /* DMCY-DAY */
1623 * SET RC=&LASTCC                          /* RETURN CODE */
1624 *
1625 *      Results:  RC          =0
1626 *                  ERRMSG    ='SUCCESSFUL REQUEST'
1627 *                  MYDATA     ='15071987'
1628 *                  MYDATA1    =8
1629 *                  MYRESULT    ='WEDNESDAY'
1630 *                  MYRESULT1=9
1631 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1633 * -----
1634 * | Function | VAR1 | VAR2 | Description |
1635 * +-----+ +-----+ +-----+ +-----+
1636 *46 | DMCY-DOW | req  | opt  | Convert date in variable VAR1 in
1637 *   |         |      |      | format DDMCCYY to weekday number.
1638 *   |         |      |      |
1639 *   |         |      |      | Weekday Number:
1640 *   |         |      |      | 0-6
1641 *   |         |      |      |
1642 *   |         |      |      | Weekday Number to Weekday Name:
1643 *   |         |      |      | 0-Sunday    1-Monday    2-Tuesday
1644 *   |         |      |      | 3-Wednesday 4-Thursday  5-Friday
1645 *   |         |      |      | 6-Saturday
1646 *   |         |      |      |
1647 *   |         |      |      | If VAR2 is omitted, results will be
1648 *   |         |      |      | placed into VAR1.
1649 * -----
1650 * SET MYDATA=&STR(15071987)                /* INIT MYDATA */
1651 * SET MYRESULT=                          /* INIT MYRESULT */
1652 * SET MYDATAL =                          /* INIT MYDATAL */
1653 * SET MYRESULTL=                        /* INIT MYRESULTL */
1654 * SET ERRMSG =                          /* INIT ERRMSG */
1655 * CUTIL00 DMCY-DOW MYDATA MYRESULT        /* DMCY-DOW */
1656 * SET RC=&LASTCC                          /* RETURN CODE */
1657 *
1658 * Results: RC =0
1659 *          ERRMSG ='SUCCESSFUL REQUEST'
1660 *          MYDATA ='15071987'
1661 *          MYDATAL =8
1662 *          MYRESULT ='3'
1663 *          MYRESULTL=1
1664 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1666 * -----
1667 * | Function | VAR1 | VAR2 | Description |
1668 * +-----+ +-----+ +-----+ +-----+
1669 *47| DMCY-CYJ*| req  | opt  | Convert date in variable VAR1 in
1670 *      |      |      |      | format DDMMCCYY to CCYYJJJ.
1671 *
1672 *      |      |      |      | For example-
1673 *      |      |      |      | DMCY-YJ  convert DDMMCCYY to YYJJJ
1674 *      |      |      |      | DMCY-J   convert DDMMCCYY to JJJ
1675 *
1676 *      |      |      |      | JJJ- 3 digit julian day (1-366)
1677 *      |      |      |      | CC - 2 digit century
1678 *      |      |      |      | YY - 2 digit year
1679 *
1680 *      |      |      |      | If VAR2 is omitted, results will be
1681 *      |      |      |      | placed into VAR1.
1682 * -----
1683 * SET MYDATA=&STR(15071987)          /* INIT MYDATA */
1684 * SET MYRESULT=                     /* INIT MYRESULT */
1685 * SET MYDATAL =                     /* INIT MYDATAL */
1686 * SET MYRESULTL=                    /* INIT MYRESULTL */
1687 * SET ERRMSG =                      /* INIT ERRMSG */
1688 * CUTIL00 DMCY-CYJ MYDATA MYRESULT /* DMCY-CYJ */
1689 * SET RC=&LASTCC                     /* RETURN CODE */
1690 *
1691 *      Results:  RC          =0
1692 *                ERRMSG     ='SUCCESSFUL REQUEST'
1693 *                MYDATA      ='15071987'
1694 *                MYDATAL     =8
1695 *                MYRESULT    ='1987196'
1696 *                MYRESULTL   =7
1697 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1699 * -----
1700 * | Function | VAR1 | VAR2 | Description |
1701 * +-----+-----+-----+-----+
1702 *48| CYMD-D8 | req  | opt  | Convert date in variable VAR1 in
1703 *   |         |      |      | format CCYYMMDD to monthname DD, CCYY
1704 *   |         |      |      | where:
1705 *   |         |      |      | monthname - Full month name (see below)
1706 *   |         |      |      | DD        - 2 digit date (01-31)
1707 *   |         |      |      | CCYY      - 4 digit century and year
1708 *
1709 *   |         |      |      | Monthnames:
1710 *   |         |      |      | 01-January 02-February 03-March
1711 *   |         |      |      | 04-April   05-May      06-June
1712 *   |         |      |      | 07-July    08-August   09-September
1713 *   |         |      |      | 10-October 11-November 12-December
1714 *
1715 *   |         |      |      | If VAR2 is omitted, results will be
1716 *   |         |      |      | placed into VAR1.
1717 * -----
1718 * SET MYDATA=&STR(19951231)          /* INIT MYDATA */
1719 * SET MYRESULT=                     /* INIT MYRESULT */
1720 * SET MYDATAL =                     /* INIT MYDATAL */
1721 * SET MYRESULTL=                     /* INIT MYRESULTL */
1722 * SET ERRMSG =                       /* INIT ERRMSG */
1723 * CUTIL00 CYMD-D8 MYDATA MYRESULT    /* CYMD-D8 */
1724 * SET RC=&LASTCC                     /* RETURN CODE */
1725 *
1726 * Results: RC =0
1727 *          ERRMSG ='SUCCESSFUL REQUEST'
1728 *          MYDATA ='19951231'
1729 *          MYDATAL =8
1730 *          MYRESULT ='DECEMBER 31, 1995'
1731 *          MYRESULTL=17
1732 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1734 * -----
1735 * | Function | VAR1 | VAR2 | Description |
1736 * +-----+ +-----+ +-----+ +-----+
1737 *49| CYMD-DAY | req  | opt  | Convert date in variable VAR1 in
1738 *      |      |      |      | format CCYYMMDD to weekday name.
1739 *
1740 *      |      |      |      | Weekday Name:
1741 *      |      |      |      | Sunday, Monday, Tuesday, Wednesday
1742 *      |      |      |      | Thursday, Friday, Saturday
1743 *
1744 *      |      |      |      | If VAR2 is omitted, results will be
1745 *      |      |      |      | placed into VAR1.
1746 * -----
1747 * SET MYDATA=&STR(19951231)                /* INIT MYDATA */
1748 * SET MYRESULT=                          /* INIT MYRESULT */
1749 * SET MYDATA1 =                          /* INIT MYDATA1 */
1750 * SET MYRESULT1=                        /* INIT MYRESULT1 */
1751 * SET ERRMSG =                          /* INIT ERRMSG */
1752 * CUTIL00 CYMD-DAY MYDATA MYRESULT      /* CYMD-DAY */
1753 * SET RC=&LASTCC                        /* RETURN CODE */
1754 *
1755 *      Results:  RC          =0
1756 *                ERRMSG    ='SUCCESSFUL REQUEST'
1757 *                MYDATA    ='19951231'
1758 *                MYDATA1   =8
1759 *                MYRESULT  ='SUNDAY'
1760 *                MYRESULT1=6
1761 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1763 * -----
1764 * | Function | VAR1 | VAR2 | Description |
1765 * +-----+ +-----+ +-----+ +-----+
1766 *50| CYMD-DOW | req  | opt  | Convert date in variable VAR1 in
1767 *      |      |      |      | format CCYYMMDD to weekday number.
1768 *      |      |      |      |
1769 *      |      |      |      | Weekday Number:
1770 *      |      |      |      | 0-6
1771 *      |      |      |      |
1772 *      |      |      |      | Weekday Number to Weekday Name:
1773 *      |      |      |      | 0-Sunday    1-Monday    2-Tuesday
1774 *      |      |      |      | 3-Wednesday 4-Thursday   5-Friday
1775 *      |      |      |      | 6-Saturday
1776 *      |      |      |      |
1777 *      |      |      |      | If VAR2 is omitted, results will be
1778 *      |      |      |      | placed into VAR1.
1779 * -----
1780 * SET MYDATA=&STR(19951231)                /* INIT MYDATA */
1781 * SET MYRESULT=                          /* INIT MYRESULT */
1782 * SET MYDATAL =                          /* INIT MYDATAL */
1783 * SET MYRESULTL=                        /* INIT MYRESULTL */
1784 * SET ERRMSG =                          /* INIT ERRMSG */
1785 * CUTIL00 CYMD-DOW MYDATA MYRESULT        /* CYMD-DOW */
1786 * SET RC=&LASTCC                          /* RETURN CODE */
1787 *
1788 *      Results:  RC          =0
1789 *                ERRMSG    ='SUCCESSFUL REQUEST'
1790 *                MYDATA    ='19951231'
1791 *                MYDATAL   =8
1792 *                MYRESULT  ='0'
1793 *                MYRESULTL=1
1794 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1796 * -----
1797 * | Function | VAR1 | VAR2 | Description |
1798 * +-----+ +-----+ +-----+ +-----+
1799 *51| CYMD-CYJ*| req  | opt  | Convert date in variable VAR1 in
1800 *      |      |      |      | format CCYYMMDD to CCYYJJJ.
1801 *
1802 *      | *CYJ can |      |      | For example-
1803 *      | be in any |      |      | CYMD-YJ  convert CCYYMMDD to YYJJJ
1804 *      | order   |      |      | CYMD-J   convert CCYYMMDD to JJJ
1805 *
1806 *      |      |      |      | JJJ- 3 digit julian day (1-366)
1807 *      |      |      |      | CC - 2 digit century
1808 *      |      |      |      | YY - 2 digit year
1809 *
1810 *      |      |      |      | If VAR2 is omitted, results will be
1811 *      |      |      |      | placed into VAR1.
1812 * -----
1813 * SET MYDATA=&STR(19951231)          /* INIT MYDATA */
1814 * SET MYRESULT=                    /* INIT MYRESULT */
1815 * SET MYDATAL =                    /* INIT MYDATAL */
1816 * SET MYRESULTL=                  /* INIT MYRESULTL */
1817 * SET ERRMSG =                    /* INIT ERRMSG */
1818 * CUTIL00 CYMD-CYJ MYDATA MYRESULT /* CYMD-CYJ */
1819 * SET RC=&LASTCC                   /* RETURN CODE */
1820 *
1821 *      Results:  RC          =0
1822 *                ERRMSG     ='SUCCESSFUL REQUEST'
1823 *                MYDATA     ='19951231'
1824 *                MYDATAL    =8
1825 *                MYRESULT   ='1995365'
1826 *                MYRESULTL  =7
1827 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1829 * -----
1830 * | Function | VAR1 | VAR2 | Description |
1831 * +-----+ +-----+ +-----+ +-----+
1832 *52| CYDM-D8 | req  | opt  | Convert date in variable VAR1 in
1833 *      |      |      |      | format CCYYDDMM to monthname DD, CCYY
1834 *      |      |      |      | where:
1835 *      |      |      |      | monthname - Full month name (see below)
1836 *      |      |      |      | DD        - 2 digit date (01-31)
1837 *      |      |      |      | CCYY      - 4 digit century and year
1838 *
1839 *      |      |      |      | Monthnames:
1840 *      |      |      |      | 01-January 02-February 03-March
1841 *      |      |      |      | 04-April   05-May      06-June
1842 *      |      |      |      | 07-July    08-August   09-September
1843 *      |      |      |      | 10-October 11-November 12-December
1844 *
1845 *      |      |      |      | If VAR2 is omitted, results will be
1846 *      |      |      |      | placed into VAR1.
1847 * -----
1848 * SET MYDATA=&STR(19951605)                /* INIT MYDATA */
1849 * SET MYRESULT=                          /* INIT MYRESULT */
1850 * SET MYDATAL =                          /* INIT MYDATAL */
1851 * SET MYRESULTL=                        /* INIT MYRESULTL */
1852 * SET ERRMSG =                          /* INIT ERRMSG */
1853 * CUTIL00 CYDM-D8 MYDATA MYRESULT        /* CYDM-D8 */
1854 * SET RC=&LASTCC                          /* RETURN CODE */
1855 *
1856 *      Results: RC          =0
1857 *                ERRMSG    ='SUCCESSFUL REQUEST'
1858 *                MYDATA    ='19951605'
1859 *                MYDATAL   =8
1860 *                MYRESULT  ='MAY 16, 1995'
1861 *                MYRESULTL=12
1862 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1864 * -----
1865 * | Function | VAR1 | VAR2 | Description |
1866 * +-----+ +-----+ +-----+ +-----+
1867 *53| CYDM-DAY | req  | opt  | Convert date in variable VAR1 in
1868 *   |         |      |      | format CCYYDDMM to weekday name.
1869 *   |         |      |      |
1870 *   |         |      |      | Weekday Name:
1871 *   |         |      |      | Sunday, Monday, Tuesday, Wednesday
1872 *   |         |      |      | Thursday, Friday, Saturday
1873 *   |         |      |      |
1874 *   |         |      |      | If VAR2 is omitted, results will be
1875 *   |         |      |      | placed into VAR1.
1876 * -----
1877 * SET MYDATA=&STR(19951605)                /* INIT MYDATA */
1878 * SET MYRESULT=                          /* INIT MYRESULT */
1879 * SET MYDATAL =                          /* INIT MYDATAL */
1880 * SET MYRESULTL=                        /* INIT MYRESULTL */
1881 * SET ERRMSG =                          /* INIT ERRMSG */
1882 * CUTIL00 CYDM-DAY MYDATA MYRESULT        /* CYDM-DAY */
1883 * SET RC=&LASTCC                          /* RETURN CODE */
1884 *
1885 *      Results:  RC          =0
1886 *                ERRMSG    ='SUCCESSFUL REQUEST'
1887 *                MYDATA     ='19951605'
1888 *                MYDATAL    =8
1889 *                MYRESULT   ='TUESDAY'
1890 *                MYRESULTL  =7
1891 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1893 * -----
1894 * | Function | VAR1 | VAR2 | Description |
1895 * +-----+-----+-----+
1896 *54| CYDM-DOW | req  | opt  | Convert date in variable VAR1 in
1897 *      |      |      |      | format CCYYDDMM to weekday number.
1898 *
1899 *      |      |      |      | Weekday Number:
1900 *      |      |      |      | 0-6
1901 *
1902 *      |      |      |      | Weekday Number to Weekday Name:
1903 *      |      |      |      | 0-Sunday    1-Monday    2-Tuesday
1904 *      |      |      |      | 3-Wednesday 4-Thursday   5-Friday
1905 *      |      |      |      | 6-Saturday
1906 *
1907 *      |      |      |      | If VAR2 is omitted, results will be
1908 *      |      |      |      | placed into VAR1.
1909 * -----
1910 * SET MYDATA=&STR(19951605)                /* INIT MYDATA */
1911 * SET MYRESULT=                            /* INIT MYRESULT */
1912 * SET MYDATAL =                           /* INIT MYDATAL */
1913 * SET MYRESULTL=                          /* INIT MYRESULTL */
1914 * SET ERRMSG =                            /* INIT ERRMSG */
1915 * CUTIL00 CYDM-DOW MYDATA MYRESULT        /* CYDM-DOW */
1916 * SET RC=&LASTCC                          /* RETURN CODE */
1917 *
1918 *      Results: RC          =0
1919 *                ERRMSG    ='SUCCESSFUL REQUEST'
1920 *                MYDATA    ='19951605'
1921 *                MYDATAL   =8
1922 *                MYRESULT  ='2'
1923 *                MYRESULTL=1
1924 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1926 * -----
1927 * | Function | VAR1 | VAR2 | Description |
1928 * +-----+ +-----+ +-----+ +-----+
1929 *55| CYMD-CYJ*| req  | opt  | Convert date in variable VAR1 in
1930 *   |         |      |      | format CCYYDDMM to CCYYJJJ.
1931 *
1932 *   *CYJ can
1933 *   be in any
1934 *   order
1935 *
1936 *   JJJ- 3 digit julian day (1-366)
1937 *   CC - 2 digit century
1938 *   YY - 2 digit year
1939 *
1940 *   If VAR2 is omitted, results will be
1941 *   placed into VAR1.
1942 * -----
1943 * SET MYDATA=&STR(19951605)          /* INIT MYDATA */
1944 * SET MYRESULT=                    /* INIT MYRESULT */
1945 * SET MYDATAL =                    /* INIT MYDATAL */
1946 * SET MYRESULTL=                  /* INIT MYRESULTL */
1947 * SET ERRMSG =                    /* INIT ERRMSG */
1948 * CUTIL00 CYMD-CYJ MYDATA MYRESULT /* CYMD-CYJ */
1949 * SET RC=&LASTCC                   /* RETURN CODE */
1950 *
1951 * Results: RC =0
1952 *          ERRMSG ='SUCCESSFUL REQUEST'
1953 *          MYDATA ='19951605'
1954 *          MYDATAL =8
1955 *          MYRESULT ='1995136'
1956 *          MYRESULTL=7
1957 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1959 * -----
1960 * | Function | VAR1 | VAR2 | Description |
1961 * +-----+ +-----+ +-----+ +-----+
1962 *56| FILL      | req   | req   | Fill variable VAR1 with a delimiter
1963 *          |          |          |          | value n times.
1964 *          |          |          |          |
1965 *          |          |          |          | VAR1 - variable for result
1966 *          |          |          |          |
1967 *          |          |          |          | VAR2 contains Fill Parms -----
1968 *          |          |          |          | Format: c n n n
1969 *          |          |          |          |
1970 *          |          |          |          | c - single byte character
1971 *          |          |          |          | n - length of fill content, 1-3 digits
1972 *          |          |          |          | valid length values, 1-256
1973 *          |          |          |          | default=1
1974 * -----
1975 * SET MYDATA=&STR(02281980)                /* INIT MYDATA */
1976 * SET MYRESULT=                          /* INIT MYRESULT */
1977 * SET MYDATAL =                          /* INIT MYDATAL */
1978 * SET MYRESULTL=                        /* INIT MYRESULTL */
1979 * SET ERRMSG =                          /* INIT ERRMSG */
1980 * CUTIL00 FILL MYDATA MYPARMS            /* FILL */
1981 * SET RC=&LASTCC                          /* RETURN CODE */
1982 *
1983 *      Results:  RC          =0
1984 *                ERRMSG     ='SUCCESSFUL REQUEST'
1985 *                MYDATA      ='*****'
1986 *                MYDATAL     =20
1987 *                MYPARMS     ='*20'
1988 *                MYPARMSL    =3
1989 *
1990 *
1991 *
1992 *
1993 *
1994 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
1996 * -----
1997 * | Function | VAR1 | VAR2 | Description |
1998 * +-----+-----+-----+-----+
1999 *57| LSTRIP   | req  | opt  | Remove leading delimiter from variable
2000 *          |      |      |      | VAR1.
2001 *          |      |      |      |
2002 *          |      |      |      | VAR1 will contain result
2003 *          |      |      |      |
2004 *          |      |      |      | VAR2 contains delimiter -----
2005 *          |      |      |      | Format: c
2006 *          |      |      |      |
2007 *          |      |      |      | c - single byte character delimiter
2008 *          |      |      |      | default=BLANK, if not specified
2009 * -----
2010 * SET MYDATA=&STR(, , 28, BC, )          /* INIT MYDATA */
2011 * SET MYRESULT=                          /* INIT MYRESULT */
2012 * SET MYDATAL =                          /* INIT MYDATAL */
2013 * SET MYRESULTL=                        /* INIT MYRESULTL */
2014 * SET ERRMSG =                          /* INIT ERRMSG */
2015 * CUTIL00 LSTRIP MYDATA MYDELIM          /* LSTRIP */
2016 * SET RC=&LASTCC                        /* RETURN CODE */
2017 *
2018 * Results: RC =0
2019 *           ERRMSG ='SUCCESSFUL REQUEST'
2020 *           MYDATA ='28,BC,'
2021 *           MYDATAL =6
2022 *           MYDELIM =' ,'
2023 *           MYDELIML =1
2024 *
2025 *
2026 *
2027 *
2028 *
2029 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2031 * -----
2032 * | Function | VAR1 | VAR2 | Description |
2033 * +-----+ +-----+ +-----+ +-----+
2034 *58| RSTRIP   | req   | opt   | Remove trailing delimiter from variable |
2035 *          |          |          |          | VAR1. |
2036 *          |          |          |          |
2037 *          |          |          |          | VAR1 will contain result |
2038 *          |          |          |          |
2039 *          |          |          |          | VAR2 contains delimiter ----- |
2040 *          |          |          |          | Format: c |
2041 *          |          |          |          |
2042 *          |          |          |          | c - single byte character delimiter |
2043 *          |          |          |          | default=BLANK, if not specified |
2044 * -----
2045 * SET MYDATA=&STR(,,28,BC,) /* INIT MYDATA */
2046 * SET MYRESULT= /* INIT MYRESULT */
2047 * SET MYDATAL = /* INIT MYDATAL */
2048 * SET MYRESULTL= /* INIT MYRESULTL */
2049 * SET ERRMSG = /* INIT ERRMSG */
2050 * CUTIL00 RSTRIP MYDATA MYDELIM /* RSTRIP */
2051 * SET RC=&LASTCC /* RETURN CODE */
2052 *
2053 * Results: RC =0
2054 *          ERRMSG ='SUCCESSFUL REQUEST'
2055 *          MYDATA =' , , 28, BC'
2056 *          MYDATAL =7
2057 *          MYDELIM =' , '
2058 *          MYDELIML =1
2059 *
2060 *
2061 *
2062 *
2063 *
2064 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2066 * -----
2067 * | Function | VAR1 | VAR2 | Description |
2068 * +-----+ +-----+ +-----+ +-----+
2069 *59| STRIP    | req  | opt  | Remove leading and trailing delimiter
2070 *          |      |      |      | from variable VAR1.
2071 *          |      |      |      |
2072 *          |      |      |      | VAR1 will contain result
2073 *          |      |      |      |
2074 *          |      |      |      | VAR2 contains delimiter -----
2075 *          |      |      |      | Format: c
2076 *          |      |      |      |
2077 *          |      |      |      | c - single byte character delimiter
2078 *          |      |      |      | default=BLANK, if not specified
2079 * -----
2080 * SET MYDATA=&STR(, , 28, BC, ) /* INIT MYDATA */
2081 * SET MYRESULT= /* INIT MYRESULT */
2082 * SET MYDATAL = /* INIT MYDATAL */
2083 * SET MYRESULTL= /* INIT MYRESULTL */
2084 * SET ERRMSG = /* INIT ERRMSG */
2085 * CUTIL00 STRIP MYDATA MYDELIM /* STRIP */
2086 * SET RC=&LASTCC /* RETURN CODE */
2087 *
2088 * Results: RC =0
2089 *          ERRMSG ='SUCCESSFUL REQUEST'
2090 *          MYDATA ='28,BC'
2091 *          MYDATAL =5
2092 *          MYDELIM =' '
2093 *          MYDELIML =1
2094 *
2095 *
2096 *
2097 *
2098 *
2099 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2101 * -----
2102 * | Function | VAR1 | VAR2 | Description |
2103 * +-----+ +-----+ +-----+ +-----+
2104 *60| STRIP$   | req  | opt  | Remove all delimiter values from
2105 *          |      |      |      | variable VAR1.
2106 *          |      |      |      |
2107 *          |      |      |      | VAR1 will contain result
2108 *          |      |      |      |
2109 *          |      |      |      | VAR2 contains delimiter -----
2110 *          |      |      |      | Format: c
2111 *          |      |      |      |
2112 *          |      |      |      | c - single byte character delimiter
2113 *          |      |      |      | default=BLANK, if not specified
2114 * -----
2115 * SET MYDATA=&STR(, , 28, BC, )          /* INIT MYDATA */
2116 * SET MYRESULT=                          /* INIT MYRESULT */
2117 * SET MYDATAL =                          /* INIT MYDATAL */
2118 * SET MYRESULTL=                        /* INIT MYRESULTL */
2119 * SET ERRMSG =                          /* INIT ERRMSG */
2120 * CUTIL00 STRIP$ MYDATA MYDELIM          /* STRIP$ */
2121 * SET RC=&LASTCC                         /* RETURN CODE */
2122 *
2123 * Results: RC =0
2124 *          ERRMSG ='SUCCESSFUL REQUEST'
2125 *          MYDATA ='28BC'
2126 *          MYDATAL =4
2127 *          MYDELIM =' , '
2128 *          MYDELIML =1
2129 *
2130 *
2131 *
2132 *
2133 *
2134 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2136 * -----
2137 * | Function | VAR1 | VAR2 | Description |
2138 * +-----+-----+-----+
2139 *61| CONCAT  | req  | req  | Concatenate VAR1 and VAR2 into variable |
2140 * |          |      |      | VAR1. |
2141 * |          |      |      |      |
2142 * |          |      |      |      |
2143 * -----
2144 * SET MYDATA=&STR(String data 1.)          /* INIT MYDATA */
2145 * SET MYRESULT=&STR(A result string 2.)     /* INIT MYRESULT */
2146 * SET MYDATAL =                            /* INIT MYDATAL */
2147 * SET MYRESULTL=                          /* INIT MYRESULTL */
2148 * SET ERRMSG =                            /* INIT ERRMSG */
2149 * CUTIL00 CONCAT MYDATA MYRESULT          /* CONCAT */
2150 * SET RC=&LASTCC                          /* RETURN CODE */
2151 *
2152 * Results: RC =0
2153 *           ERRMSG ='SUCCESSFUL REQUEST '
2154 *           MYDATA ='String data 1.A result string 2.'
2155 *           MYDATAL =32
2156 *           MYRESULT ='A result string 2.'
2157 *           MYRESULTL=18
2158 *
2159 *
2160 *
2161 *
2162 *
2163 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2165 * -----
2166 * | Function | VAR1 | VAR2 | Description |
2167 * +-----+ +-----+ +-----+ +-----+
2168 *62 | UNSTR    | req  | opt  | Unstring content in variable VAR1 into
2169 *   |         |      |      | separate variables (VAR1n) based on a
2170 *   |         |      |      | 1-byte delimiter.
2171 *   |         |      |      |
2172 *   |         |      |      | VAR1 contains content for unstring.
2173 *   |         |      |      |
2174 *   |         |      |      | VAR2 contains delimiter -----
2175 *   |         |      |      | Format: c
2176 *   |         |      |      |
2177 *   |         |      |      | c - single byte character delimiter
2178 *   |         |      |      | default=BLANK, if not specified
2179 *   |         |      |      |
2180 *   |         |      |      | VAR10 contains number of variables
2181 *   |         |      |      |
2182 *   |         |      |      | VAR10,1,2,3,4,5,6,7,8,9,10,11 to 99
2183 *   |         |      |      | are possible variable names with
2184 *   |         |      |      | representing unstringed values.
2185 * -----
2186 * SET MYDATA=&STR(1234567 AND SO ON . I AM 78 LONG. ) /* MYDATA */
2187 * SET MYRESULT=&STR(7) /* INIT MYRESULT */
2188 * SET MYDATA1 = /* INIT MYDATA1 */
2189 * SET MYRESULT1= /* INIT MYRESULT1 */
2190 * SET ERRMSG = /* INIT ERRMSG */
2191 * CUTIL00 CONCAT MYDATA MYRESULT /* CONCAT */
2192 * SET RC=&LASTCC /* RETURN CODE */
2193 *
2194 * Results: RC =0
2195 *          ERRMSG ='SUCCESSFUL REQUEST'
2196 *          MYDATA ='1234567 AND SO ON . I AM 78 LONG. '
2197 *          MYDATA1 =34
2198 *          MYRESULT ='7'
2199 *          MYRESULT1=1
2200 *          MYDATA0 = '003'
2201 *          MYDATA1 = '123456'
2202 *          MYDATA2 = ' AND SO ON . I AM '
2203 *          MYDATA3 = '8 LONG. '
2204 *
2205 *
2206 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2208 * -----
2209 * | Function | VAR1 | VAR2 | Description |
2210 * +-----+ +-----+ +-----+ +-----+
2211 *63| REPLACE | req  | req  | Replace str with st2
2212 *      |      |      |      | get some desp from cound and find
2213 *      |      |      |      | 1-byte delimiter.
2214 *
2215 *      |      |      |      | RC=occurance count
2216 *      |      |      |      | occurance count of 0 = NOT FOUND
2217 *
2218 *      |      |      |      | VAR1 contains content for replace.
2219 *
2220 *      |      |      |      | VAR2 contains Search Params -----
2221 *      |      |      |      | Format: "ssss","rrrrr",bbb,eee
2222 *
2223 *      |      |      |      | s - search string in double quotes
2224 *      |      |      |      | r - replace string in double quotes
2225 *      |      |      |      | b - optional, begining search position
2226 *      |      |      |      | 1-3 numeric digits, default=1
2227 *      |      |      |      | default=1
2228 *      |      |      |      | e - optional, ending search position
2229 *      |      |      |      | 1-3 numeric digits, default=end
2230 *      |      |      |      | Params separated by commas (,):
2231 *
2232 * -----
2233 * SET MYDATA=&STR(1234567 AND SO ON . I AM 7879.) /* INIT MYDATA */
2234 * SET MYRESULT=&STR("SO ON ","890123456--") /* INIT MYRESULT */
2235 * SET MYDATAL = /* INIT MYDATAL */
2236 * SET MYRESULTL= /* INIT MYRESULTL */
2237 * SET ERRMSG = /* INIT ERRMSG */
2238 * CUTIL00 REPLACE MYDATA MYRESULT /* REPLACE */
2239 * SET RC=&LASTCC /* RETURN CODE */
2240 *
2241 * Results: RC =0
2242 * ERRMSG ='SUCCESSFUL REQUEST'
2243 * MYDATA ='1234567 AND 890123456--. I AM 7879.'
2244 * MYDATAL =35
2245 * MYRESULT ='SO ON ","890123456--'
2246 * MYRESULTL=22
2247 *
2248 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2250 * -----
2251 * | Function | VAR1 | VAR2 | Description |
2252 * +-----+ +-----+ +-----+ +-----+
2253 *64| VEXIST  | req   | no    | Test for existance of VAR1 variable.
2254 *          |          |          |          | Results in return code (RC).
2255 *          |          |          |          |
2256 *          |          |          |          | RC=nnn Length of variable
2257 *          |          |          |          | RC=4000 Variable NOT found
2258 *          |          |          |          |
2259 *          |          |          |          | No content of variable is returned.
2260 * -----
2261 *
2262 * SET MYDATA=&STR(TEST)                /* INIT MYDATA */
2263 * SET MYRESULT=                        /* INIT MYRESULT */
2264 * SET MYDATAAL =                      /* INIT MYDATAAL */
2265 * SET MYRESULTL=                      /* INIT MYRESULTL */
2266 * SET ERRMSG =                        /* INIT ERRMSG */
2267 * CUTIL00 VEXIST MYDATA                /* VEXIST */
2268 * SET RC=&LASTCC                       /* RETURN CODE */
2269 *
2270 *      Results:  RC          =4
2271 *                ERRMSG     ='SUCCESSFUL REQUEST'
2272 *                MYDATA     ='TEST'
2273 *                MYDATAAL   =4
2274 *                MYRESULT   =' '
2275 *                MYRESULTL=
2276 *
2277 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2279 * -----
2280 * | Function | VAR1 | VAR2 | Description |
2281 * +-----+-----+-----+
2282 *65| TDSN      | req   | no    | Generate temporary DSN nodes formatted
2283 * |           |       |       | as 'Dyyddd.Thhmmss' concatenated to
2284 * |           |       |       | end of VAR1 variable.
2285 * |           |       |       |
2286 * |           |       |       | Note: A period '.' may be placed before
2287 * |           |       |       | the date and time nodes should one not
2288 * |           |       |       | exist in the provided variable VAR1.
2289 * |           |       |       |
2290 * -----
2291 * SET MYDATA='HERC01.CARDS'                /* INIT MYDATA */
2292 * SET MYRESULT=                            /* INIT MYRESULT */
2293 * SET MYDATA1 =                            /* INIT MYDATA1 */
2294 * SET MYRESULT1=                           /* INIT MYRESULT1 */
2295 * SET ERRMSG =                             /* INIT ERRMSG */
2296 * CUTIL00 TDSN MYDATA                      /* TDSN */
2297 * SET RC=&LASTCC                            /* RETURN CODE */
2298 *
2299 * Results: RC =0
2300 * ERRMSG = 'SUCCESSFUL REQUEST '
2301 * MYDATA = 'HERC01.CARDS.D19231.T183011'
2302 * MYDATA1 =27
2303 * MYRESULT = ' '
2304 * MYRESULT1=
2305 *
2306 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2308 * -----
2309 * | Function | VAR1 | VAR2 | Description |
2310 * +-----+ +-----+ +-----+ +-----+
2311 *66| NOW      | req   | no    | Current system date and time returned |
2312 * |          |      |      | in variable VAR01. |
2313 * |          |      |      | |
2314 * |          |      |      | Format: |
2315 * |          |      |      | 'MM/DD/CCYY.JJJ HH:MM:SS.TT N nnnnnnnnn' |
2316 * -----
2317 * SET MYDATA=                                /* INIT MYDATA */
2318 * SET MYRESULT=                              /* INIT MYRESULT */
2319 * SET MYDATA1 =                             /* INIT MYDATA1 */
2320 * SET MYRESULT1=                            /* INIT MYRESULT1 */
2321 * SET ERRMSG =                               /* INIT ERRMSG */
2322 * CUTIL00 NOW MYDATA                        /* NOW */
2323 * SET RC=&LASTCC                             /* RETURN CODE */
2324 *
2325 * Results: RC =0
2326 * ERRMSG = 'SUCCESSFUL REQUEST'
2327 * MYDATA = '03/29/2020.089 21:07:26.42 0 SUNDAY'
2328 * MYDATA1 =38
2329 * MYRESULT =''
2330 * MYRESULT1=
2331 *
2332 *
```

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
2334 * -----
2335 * | Function | VAR1 | VAR2 | Description |
2336 * +-----+ +-----+ +-----+ +-----+
2337 *67| PAD      | req  | req  | Pad current string with delimiter
2338 *      |          |      |      | for a given length.
2339 *      |          |      |      |
2340 *      |          |      |      | VAR1 - variable for result
2341 *      |          |      |      |
2342 *      |          |      |      | VAR2 contains Fill Parms -----
2343 *      |          |      |      | Format: cnnn
2344 *      |          |      |      |
2345 *      |          |      |      | c - single byte character
2346 *      |          |      |      | n - resulting length, 1-3 digits
2347 *      |          |      |      | valid length values, 1-256
2348 *      |          |      |      | default=1
2349 *      |          |      |      | Note: No padding will occur when
2350 *      |          |      |      | resulting length less than or
2351 *      |          |      |      | equal to VAR1 length
2352 *      |          |      |      |
2353 *      |          |      |      |
2354 *      |          |      |      | RC=4040 Invalid Length
2355 *      |          |      |      | one of the following errors:
2356 *      |          |      |      | - VAR2 contains more than 4
2357 *      |          |      |      | characters per above format
2358 *      |          |      |      | - nnn is zero
2359 *      |          |      |      | - VAR2 is null
2360 *      |          |      |      | - resulting string > 256 bytes
2361 * -----
2362 * SET MYDATA='PAD ME TO 30 LONG' /* INIT MYDATA */
2363 * SET MYRESULT=' 30' /* INIT MYRESULT */
2364 * SET MYDATAL = /* INIT MYDATAL */
2365 * SET MYRESULTL= /* INIT MYRESULTL */
2366 * SET ERRMSG = /* INIT ERRMSG */
2367 * CUTIL00 PAD MYDATA /* PAD */
2368 * SET RC=&LASTCC /* RETURN CODE */
2369 *
2370 * Results: RC =0
2371 * ERRMSG ='SUCCESSFUL REQUEST'
2372 * MYDATA ='PAD ME TO 30 LONG'
2373 * MYDATAL =30
2374 * MYRESULT =' 30'
2375 * MYRESULTL=3
2376 *
2377 *
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

				2379 *	CUTIL00 Command Examples:	
--	--	--	--	--------	---------------------------	--

				2380 *	=====	
--	--	--	--	--------	-------	--

				2381 *		
--	--	--	--	--------	--	--

				2382 *	1) CUTIL00 TRIM VAR01 VAR02	
--	--	--	--	--------	-----------------------------	--

				2383 *	Trim contents of &VAR01 variable and place results in &VAR02	
				2384 *	variable.	

				2385 *		
--	--	--	--	--------	--	--

				2386 *	2) CUTIL00 TRIM VAR01	
--	--	--	--	--------	-----------------------	--

				2387 *	Trim contents of &VAR01 variable and place results in &VAR01	
				2388 *	variable. Note: Original content is replaced.	

				2389 *		
--	--	--	--	--------	--	--

				2390 *	3) CUTIL00 TRIM VAR01 VAR02 QUOTE	
--	--	--	--	--------	-----------------------------------	--

				2391 *	Trim contents of &VAR01 variable and place results in &VAR02	
				2392 *	variable. Note: VAR01 content will be delimited by single	

				2393 *	quotes (').	
--	--	--	--	--------	-------------	--

				2394 *		
--	--	--	--	--------	--	--

				2395 *		
--	--	--	--	--------	--	--

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

2397 *

2398 * Module: 3 base registers, about 12kb

2399 * Storage: GETMAIN for about 3.3kb

2400 *

2401 *

2402 *

2403 * Table 1 - Program Register Usage

2404 *

2405 *

2406 *

2407 *

2408 *

2409 *

2410 *

2411 *

2412 *

2413 *

2414 *

2415 *

2416 *

2417 *

2418 *

2419 *

2420 *

2421 *

2422 *

2423 *

2424 *

2425 *

2426 *

2427 *

2428 *

2429 *

2430 *

2431 *

REG	DESCRIPTION
USAGE	DESCRIPTION
R13	Calling program registers (R14-R12) upon entry
R14	Calling program return address upon entry
R15	Called program entry address upon entry
R1	Called program parms starting address
R0	Working Register
R1	Working Register
R2	Working Register
R3	Working Register
R4	Working Register
R5	Working Register
R6	Working Register
R7	Working Register
R8	Working Register
R9	Working Register
R10	Base Register 1
R11	Base Register 2
R12	Base Register 3
R13	Address of WORKAREA including SAVEAREA
R15	Return Code upon exit

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

2433 * Table 2 - Return Codes

2434 *

2435 *

2436 * |RETURN|

2437 * | CODE | DESCRIPTION

2438 *

2439 * | 4000 | FALSE

2440 *

2441 * | 4001 | TRUE

2442 *

2443 * | 4002 | VAR1 not found

2444 *

2445 * | 4004 | No PARM

2446 *

2447 * | 4005 | WORD GT 8 bytes

2448 *

2449 * | 4008 | CPPL no Parm

2450 *

2451 * | 4009 | Too many WORDS

2452 *

2453 * | 4010 | SETVAR no content

2454 *

2455 * | 4011 | Invalid function

2456 *

2457 * | 4012 | VAR2 not updated

2458 *

2459 * | 4014 | VAR1 GT 256

2460 *

2461 * | 4015 | VAR2 required

2462 *

2463 * | 4016 | VAR2 not found

2464 *

2465 * | 4017 | VAR2 GT 256

2466 *

2467 * | 4018 | VAR2 null

2468 *

2469 * | 4019 | VAR1 null

2470 *

2471 * | 4020 | ERRMSG cannot create updt

2472 *

2473 * | 4021 | VAR1 required

2474 *

2475 * | 4022 | VAR2 not required

2476 *

2477 * | 4023 | Result GT 256

2478 *

2479 * | 4030 | No begin "

2480 *

2481 * | 4031 | No end "

2482 *

2483 * | 4032 | Start GT End

2484 *

2485 * | 4033 | Invalid Char

2486 *

	2488	*	Table 2 - Return Codes (continued)					
	2489	*						
	2490	*	+-----+	+-----+				
	2491	*	4034	Not numeric				
	2492	*	+-----+	+-----+				
	2493	*	4035	Too many digits				
	2494	*	+-----+	+-----+				
	2495	*	4036	Null value not allowed				
	2496	*	+-----+	+-----+				
	2497	*	4037	No begin QUOTE				
	2498	*	+-----+	+-----+				
	2499	*	4038	No end QUOTE				
	2500	*	+-----+	+-----+				
	2501	*	4039	No end QUOTE found				
	2502	*	+-----+	+-----+				
	2503	*	4040	Invalid length				
	2504	*	+-----+	+-----+				
	2505	*	4047	Invalid DD				
	2506	*	+-----+	+-----+				
	2507	*	4049	Invalid MM				
	2508	*	+-----+	+-----+				
	2509	*	4050	Invalid DD for MM				
	2510	*	+-----+	+-----+				
	2511	*	4051	Invalid JJJ				
	2512	*	+-----+	+-----+				
	2513	*	4098	Cannot link to IKJCT441				
	2514	*	+-----+	+-----+				
	2515	*	4099	Must run under TSO				
	2516	*	+-----+	+-----+				
	2517	*	Note: -Symbolic Variable &LASTCC contains CUTIL00 return code					

2519	*	Macros and SYSLIB Location:		
2520	*	=====		
2521	*			
2522	*	Macro	Description	Library
2523	*	-----	-----	-----
2524	*	YREGS	Register Equates	SYS2.MACLIB
2525	*	IHAPSA	Prefixed Save Area	SYS1.AMODGEN
2526	*	CVT	Communication Vector Table	SYS1.AMODGEN
2527	*	IKJCPPL	Command Program Parameter Table	SYS1.AMODGEN
2528	*	IKJTCB	Task Control Block	SYS1.AMODGEN
2529	*	IEFTIOT1	Task Input/Output Table	SYS1.AMODGEN
2530	*	IEFJFCBN	Job File Control Block	SYS1.AMODGEN
2531	*	IHAASCB	Address Space Control Block	SYS1.AMODGEN
2532	*			
2533	*			
2534	*	References:		
2535	*	=====		
2536	*			
2537	*	- SC34-2088-2 ISPF Dialog Management MVS (March 1985)		
2538	*	- GC28-0646-4 OS/VS2 TSO Command Language Reference (VS2 Rel 3.8)		
2539	*			
2540	*			

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	-------------	-------	-------	------	------------------	-------------------------

				2542 *	Change History: <CHGHIST>	
--	--	--	--	--------	---------------------------	--

				2543 *	=====	
--	--	--	--	--------	-------	--

				2544 *		
--	--	--	--	--------	--	--

				2545 *	MM/DD/CCYY Version Name / Description	
--	--	--	--	--------	--	--

				2546 *	-----	
--	--	--	--	--------	-------	--

				2547 *	03/20/2020 1.0.00 Larry Belmontes Jr.	
--	--	--	--	--------	--	--

				2548 *	- Initial version released to MVS 3.8J	
--	--	--	--	--------	--	--

				2549 *	hobbyist public domain	
--	--	--	--	--------	------------------------	--

				2550 *		
--	--	--	--	--------	--	--

				2551 *		
--	--	--	--	--------	--	--

				2552 *		
--	--	--	--	--------	--	--

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
-----	--------	------	-------	-------	------	------------------	-------------------------

					2554	* Potential Enhancements: <ENHMNTS>	
					2555	* =====	
					2556	*	
					2557	* 01) Hmmm... thinking...	
					2558	*	
					2559	* * * * *	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
000000				2561	CUTIL00 CSECT	
			00000	2562	USING CUTIL00,R10,R11,R12 my BASE REGISTER(S)	
				2563	PRINT NOGEN	
				2564	* /* *****/	
				2565	* /* Save regs and declare base registers R10,R11,R12 */	
				2566	* /* including saving R1 (PARMS) in R2 */	
				2567	* /* - R1 myPARMS address on entry */	
				2568	* /* - R15 myENTRY address on entry */	
				2569	* /* *****/	
000000	90EC D00C	0000C		2570	STM R14,R12,12(R13) Save registers in HSA	
000004	18AF			2571	LR R10,R15 Load base1 (R10) w R15	
000006	41BA 0FFF	00FFF		2572	LA R11,4095(R10) Load base2 (R11) w/ offset	
00000A	41BB 0001	00001		2573	LA R11,1(R11) Adjust base2	
00000E	41CB 0FFF	00FFF		2574	LA R12,4095(R11) Load base3 (R12) w/ offset	
000012	41CC 0001	00001		2575	LA R12,1(R12) Adjust base3	
000016	1821			2576	LR R2,R1 Save PARM addr in R2	
				2577	* /* *****/	
				2578	* /* Program Eyecatcher */	
				2579	* /* *****/	
000018	47F0 A0A6	000A6		2580	B OVRSTAMP Branch past eye-catcher	
			0001C	2581	EQU * EQU	
00001C	C3E4E3C9D3F0F040			2582	PGMID DC CL8'CUTIL00 ' My Program STAMP	
000024	D4E5E2F34BF8D140			2583	DC CL8'MVS3.8J ' OS	
00002C	E5F14BF04BF0F040			2584	DC CL8'V1.0.00 ' .Version	
000034	F0F3F2F0F2F0F2F0			2585	DC CL8'03202020' ..date MMDDCCYY	
00003C	40			2586	DC CL1' ' CL1'	
				2587	PGMDS DC CL8'&SYSDATE' Assemble Date	
000045	407C40			2589	DC CL3' @ ' .and	
				2590	PGMTS DC CL8'&SYSTIME' ..Time	
			00034	2592	MYSTAMPL EQU *-MYSTAMP	
000050	C39697A899898788			2593	DC C'Copyright (C) 2020 '	
000063	D3819999A840C285			2594	DC C'Larry Belmontes, Jr. '	
000078	88A3A397A27A6161			2595	DC C'https://ShareABitofIT.net/CUTIL00-for-MVS-3-8J'	
0000A6				2596	OVRSTAMP DS 0H	
				2597	*	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					2599 *	* /*****	
					2600 *	* /* GETMAIN, working storage, using R9	*/
					2601 *	* /* - R1 Working register (GETMAIN address)	*/
					2602 *	* /* - R13 Working Storage DSECT address	*/
					2603 *	* /*****	
					2604	PRINT GEN	
					2605	GETMAIN R, LV=WSAREAL	Get Storage, R1 = addr of storage
0000A6	0700				2606+	CNOP 0,4	
0000A8	4510	A0B0	000B0		2607+	BAL 1, *+8	BRANCH AROUND LENGTH
0000AC	00000CE8				2608+	DC A(WSAREAL)	LENGTH
0000B0	5800	1000	00000		2609+	L 0,0(0,1)	LOAD LENGTH
0000B4	0A0A				2610+	SVC 10	ISSUE GETMAIN SVC
0000B6	12FF				2611	LTR R15,R15	Did we get the storage?
0000B8	4770	BA30	01A30		2612	BNZ NOMAINS	NO, return with rc in R15
					2613	PRINT NOGEN	
					2614 *	* /*****	
					2615 *	* /* Update Save Areas for caller and calling programs	*/
					2616 *	* /* - R1 myPARMS address on entry (restored)	*/
					2617 *	* /*****	
0000BC	50D1	0004	00004		2618	ST R13,4(R1)	Store HSA addr, chain caller > mySA
0000C0	501D	0008	00008		2619	ST R1,8(R13)	Store LSA addr, chain mySA > caller
0000C4	18D1				2620	LR R13,R1	R13 = Workarea DSECT Register
			00000		2621	USING WSDSECT,R13	Tell Assembler
0000C6	1812				2622	LR R1,R2	Restore R1 as PARM addr
					2623 *	* /*****	
					2624 *	* /* Load IKJCT441 Entry Point	*/
					2625 *	* /* - R0, R1	Working Register
					2626 *	* /*****	
					2627	LOAD EP=IKJCT441,ERRET=ERR4098	
0000E8	5000	D25C	0025C		2637	ST R0,IKJCT441	Load and Save IKJCT441 Entry Point
0000EC	1812				2638	LR R1,R2	Restore R1 as PARM addr
					2639 *	* /*****	
					2640 *	* /* Check if we are running under TS0	*/
					2641 *	* /* - R0	Working Register
					2642 *	* /*****	
0000EE	5800	0224	00224		2643	L R0,PSAA0LD-PSA(0)	R0=ASCB Address
0000F2	5800	003C	0003C		2644	L R0,ASCBTSB-ASCB(R0)	R0=ASCBTSB Address
0000F6	1200				2645	LTR R0,R0	If 0, program not under TS0
0000F8	4780	B9F4	019F4		2646	BZ ERR4099	..error, not under TS0!
					2647 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					2649 *	*/*****	
					2650 *	/* Initialize Working Storage	*/
					2651 *	/* - R2, R3 Working Register	*/
					2652 *	*/*****	
0000FC	D203	D1EC	C7D8	001EC	027D8	2653 MVC TOKEN,=F'0' TOKEN	
					2654 *	*/*****	
					2655 *	/* Initialize VAR pointers	*/
					2656 *	*/*****	
000102	4120	0008		00008	2657	LA R2,L'NAM1	VAR1 name length
000106	5020	D278		00278	2658	ST R2,NAML1	
00010A	4120	D4D0		004D0	2659	LA R2,NAM1	VAR1 name pointer
00010E	5020	D27C		0027C	2660	ST R2,NAMP1	
000112	4120	0100		00100	2661	LA R2,L'VAL1	VAR1 value length
000116	5020	D280		00280	2662	ST R2,VALL1	
00011A	4120	D4E1		004E1	2663	LA R2,VAL1	VAR1 value pointer
00011E	5020	D284		00284	2664	ST R2,VALP1	
					2665 *		
000122	4120	0008		00008	2666	LA R2,L'NAM2	VAR2 name length
000126	5020	D288		00288	2667	ST R2,NAML2	
00012A	4120	D4D8		004D8	2668	LA R2,NAM2	VAR2 name pointer
00012E	5020	D28C		0028C	2669	ST R2,NAMP2	
000132	4120	0100		00100	2670	LA R2,L'VAL2	VAR2 value length
000136	5020	D290		00290	2671	ST R2,VALL2	
00013A	4120	D5E3		005E3	2672	LA R2,VAL2	VAR2 value pointer
00013E	5020	D294		00294	2673	ST R2,VALP2	
					2674 *		
000142	4120	0009		00009	2675	LA R2,L'NAM3	VAR3 name length
000146	5020	D2A4		002A4	2676	ST R2,NAML3	
00014A	4120	D298		00298	2677	LA R2,NAM3	VAR3 name pointer
00014E	5020	D2A8		002A8	2678	ST R2,NAMP3	
000152	4120	0100		00100	2679	LA R2,L'VAL3	VAR3 value length
000156	5020	D3AC		003AC	2680	ST R2,VALL3	
00015A	4120	D2AC		002AC	2681	LA R2,VAL3	VAR3 value pointer
00015E	5020	D3B0		003B0	2682	ST R2,VALP3	
					2683 *		
000162	4120	000B		0000B	2684	LA R2,L'NAM4	VAR4 name length
000166	5020	D3C0		003C0	2685	ST R2,NAML4	
00016A	4120	D3B4		003B4	2686	LA R2,NAM4	VAR4 name pointer
00016E	5020	D3C4		003C4	2687	ST R2,NAMP4	
000172	4120	0100		00100	2688	LA R2,L'VAL4	VAR4 value length
000176	5020	D4C8		004C8	2689	ST R2,VALL4	
00017A	4120	D3C8		003C8	2690	LA R2,VAL4	VAR4 value pointer
00017E	5020	D4CC		004CC	2691	ST R2,VALP4	
					2692 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					2694	*	*/*****/	
					2695	*	/* Initialize VARBLK */	
					2696	*	*/*****/	
000182	4120	D4D0	004D0		2697		LA R2,VARBLK	Blank out VARBLK
000186	4130	0639	00639		2698		LA R3,VARBLKL	
				0018A	2699	SPCVARB	EQU *	
00018A	9240	2000	00000		2700		MVI 0(R2),C' '	
00018E	4122	0001	00001		2701		LA R2,1(R2)	
000192	4630	A18A	0018A		2702		BCT R3,SPCVARB	
					2703	*	*/*****/	
					2704	*	/* Initialize WORKBLK */	
					2705	*	*/*****/	
000196	4120	DB09	00B09		2706		LA R2,WORDBLK	Blank out WORDBLK
00019A	4130	0020	00020		2707		LA R3,WORDBLKL	
				0019E	2708	SPCWORDB	EQU *	
00019E	9240	2000	00000		2709		MVI 0(R2),C' '	
0001A2	4122	0001	00001		2710		LA R2,1(R2)	
0001A6	4630	A19E	0019E		2711		BCT R3,SPCWORDB	
					2712	*	*/*****/	
					2713	*	/* Initialize BLANKS */	
					2714	*	*/*****/	
0001AA	9240	DBBA	00BBA		2715		MVI BLANKS,C' '	Blank out for Case Translation
0001AE	D2FE	DBBB	DBBA	00BBB	2716		MVC BLANKS+1(L'BLANKS-1),BLANKS	
					2717	*	*/*****/	
					2718	*	/* Initialize EX Instructions */	
					2719	*	*/*****/	
0001B4	D205	DCBA	CF2E	00CBA	2720		MVC MVCVARWK(MVCVARWL),MVCVARW#	
0001BA	D205	DCC0	CF28	00CC0	2721		MVC MVCVAR1(MVCVAR1L),MVCVAR1#	
0001C0	D205	DCC6	CF34	00CC6	2722		MVC UPPERCS(UPPERCSL),UPPERCS#	
0001C6	D205	DCCC	CF3A	00CCC	2723		MVC MVCR8\$6(MVCR8\$6L),MVCR8\$6#	
0001CC	D205	DCD2	CF40	00CD2	2724		MVC EXTRLC(EXTRLCL),EXTRLC#	
0001D2	D205	DCD8	CF46	00CD8	2725		MVC SRCHCLC(SRCHCLCL),SRCHCLC#	
0001D8	D205	DCDE	CF4C	00CDE	2726		MVC REVRS(REVRSL),REVRS#	
					2727	*		
0001DE	D703	DB3C	DB3C	00B3C	2728		XC VARSET1L,VARSET1L	
					2729	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					2731	*	/* ***** */	
					2732	*	/* Get PARAMTER information passed to program */	
					2733	*	/* - R1 myPARMS address on entry */	
					2734	*	/* - R4 Starting parms address */	
					2735	*	/* - R6 Starting parms length */	
					2736	*	/* - R3, R5 Working Register */	
					2737	*	/* ***** */	
0001E4	1211				2738		LTR R1,R1	Do I have a PARM?
0001E6	4780	B826		01826	2739		BZ ERR4004	NO, EXIT with ERR4004
0001EA	1831				2740		LR R3,R1	PARM/CPPL addr
0001EC	9180	3000		00000	2741		TM 0(R3),X'80'	Is it PARM or CPPL addr?
0001F0	4780	A20A		0020A	2742		BZ CPPL\$IN	YES, CPPL addr
				001F4	2743	PARM\$IN	EQU *	NO, PARM addr
0001F4	5840	3000		00000	2744		L R4,0(,R3)	Addr of PARM
0001F8	4860	4000		00000	2745		LH R6,0(,R4)	Length of PARM
0001FC	1266				2746		LTR R6,R6	PARM > 0 length?
0001FE	4780	B826		01826	2747		BZ ERR4004	NO, EXIT with ERR4004
000202	4140	4002		00002	2748		LA R4,2(,R4)	YES, point to start of PARM data
000206	47F0	A236		00236	2749		B DOPARMS	and continue...
				0020A	2750	CPPL\$IN	EQU *	
				00000	2751		USING CPPL,R1	CPPL, tell assembler
00020A	D203	D07C	1004	0007C	00004		MVC MYUPT,CPPLUPT	Save UPT
000210	D203	D080	100C	00080	0000C		MVC MYECT,CPPLECT	Save ECT
					2754		DROP R1	Drop R1
000216	5840	3000		00000	2755		L R4,0(,R3)	Command Buffer addr
00021A	4850	4002		00002	2756		LH R5,2(,R4)	Offset to Command Variables
00021E	1255				2757		LTR R5,R5	Any Variables?
000220	4780	B842		01842	2758		BZ ERR4008	NO, EXIT with RC = 8
000224	4860	4000		00000	2759		LH R6,0(,R4)	Length of Command Buffer
000228	1B65				2760		SR R6,R5	Subtract variable offset
00022A	4B60	C85C		0285C	2761		SH R6,=H'4'	Subtract prefix
00022E	4740	B842		01842	2762		BM ERR4008	EXIT with RC = 8 IF NO VARIABLES
000232	4144	5004		00004	2763		LA R4,4(R4,R5)	Point to variables start addr
				00236	2764	DOPARMS	EQU *	
000236	5010	D050		00050	2765		ST R1,MYR1	My PARM address on entry
00023A	5040	D054		00054	2766		ST R4,MYR4	My Starting Parms address
00023E	5060	D058		00058	2767		ST R6,MYR6	My Starting Parms length
					2768	*	/* ***** */	
					2769	*	/* Uppercase translation of PARMIN */	
					2770	*	/* - R4 Starting parms address */	
					2771	*	/* - R6 Parm Length */	
					2772	*	/* - R2, R7 Working Register */	
					2773	*	/* ***** */	
				00242	2774	UP#EM	EQU *	
000242	1824				2775		LR R2,R4	Start addr - PARMS
000244	1876				2776		LR R7,R6	Length of - PARMS
000246	0670				2777		BCTR R7,0	Adjust for EXecute
000248	4470	DCC6		00CC6	2778		EX R7,UPPERCS	Execute Uppercase translate FUNCT
					2779	*UPPERCS	OC 0(0,R2),BLANKS	EBCDIC Lower-Upper Case Translate
					2780	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					2782	*	*/*****/	
					2783	*	/* Get WORDS from PARM data	*/
					2784	*	/* Each WORD is a MAX of 8 bytes	*/
					2785	*	/* Process SETVAR keyword	*/
					2786	*	/* - R4 Starting parms address	*/
					2787	*	/* - R6 Starting parms length	*/
					2788	*	/* - R7, R8, R5, R3, R2, R0 Working Registers	*/
					2789	*	*/*****/	
00024C	4170	0004	00004		2790		LA R7,MAXWD	
000250	5070	D210	00210		2791		ST R7,MAXWORDS	Initialize MAXWORDS
000254	4170	0008	00008		2792		LA R7,WORDLEN	
000258	5070	D20C	0020C		2793		ST R7,WORDL	Initialize WORKLENGTH
00025C	4170	DB09	00B09		2794		LA R7,WORDBLK	Addr of WORDBLK r10->r0
000260	1884				2795		LR R8,R4	Copy addr of PARM data
000262	1B22				2796		SR R2,R2 r9	Count WORD length
000264	4150	0001	00001		2797		LA R5,1 r11	WORD Count
000268	4130	DB2C	00B2C		2798		LA R3,WORDBLKL	Addr of WORDBLKL Lengths
			0026C		2799	GETWORDS	EQU *	
00026C	1266				2800		LTR R6,R6	
00026E	4780	B826	01826		2801		BZ ERR4004	
000272	9540	8000	00000		2802		CLI 0(R8),C' '	WORD delimiter in PARM?
000276	4780	A294	00294		2803		BE NEXTWORD	YES, Bump to next word in WORDBLK
00027A	4122	0001	00001		2804		LA R2,1(R2) r9	NO, Bump up WORD length
00027E	5920	D20C	0020C		2805		C R2,WORDL r9	Word LENGTH > 8?
000282	4720	B834	01834		2806		BH ERR4005	Yes, EXIT with RC = 5
000286	D200	7000 8000	00000 00000		2807		MVC 0(1,R7),0(R8)	Move PARM data to WORDBLK
00028C	4177	0001	00001		2808		LA R7,1(R7)	Bump to next WORD char
000290	47F0	A362	00362		2809		B NEXTCHAR	Bump to next PARM char
			00294		2810	NEXTWORD	EQU *	
000294	1222				2811		LTR R2,R2	Started moving new WORD?
000296	4780	A362	00362		2812		BE NEXTCHAR	No, look at next PARM char
00029A	4155	0001	00001		2813		LA R5,1(R5) r11	Add 1 to WORD count
00029E	5950	D210	00210		2814		C R5,MAXWORDS r11	More than MAX words?
0002A2	4720	B850	01850		2815		BH ERR4009	YES, Too Many WORDS
0002A6	5950	C7DC	027DC		2816		C R5,=F'3'	3rd WORD?
0002AA	4770	A350	00350		2817		BNE NXTCONT	NO, continue
0002AE	D507	DB09 C5C0	00B09 025C0		2818		CLC FUNCT,=C'SETVAR '	YES. Is it SETVAR function?
0002B4	4770	A350	00350		2819		BNE NXTCONT	NO, continue

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				002B8	2821	DOSETVAR	EQU *	
0002B8	9012	D218	00218		2822	STM	R1,R2,DW	SAVE R1,R2 to DW before TPUT
0002BC	4110	C7E0	027E0		2823	LA	R1,=C'-- SETVAR word 3	
0002C0	4120	0014	00014		2824	LA	R2,20	
					2825	TPUT	(R1),(R2)	
0002CC	9812	D218	00218		2831	LM	R1,R2,DW	RESTORE R1,R2 from DW after TPUT
0002D0	4170	DA09	00A09		2832	LA	R7,VARSET1	R7=VAL1
0002D4	5060	DB3C	00B3C		2833	ST	R6,VARSET1L	Store LEN of VARSET
0002D8	D703	DB3C	DB3C	00B3C	2834	XC	VARSET1L,VARSET1L	Set VARSET1L to ZERO
				002DE	2835	FNXTBLNK	EQU *	Find next non-blank after WORD
0002DE	9540	8000	00000		2836	CLI	0(R8),C' '	Blank ?
0002E2	4770	A2F2	002F2		2837	BNE	SV1CHKQB	No, check for QUOTE
0002E6	4188	0001	00001		2838	LA	R8,1(R8)	Yes, increment and
0002EA	4660	A2DE	002DE		2839	BCT	R6,FNXTBLNK	... keep checking
0002EE	47F0	B85E	0185E		2840	B	ERR4010	Error - No SETVAR content found
				002F2	2841	SV1CHKQB	EQU *	Check for QUOTE at begining
0002F2	957D	8000	00000		2842	CLI	0(R8),C''''	Is it QUOTE?
0002F6	4770	A300	00300		2843	BNE	SV1NOQ	No, save LEN and move data
0002FA	4188	0001	00001		2844	LA	R8,1(R8)	Yes, increment to next location
0002FE	0660				2845	BCTR	R6,0	Adjust LEN
				00300	2846	SV1NOQ	EQU *	
000300	5060	DB3C	00B3C		2847	ST	R6,VARSET1L	Store LEN of VARSET
				00304	2848	MOVVAR1L	EQU *	
000304	5960	C7F4	027F4		2849	C	R6,=F'1'	Last position in PARM data?
000308	4770	A314	00314		2850	BNE	FFF1	NO, keep moving data
00030C	957D	8000	00000		2851	CLI	0(R8),C''''	Is is a QUOTE?
000310	4780	A32A	0032A		2852	BE	FFF2	Yes
				00314	2853	FFF1	EQU *	No
000314	D200	7000	8000	00000	2854	MVC	0(1,R7),0(R8)	
00031A	4177	0001	00001		2855	LA	R7,1(R7)	Bump to next PARM char
00031E	4188	0001	00001		2856	LA	R8,1(R8)	Bump to next PARM char
000322	4660	A304	00304		2857	BCT	R6,MOVVAR1L	Do PARM IN Length
000326	47F0	A334	00334		2858	B	FFF3	
				0032A	2859	FFF2	EQU *	
00032A	5860	DB3C	00B3C		2860	L	R6,VARSET1L	Adjust length of VARSET
00032E	0660				2861	BCTR	R6,0	...variable
000330	5060	DB3C	00B3C		2862	ST	R6,VARSET1L	Store LEN of VARSET
				00334	2863	FFF3	EQU *	
000334	9012	D218	00218		2864	STM	R1,R2,DW	SAVE R1,R2 to DW before TPUT
000338	4110	DA09	00A09		2865	LA	R1,VARSET1	
00033C	5820	DB3C	00B3C		2866	L	R2,VARSET1L	
					2867	TPUT	(R1),(R2)	
000348	9812	D218	00218		2873	LM	R1,R2,DW	RESTORE R1,R2 from DW after TPUT
00034C	47F0	A36A	0036A		2874	B	CONT00L	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				00350	2876	NXTCONT	EQU *	
000350	5023	0000	00000		2877		ST R2,0(R3) r9	Store WORD length
000354	4100	0008	00008		2878		LA R0,8 r10	NO, position to next WORD
000358	1B02				2879		SR R0,R2 r10	
00035A	1A70				2880		AR R7,R0 r10	Initialize to start of next WORD
00035C	1B22				2881		SR R2,R2 r9 r9	Reset WORD length count
00035E	4133	0004	00004		2882		LA R3,4(R3)	Bump to next WORD length store
				00362	2883	NEXTCHAR	EQU *	
000362	4188	0001	00001		2884		LA R8,1(R8)	Bump to next PARM char
000366	4660	A26C	0026C		2885		BCT R6,GETWORDS	r5Do PARM IN Lenght
				0036A	2886	CONT00L	EQU *	
00036A	5023	0000	00000		2887		ST R2,0(R3) r9	Store WORD length - last one
00036E	5860	D058	00058		2888		L R6,MYR6	Restore Starting Parm Length
					2889	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					2891	*	*/*****	
					2892	*	/* Process function	
					2893	*	*/*****	
				00372	2894	PRCFUNCT	EQU *	
000372	D507	DB09	C5C8	00B09 025C8	2895	CLC	FUNCT,=C'ABOUT	' *00 About, myStamp
000378	4780	A628		00628	2896	BE	PABOUT	
00037C	D507	DB09	C5D0	00B09 025D0	2897	CLC	FUNCT,=C'LTRIM	' *01 Trim Leading spaces
000382	4780	B3E4		013E4	2898	BE	PLSTRIP	
000386	D507	DB09	C5D8	00B09 025D8	2899	CLC	FUNCT,=C'RTRIM	' *02 Trim Trailing spaces
00038C	4780	B446		01446	2900	BE	PRSTRIP	
000390	D507	DB09	C5E0	00B09 025E0	2901	CLC	FUNCT,=C'TRIM	' *03 Trim Leading/Trailing spaces
000396	4780	B4AE		014AE	2902	BE	PSTRIP	
00039A	D507	DB09	C5E8	00B09 025E8	2903	CLC	FUNCT,=C'TRIM\$	' *04 Trim Leading/Trailing/Dup spaces
0003A0	4780	A64A		0064A	2904	BE	PTRIM\$	
0003A4	D507	DB09	C5F0	00B09 025F0	2905	CLC	FUNCT,=C'INDEX	' *05 Alias FIND
0003AA	4780	A9EA		009EA	2906	BE	PCOUNT	
0003AE	D507	DB09	C5F8	00B09 025F8	2907	CLC	FUNCT,=C'INDEXB	' *06 Alias FINDL
0003B4	4780	A9EA		009EA	2908	BE	PCOUNT	
0003B8	D507	DB09	C600	00B09 02600	2909	CLC	FUNCT,=C'ISNUM	' *07 ISnumeric
0003BE	4780	A6C2		006C2	2910	BE	PISNUM	
0003C2	D507	DB09	C608	00B09 02608	2911	CLC	FUNCT,=C'ISALPHA	' *08 ISalpha
0003C8	4780	A6E8		006E8	2912	BE	PISALPHA	
0003CC	D507	DB09	C610	00B09 02610	2913	CLC	FUNCT,=C'ISLOWER	' *09 ISlower
0003D2	4780	A6E8		006E8	2914	BE	PISLOWER	
0003D6	D507	DB09	C618	00B09 02618	2915	CLC	FUNCT,=C'ISUPPER	' *10 ISupper
0003DC	4780	A6E8		006E8	2916	BE	PISUPPER	
0003E0	D507	DB09	C620	00B09 02620	2917	CLC	FUNCT,=C'ISBLANK	' *11 ISblank
0003E6	4780	A7FC		007FC	2918	BE	PISBLANK	
0003EA	D507	DB09	C628	00B09 02628	2919	CLC	FUNCT,=C'ISDSN	' *12 ISdsn
0003F0	4780	A826		00826	2920	BE	PISDSN	
0003F4	D507	DB09	C630	00B09 02630	2921	CLC	FUNCT,=C'ECHO	' *13 ECHO to terminal
0003FA	4780	A914		00914	2922	BE	PECHO	
0003FE	D507	DB09	C638	00B09 02638	2923	CLC	FUNCT,=C'ECHOQ	' *14 ECHO to terminal w quotes
000404	4780	A914		00914	2924	BE	PECHO	
000408	D507	DB09	C640	00B09 02640	2925	CLC	FUNCT,=C'REVRS	' *15 Translate to Reverse
00040E	4780	A998		00998	2926	BE	PREVRS	
000412	D507	DB09	C648	00B09 02648	2927	CLC	FUNCT,=C'UPPERC	' *16 Upper Case Translate
000418	4780	A9B6		009B6	2928	BE	PUPPERC	
00041C	D507	DB09	C650	00B09 02650	2929	CLC	FUNCT,=C'LOWERC	' *17 Lower Case Translate
000422	4780	A9B6		009B6	2930	BE	PLOWERC	
000426	D507	DB09	C658	00B09 02658	2931	CLC	FUNCT,=C'COUNT	' *18 COUNT occurrence of substr
00042C	4780	A9EA		009EA	2932	BE	PCOUNT	
000430	D507	DB09	C660	00B09 02660	2933	CLC	FUNCT,=C'FINDALL	' *18 Alias COUNT
000436	4780	A9EA		009EA	2934	BE	PCOUNT	
00043A	D507	DB09	C668	00B09 02668	2935	CLC	FUNCT,=C'FINDL	' *19 Find last occurrence of substr
000440	4780	A9EA		009EA	2936	BE	PCOUNT	
000444	D507	DB09	C670	00B09 02670	2937	CLC	FUNCT,=C'FIND	' *20 Find first occurrence of substr
00044A	4780	A9EA		009EA	2938	BE	PCOUNT	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
00044E	D507	DB09	C678	00B09	02678	2940	CLC	FUNCT,=C'CENTER ' *21 Center string content
000454	4780	AB18		00B18		2941	BE	PCENTER
000458	D507	DB09	C680	00B09	02680	2942	CLC	FUNCT,=C'LJUST ' *22 Left Justify content
00045E	4780	AB90		00B90		2943	BE	PLJUST
000462	D507	DB09	C688	00B09	02688	2944	CLC	FUNCT,=C'RJUST ' *23 Right Justify content
000468	4780	ABD4		00BD4		2945	BE	PRJUST
00046C	D507	DB09	C690	00B09	02690	2946	CLC	FUNCT,=C'ZFILL ' *24 Zero fill and right justify
000472	4780	ABD4		00BD4		2947	BE	PRJUST
000476	D507	DB09	C698	00B09	02698	2948	CLC	FUNCT,=C'WORDS ' *25 Count of WORDS in string
00047C	4780	AC40		00C40		2949	BE	PWORDS
000480	D507	DB09	C6A0	00B09	026A0	2950	CLC	FUNCT,=C'GEN# ' *26 Generate 3 digit number
000486	4780	AC90		00C90		2951	BE	PGEN#
00048A	D507	DB09	C6A8	00B09	026A8	2952	CLC	FUNCT,=C'DD2DSN ' *27 DD to DSN
000490	4780	AE02		00E02		2953	BE	PDD2DSN
000494	D507	DB09	C6B0	00B09	026B0	2954	CLC	FUNCT,=C'JOBINFO ' *28 JOB INFORMATION
00049A	4780	AEBE		00EBE		2955	BE	PJOBINFO
						2956	CLC	FUNCT,=C'SETVAR ' SET VAR1
						2957	BE	PSETVAR
00049E	D507	DB09	C6B8	00B09	026B8	2958	CLC	FUNCT,=C'DAYSMM ' *29 Days in Month
0004A4	4780	AF30		00F30		2959	BE	PDAYSMM
0004A8	D507	DB09	C6C0	00B09	026C0	2960	CLC	FUNCT,=C'DAYSYY ' *30 Days in Year
0004AE	4780	AFDC		00FDC		2961	BE	PDAYSYY
0004B2	D507	DB09	C6C8	00B09	026C8	2962	CLC	FUNCT,=C'ISLEAP ' *31 ISleap year
0004B8	4780	AFDC		00FDC		2963	BE	PDAYSYY

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
0004BC	D507	DB09	C6D0	00B09	026D0	2965	CLC	FUNCT,=C'CYJ-D8 ' *32 Julian to Monthname-Day-Year
0004C2	4780	B0AE		010AE		2966	BE	PJDATE
0004C6	D507	DB09	C6D8	00B09	026D8	2967	CLC	FUNCT,=C'CYJ-DAY ' *33 Julian to Day Name
0004CC	4780	B0AE		010AE		2968	BE	PJDAY
0004D0	D507	DB09	C6E0	00B09	026E0	2969	CLC	FUNCT,=C'CYJ-DOW ' *34 Julian to Day of Week Number
0004D6	4780	B0AE		010AE		2970	BE	PJDOW
0004DA	D503	DB09	C7F8	00B09	027F8	2971	CLC	FUNCT(4),=C'CYJ-' *35 Julian to MMDDCCYY (any order)
0004E0	4780	B068		01068		2972	BE	PJMDCY
0004E4	D507	DB09	C6E8	00B09	026E8	2973	CLC	FUNCT,=C'JCY-D8 ' *36 Julian to Monthname-Day-Year
0004EA	4780	B0AE		010AE		2974	BE	PJDATE
0004EE	D507	DB09	C6F0	00B09	026F0	2975	CLC	FUNCT,=C'JCY-DAY ' *37 Julian to Day Name
0004F4	4780	B0AE		010AE		2976	BE	PJDAY
0004F8	D507	DB09	C6F8	00B09	026F8	2977	CLC	FUNCT,=C'JCY-DOW ' *38 Julian to Day of Week Number
0004FE	4780	B0AE		010AE		2978	BE	PJDOW
000502	D503	DB09	C7FC	00B09	027FC	2979	CLC	FUNCT(4),=C'JCY-' *39 Julian to MMDDCCYY (any order)
000508	4780	B068		01068		2980	BE	PJMDCY
00050C	D507	DB09	C700	00B09	02700	2981	CLC	FUNCT,=C'MDCY-D8 ' *40 MDCY to Monthname-Day-Year
000512	4780	B23C		0123C		2982	BE	PMDATE
000516	D507	DB09	C708	00B09	02708	2983	CLC	FUNCT,=C'MDCY-DAY' *41 MDCY to Day Name
00051C	4780	B23C		0123C		2984	BE	PMDAY
000520	D507	DB09	C710	00B09	02710	2985	CLC	FUNCT,=C'MDCY-DOW' *42 MDCY to Day of Week Number
000526	4780	B23C		0123C		2986	BE	PMDOW
00052A	D504	DB09	C874	00B09	02874	2987	CLC	FUNCT(5),=C'MDCY-' *43 MDCY to CCYYJJJ (any order)
000530	4780	B1FE		011FE		2988	BE	PMDCYJ
000534	D507	DB09	C718	00B09	02718	2989	CLC	FUNCT,=C'DMCY-D8 ' *44 DMCY to Monthname-Day-Year
00053A	4780	B23C		0123C		2990	BE	PMDATE
00053E	D507	DB09	C720	00B09	02720	2991	CLC	FUNCT,=C'DMCY-DAY' *45 DMCY to Day Name
000544	4780	B23C		0123C		2992	BE	PMDAY
000548	D507	DB09	C728	00B09	02728	2993	CLC	FUNCT,=C'DMCY-DOW' *46 DMCY to Day of Week Number
00054E	4780	B23C		0123C		2994	BE	PMDOW
000552	D504	DB09	C879	00B09	02879	2995	CLC	FUNCT(5),=C'DMCY-' *47 DMCY to CCYYJJJ (any order)
000558	4780	B1FE		011FE		2996	BE	PMDCYJ
00055C	D507	DB09	C730	00B09	02730	2997	CLC	FUNCT,=C'CYMD-D8 ' *48 CYMD to Monthname-Day-Year
000562	4780	B23C		0123C		2998	BE	PMDATE
000566	D507	DB09	C738	00B09	02738	2999	CLC	FUNCT,=C'CYMD-DAY' *49 CYMD to Day Name
00056C	4780	B23C		0123C		3000	BE	PMDAY
000570	D507	DB09	C740	00B09	02740	3001	CLC	FUNCT,=C'CYMD-DOW' *50 CYMD to Day of Week Number
000576	4780	B23C		0123C		3002	BE	PMDOW
00057A	D504	DB09	C87E	00B09	0287E	3003	CLC	FUNCT(5),=C'CYMD-' *51 CYMD to CCYYJJJ (any order)
000580	4780	B1FE		011FE		3004	BE	PMDCYJ
000584	D507	DB09	C748	00B09	02748	3005	CLC	FUNCT,=C'CYDM-D8 ' *52 CYDM to Monthname-Day-Year
00058A	4780	B23C		0123C		3006	BE	PMDATE
00058E	D507	DB09	C750	00B09	02750	3007	CLC	FUNCT,=C'CYDM-DAY' *53 CYDM to Day Name
000594	4780	B23C		0123C		3008	BE	PMDAY
000598	D507	DB09	C758	00B09	02758	3009	CLC	FUNCT,=C'CYDM-DOW' *54 CYDM to Day of Week Number
00059E	4780	B23C		0123C		3010	BE	PMDOW
0005A2	D504	DB09	C883	00B09	02883	3011	CLC	FUNCT(5),=C'CYDM-' *55 CYDM to CCYYJJJ (any order)
0005A8	4780	B1FE		011FE		3012	BE	PMDCYJ

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
0005AC	D507	DB09	C760	00B09	02760	3014	CLC	FUNCT,=C'FILL ' *56 FILL with delimiter
0005B2	4780	B2BC		012BC		3015	BE	PFILL
0005B6	D507	DB09	C768	00B09	02768	3016	CLC	FUNCT,=C'LSTRIP ' *57 LSTRIP delimiter (leading)
0005BC	4780	B3E4		013E4		3017	BE	PLSTRIP
0005C0	D507	DB09	C770	00B09	02770	3018	CLC	FUNCT,=C'RSTRIP ' *58 RSTRIP delimiter (trailing)
0005C6	4780	B446		01446		3019	BE	PRSTRIP
0005CA	D507	DB09	C778	00B09	02778	3020	CLC	FUNCT,=C'STRIP ' *59 STRIP delimiter (lead/trail)
0005D0	4780	B4AE		014AE		3021	BE	PSTRIP
0005D4	D507	DB09	C780	00B09	02780	3022	CLC	FUNCT,=C'STRIP\$ ' *60 STRIP delimiter (all)
0005DA	4780	B532		01532		3023	BE	PSTRIP\$
0005DE	D507	DB09	C788	00B09	02788	3024	CLC	FUNCT,=C'CONCAT ' *61 Concatenate two variables
0005E4	4780	B598		01598		3025	BE	PCONCAT
0005E8	D507	DB09	C790	00B09	02790	3026	CLC	FUNCT,=C'UNSTR ' *62 UNSTRing into variables
0005EE	4780	B60E		0160E		3027	BE	PUNSTR
0005F2	D507	DB09	C798	00B09	02798	3028	CLC	FUNCT,=C'REPLACE ' *63 REPLACE str with st2
0005F8	4780	B694		01694		3029	BE	PREPL
0005FC	D507	DB09	C7A0	00B09	027A0	3030	CLC	FUNCT,=C'VEXIST ' *64 VEXIST variable exist
000602	4780	B7A6		017A6		3031	BE	PVEXIST
000606	D507	DB09	C7A8	00B09	027A8	3032	CLC	FUNCT,=C'TDSN ' *65 TDSN temporary DSN nodes
00060C	4780	AC90		00C90		3033	BE	PTDSN
000610	D507	DB09	C7B0	00B09	027B0	3034	CLC	FUNCT,=C'NOW ' *66 NOW current date time info
000616	4780	AC90		00C90		3035	BE	PNOW
00061A	D507	DB09	C7B8	00B09	027B8	3036	CLC	FUNCT,=C'PAD ' *67 PAD Pad current string
000620	4780	B2BC		012BC		3037	BE	PPAD
				00624	3038	PRCFUNCX	EQU	*
000624	47F0	B86C		0186C	3039		B	ERR4011 Invalid function *****
					3040	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3042	*	*/*****	
					3043	*00	/* Function: ABOUT - -	
					3044	*	*/*****	
					3045	*	/* Display mystamp	
					3046	*	/* - R1, R2, R5 Working Register	
					3047	*	*/*****	
					3048	*	/* - RC=0 Successful Request	
					3049	*	*/*****	
			00628		3050	PABOUT	EQU * Process ABOUT	
					3051	*	/-----	
					3052	*	/* Display mySTAMP info on terminal	
					3053	*	/-----	
000628	4150	0034	00034		3054		LA R5,MYSTAMPL Load length of mySTAMP	
00062C	4155	0004	00004		3055		LA R5,4(R5) Adj length for header (+4)	
000630	4050	D0A4	000A4		3056		STH R5,OUTTXT	
000634	D201	D0A6	C85E 000A6	0285E	3057		MVC OUTTXT+2(2),=X'0000' Init reserved fld to x'0'	
00063A	D233	D0A8	A01C 000A8	0001C	3058		MVC INFOLINE(MYSTAMPL),MYSTAMP	
000640	4550	BDF0	01DF0		3059		BAL R5,S#PUTL Do PUTLINE	
			00644		3060	PABOUTX	EQU *	
000644	1BFF				3061		SR R15,R15	
000646	47F0	B7E2	017E2		3062		B ERRDONE Successful Request	
					3063	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3065	*	/* ***** */	
					3066	*04	/* Function: TRIM\$ REQ OPT */	
					3067	*	/* ***** */	
					3068	*	/* Strip leading and trailing spaces including remaining */	
					3069	*	/* duplicate spaces in string */	
					3070	*	/* - R2, R5 Working Register */	
					3071	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1) */	
					3072	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1) */	
					3073	*	/* ***** */	
					3074	*	/* - RC=0 Successful Request */	
					3075	*	/* ***** */	
00064A	4550	BA3E	01A3E	0064A	3076	PTRIM\$	EQU *	Process TRIM\$\$
00064E	4550	BC4A	01C4A		3077		BAL R5,S#GVAR1	Get VAR1
000652	4120	D6E4	006E4		3078		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
000656	1B55				3079		LA R2,VARWORK	
					3080		SR R5,R5	VARWORK Length
					3081	*	/* ----- */	
					3082	*	/* Eliminate repeating blanks in string */	
					3083	*	/* ----- */	
000658	D200	2000	8000	00000	00658	3084	TRIML\$ EQU *	
00065E	4155	0001	00001	00000	3085		MVC 0(1,R2),0(R8)	
000662	9540	8000	00000		3086		LA R5,1(R5)	Add 1 to length - VARWORK
000666	4780	A66E	0066E		3087		CLI 0(R8),C' '	Blank character moved?
00066A	47F0	A682	00682		3088		BE SKIPBLK\$	YES, skip subsequent blanks
					3089		B BUMPV\$	NO, move more data
00066E	4188	0001	00001	0066E	3090	SKIPBLK\$	EQU *	
000672	9540	8000	00000		3091		LA R8,1(R8)	Bump to next position - VAR1
000676	4770	A686	00686		3092		CLI 0(R8),C' '	Blank character?
00067A	4670	A66E	0066E		3093		BNE BUMPVW\$	NO, go move more data
00067E	47F0	A68E	0068E		3094		BCT R7,SKIPBLK\$	YES, skip it and check next pos
					3095		B BUMPEND\$	
000682	4188	0001	00001	00682	3096	BUMPV\$	EQU *	
					3097		LA R8,1(R8)	Bump to next position - VAR1
000686	4122	0001	00001	00686	3098	BUMPVW\$	EQU *	
00068A	4670	A658	00658		3099		LA R2,1(R2)	Bump to next position - VARWORK
					3100		BCT R7,TRIML\$	Decrement length while BLANK byte
00068E	1875			0068E	3101	BUMPEND\$	EQU *	
					3102		LR R7,R5	Load revised length for VAR1
					3103	*	/* ----- */	
					3104	*	/* At this point, VARWORK contains text with only one */	
					3105	*	/* blank between content i.e. ' this is my content ' */	
					3106	*	/* ----- */	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3108	*	*	/-----/
					3109	*	*	/* If last byte is blank, decrement length */
					3110	*	*	/-----/
				00690	3111	CHKLA\$T	EQU *	
000690	4120	D6E4	006E4		3112	LA	R2,VARWORK	Start addr - VARWORK
000694	1A27				3113	AR	R2,R7	Add length to start addr - VARWORK
000696	0620				3114	BCTR	R2,0	.. make adjustment, end addr
000698	9540	2000	00000		3115	CLI	0(R2),C' '	Last position blank?
00069C	4770	A6A2	006A2		3116	BNE	CHKFIR\$T	NO, check starting position
0006A0	0670				3117	BCTR	R7,0	YES, adjust length
					3118	*	*	/-----/
					3119	*	*	/* If first byte is blank, move string left by 1 */
					3120	*	*	/-----/
				006A2	3121	CHKFIR\$T	EQU *	
0006A2	4120	D6E4	006E4		3122	LA	R2,VARWORK	Start addr - VARWORK
0006A6	9540	2000	00000		3123	CLI	0(R2),C' '	First position blank?
0006AA	4770	A6BA	006BA		3124	BNE	PTRIM\$X	NO, done...
					3125	*		YES, adjust and reseed data
0006AE	4122	0001	00001		3126	LA	R2,1(R2)	Move start addr - VARWORK
0006B2	1882				3127	LR	R8,R2	Load start addr for EX
0006B4	0670				3128	BCTR	R7,0	Adjust for EXecute
0006B6	4470	DCBA	00CBA		3129	EX	R7,MVCVARWK	Execute VARWORK value move
					3130	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
					3131	****	LA R7,1(R7)	Reset after EXecute
					3132	****		No need to adjust R7, as data
					3133	****		is 1 byte less!!
				006BA	3134	PTRIM\$X	EQU *	
0006BA	4550	BBC4	01BC4		3135	BAL	R5,S#UVAR2	Update VAR2
0006BE	47F0	B7E2	017E2		3136	B	ERRDONE	Successful Request
					3137	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3139	*	*/*****	
					3140	*07	/* Function: ISNUM REQ NO	
					3141	*	*/*****	
					3142	*	/* Check string for all numeric (0-9)	
					3143	*	/* - R2, R5 Working Register	
					3144	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					3145	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					3146	*	*/*****	
					3147	*	/* - RC=4000 FALSE	
					3148	*	/* - RC=4001 TRUE	
					3149	*	/* - RC=4022 VAR2 not required	
					3150	*	*/*****	
			006C2		3151	PISNUM	EQU * Process ISNUM	
					3152	*	/-----	
					3153	*	/* No VAR2 content in position 1 needed	
					3154	*	/-----	
0006C2	D507	DB19	C7C0	00B19	027C0	3155	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
0006C8	4780	A6D4		006D4		3156	BE PISNUM#	YES, continue...
0006CC	9540	DB19		00B19		3157	CLI VAR2,C' '	VAR2 name captured?
0006D0	4770	B8F8		018F8		3158	BNE ERR4022	YES, VAR2 not needed
				006D4	3159	PISNUM#	EQU *	
0006D4	4550	BA3E		01A3E		3160	BAL R5,S#GVAR1	Get VAR1
0006D8	4550	BE60		01E60		3161	BAL R5,S#ISNUM	Numeric Test
0006DC	59F0	C800		02800		3162	C R15,=F'4000'	Numeric?
0006E0	4780	B7FC		017FC		3163	BE ERR4000	NO, FALSE
0006E4	47F0	B80A		0180A		3164	B ERR4001	YES, TRUE
					3165	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3167	*	*/*****	*/
					3168	*08	/* Function: ISALPHA	REQ NO */
					3169	*09	/* Function: ISLOWER	REQ NO */
					3170	*10	/* Function: ISUPPER	REQ NO */
					3171	*	*/*****	*/
					3172	*	/* ISALPHA Check string for all alphabetic (A-Z,a-z)	*/
					3173	*	/* ISLOWER Check string for all UPPER alphabetic (A-Z)	*/
					3174	*	/* ISUPPER Check string for all LOWER alphabetic (a-z)	*/
					3175	*	/* - R2, R5	Working Register */
					3176	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3177	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3178	*	*/*****	*/
					3179	*	/* - RC=4000 FALSE	*/
					3180	*	/* - RC=4001 TRUE	*/
					3181	*	/* - RC=4022 VAR2 not required	*/
					3182	*	*/*****	*/
			006E8		3183	PISALPHA	EQU *	Process ISALPHA
			006E8		3184	PISLOWER	EQU *	Process ISLOWER
			006E8		3185	PISUPPER	EQU *	Process ISUPPER
					3186	*	*/-----	*/
					3187	*	/* No VAR2 content in position 1 needed	*/
					3188	*	*/-----	*/
0006E8	D507	DB19	C7C0	00B19	027C0	3189	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
0006EE	4780	A6FA		006FA		3190	BE PISALPH#	YES, continue...
0006F2	9540	DB19		00B19		3191	CLI VAR2,C' '	VAR2 name captured?
0006F6	4770	B8F8		018F8		3192	BNE ERR4022	YES, VAR2 not needed
				006FA	3193	PISALPH#	EQU *	
0006FA	4550	BA3E		01A3E		3194	BAL R5,S#GVAR1	Get VAR1
0006FE	D507	DB09	C610	00B09	02610	3195	CLC FUNCT,=C'ISLOWER '	ISLOWER request?
000704	4780	A76C		0076C		3196	BE PISLOWRL	Yes.
000708	D507	DB09	C618	00B09	02618	3197	CLC FUNCT,=C'ISUPPER '	ISUPPER request?
00070E	4780	A7B4		007B4		3198	BE PISUPPRL	Yes.
					3199	*		Must be ISALPHA request
					3200	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3202	*	*/*****	
					3203	*	/* Check string for all alphabetic (A-Z,a-z)	
					3204	*	*/*****	
000712	0670				3205		BCTR R7,0	Adjust for EXecute
000714	4470	DCBA	00CBA		3206		EX R7,MVCVARWK	Execute VARWORK value move
					3207	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
000718	4120	D6E4	006E4		3208		LA R2,VARWORK	Start addr - VARWORK
00071C	4470	DCC6	00CC6		3209		EX R7,UPPERCS	Execute Uppercase translation
					3210	*UPPERCS	OC 0(0,R2),BLANKS	EBCDIC Lower-Upper Case Translate
000720	4177	0001	00001		3211		LA R7,1(R7)	Reset R7 to original value
				00724	3212	PISALPHL	EQU *	Process ISALPHA
000724	95E9	2000	00000		3213		CLI 0(R2),C'Z'	Byte > Z?
000728	4720	B7FC	017FC		3214		BH ERR4000	YES, not alphabetic, FALSE
00072C	95C1	2000	00000		3215		CLI 0(R2),C'A'	Byte < A?
000730	4740	B7FC	017FC		3216		BL ERR4000	YES, not alphabetic, FALSE
000734	95C9	2000	00000		3217		CLI 0(R2),C'I'	Byte <=I?
000738	47D0	A760	00760		3218		BNH PISLAPNX	Alphabetic
00073C	95D1	2000	00000		3219		CLI 0(R2),C'J'	Byte < J?
000740	4740	B7FC	017FC		3220		BL ERR4000	YES, not alphabetic, FALSE
000744	95D9	2000	00000		3221		CLI 0(R2),C'R'	Byte <=R?
000748	47D0	A760	00760		3222		BNH PISLAPNX	Alphabetic
00074C	95E2	2000	00000		3223		CLI 0(R2),C'S'	Byte < S?
000750	4740	B7FC	017FC		3224		BL ERR4000	YES, not alphabetic, FALSE
000754	95E9	2000	00000		3225		CLI 0(R2),C'Z'	Byte <=Z?
000758	47D0	A760	00760		3226		BNH PISLAPNX	Alphabetic
00075C	47F0	B7FC	017FC		3227		B ERR4000	Not alphabetic, FALSE
				00760	3228	PISLAPNX	EQU *	
000760	4122	0001	00001		3229		LA R2,1(R2)	Bump up to next byte of VARWORK
000764	4670	A724	00724		3230		BCT R7,PISALPHL	LOOP for alphabetic
000768	47F0	B80A	0180A		3231		B ERR4001	All pass alphabetic, TRUE
					3232	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3234	*	*/*****	
					3235	*	/* Check string for all LOWER alphabetic (a-z)	*/
					3236	*	*/*****	
				0076C	3237	PISLOWRL	EQU *	Process ISLOWER
00076C	95A9	8000	00000		3238		CLI 0(R8),C'z'	Byte > z?
000770	4720	B7FC	017FC		3239		BH ERR4000	YES, not low alpha, FALSE
000774	9581	8000	00000		3240		CLI 0(R8),C'a'	Byte < a?
000778	4740	B7FC	017FC		3241		BL ERR4000	YES, not low alpha, FALSE
00077C	9589	8000	00000		3242		CLI 0(R8),C'i'	Byte <=i?
000780	47D0	A7A8	007A8		3243		BNH PISLOWRX	Low alpha
000784	9591	8000	00000		3244		CLI 0(R8),C'j'	Byte < j?
000788	4740	B7FC	017FC		3245		BL ERR4000	YES, not low alpha, FALSE
00078C	9599	8000	00000		3246		CLI 0(R8),C'r'	Byte <=r?
000790	47D0	A7A8	007A8		3247		BNH PISLOWRX	Low alpha
000794	95A2	8000	00000		3248		CLI 0(R8),C's'	Byte < s?
000798	4740	B7FC	017FC		3249		BL ERR4000	YES, not low alpha, FALSE
00079C	95A9	8000	00000		3250		CLI 0(R8),C'z'	Byte <=z?
0007A0	47D0	A7A8	007A8		3251		BNH PISLOWRX	Low alpha
0007A4	47F0	B7FC	017FC		3252		B ERR4000	Not low alpha, FALSE
				007A8	3253	PISLOWRX	EQU *	
0007A8	4188	0001	00001		3254		LA R8,1(R8)	Bump up to next byte of VAR1
0007AC	4670	A76C	0076C		3255		BCT R7,PISLOWRL	LOOP for lowercase alphabetic
0007B0	47F0	B80A	0180A		3256		B ERR4001	All pass lower alphabetic, TRUE
					3257	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3259	*	*/*****	
					3260	*	/* Check string for all UPPER alphabetic (A-Z)	*/
					3261	*	*/*****	
				007B4	3262	PISUPPRL	EQU *	Process ISUPPER
0007B4	95E9	8000	00000		3263		CLI 0(R8),C'Z'	Byte > Z?
0007B8	4720	B7FC	017FC		3264		BH ERR4000	YES, not upperalpha, FALSE
0007BC	95C1	8000	00000		3265		CLI 0(R8),C'A'	Byte < A?
0007C0	4740	B7FC	017FC		3266		BL ERR4000	YES, not upperalpha, FALSE
0007C4	95C9	8000	00000		3267		CLI 0(R8),C'I'	Byte <=I?
0007C8	47D0	A7F0	007F0		3268		BNH PISUPPRX	upperalpha
0007CC	95D1	8000	00000		3269		CLI 0(R8),C'J'	Byte < J?
0007D0	4740	B7FC	017FC		3270		BL ERR4000	YES, not upperalpha, FALSE
0007D4	95D9	8000	00000		3271		CLI 0(R8),C'R'	Byte <=R?
0007D8	47D0	A7F0	007F0		3272		BNH PISUPPRX	upperalpha
0007DC	95E2	8000	00000		3273		CLI 0(R8),C'S'	Byte < S?
0007E0	4740	B7FC	017FC		3274		BL ERR4000	YES, not upperalpha, FALSE
0007E4	95E9	8000	00000		3275		CLI 0(R8),C'Z'	Byte <=Z?
0007E8	47D0	A7F0	007F0		3276		BNH PISUPPRX	upperalpha
0007EC	47F0	B7FC	017FC		3277		B ERR4000	Not upperalpha, FALSE
				007F0	3278	PISUPPRX	EQU *	
0007F0	4188	0001	00001		3279		LA R8,1(R8)	Bump up to next byte of VAR1
0007F4	4670	A7B4	007B4		3280		BCT R7,PISUPPRL	LOOP for uppercase alphabetic
0007F8	47F0	B80A	0180A		3281		B ERR4001	All pass upper alphabetic, TRUE
					3282	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3284	*	*/*****	
					3285	*11	/* Function: ISBLANK REQ NO	
					3286	*	*/*****	
					3287	*	/* Check string for all whitespace (space)	
					3288	*	/* - R2, R5 Working Register	
					3289	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					3290	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					3291	*	*/*****	
					3292	*	/* - RC=4000 FALSE	
					3293	*	/* - RC=4001 TRUE	
					3294	*	/* - RC=4022 VAR2 not required	
					3295	*	*/*****	
			007FC		3296	PISBLANK	EQU * Process ISBLANK	
					3297	*	/-----	
					3298	*	/* No VAR2 content in position 1 needed	
					3299	*	/-----	
0007FC	D507	DB19	C7C0	00B19	027C0	3300	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
000802	4780	A80E		0080E		3301	BE PISBLNK#	YES, continue...
000806	9540	DB19		00B19		3302	CLI VAR2,C' '	VAR2 name captured?
00080A	4770	B8F8		018F8		3303	BNE ERR4022	YES, VAR2 not needed
				0080E	3304	PISBLNK#	EQU *	
00080E	4550	BA3E		01A3E		3305	BAL R5,S#GVAR1	Get VAR1
				00812	3306	PISWHTL	EQU *	
000812	9540	8000		00000		3307	CLI 0(R8),C' '	Byte = SPACE?
000816	4770	B7FC		017FC		3308	BNE ERR4000	NO, not blank, FALSE
00081A	4188	0001		00001		3309	LA R8,1(R8)	Bump up to next byte of VAR1
00081E	4670	A812		00812		3310	BCT R7,PISWHTL	LOOP for blanks
				00822	3311	PISBLANX	EQU *	
000822	47F0	B80A		0180A		3312	B ERR4001	All pass spaces, TRUE
					3313	*	*	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3315	*	*/*****	*/
					3316	*12	/* Function: ISDSN	REQ NO
					3317	*	*/*****	*/
					3318	*	/* Check string for valid dataset name	*/
					3319	*	/* - string 1 to 44 bytes	*/
					3320	*	/* - contain up to 22 segments	*/
					3321	*	/* - segment delimiter is period (.)	*/
					3322	*	/* - each segment 1-8 bytes	*/
					3323	*	/* - char 1-1 in segment A-Z #@\$	*/
					3324	*	/* - char 2-8 in segment A-Z #@\$ 0-9	*/
					3325	*	/* - R2, R5	Working Register
					3326	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3327	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3328	*	*/*****	*/
					3329	*	/* - RC=4000 FALSE	*/
					3330	*	/* - RC=4001 TRUE	*/
					3331	*	/* - RC=4022 VAR2 not required	*/
					3332	*	*/*****	*/
			00826		3333	PISDSN	EQU *	Process ISDSN
					3334	*	/*-----	*/
					3335	*	/* No VAR2 content in position 1 needed	*/
					3336	*	/*-----	*/
000826	D507	DB19	C7C0	00B19	027C0	3337	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
00082C	4780	A838		00838		3338	BE PISDSN#	YES, continue...
000830	9540	DB19		00B19		3339	CLI VAR2,C' '	VAR2 name captured?
000834	4770	B8F8		018F8		3340	BNE ERR4022	YES, VAR2 not needed
				00838	3341	PISDSN#	EQU *	
000838	4550	BA3E		01A3E		3342	BAL R5,S#GVAR1	Get VAR1
00083C	0670					3343	BCTR R7,0	Adjust for EXecute
00083E	4470	DCBA		00CBA		3344	EX R7,MVCVARWK	Execute VARWORK value move
						3345	*MVCVARWK MVC VARWORK(0),0(R8)	Save VARWORK value
000842	4177	0001		00001		3346	LA R7,1(R7)	Reset R7 to original value
000846	4120	D6E4		006E4		3347	LA R2,VARWORK	Start addr - VARWORK
					3348	*	/*-----	*/
					3349	*	/* Check DSN max length 44 bytes	*/
					3350	*	/*-----	*/
00084A	5970	C804		02804		3351	C R7,=F'44'	Var > 44 bytes?
00084E	4720	B7FC		017FC		3352	BH ERR4000	Failed valid length, FALSE
					3353	*	/*-----	*/
					3354	*	/* Start DSN name check	*/
					3355	*	/*-----	*/
000852	F810	D065	C888	00065	02888	3356	ZAP KSEGCHAR,=P'+0'	Segment character count
000858	F810	D067	C888	00067	02888	3357	ZAP KSEGCNT,=P'+0'	Segment count
				0085E		3358	PISDSNC	EQU *
00085E	954B	2000		00000		3359	CLI 0(R2),C'.'	Segment delimiter? 4B
000862	4770	A88A		0088A		3360	BNE DODSNSEG	NO. Test for valid chars
					3361	*		YES. Process segment mark
000866	FA10	D067	C889	00067	02889	3362	AP KSEGCNT,=P'+1'	Bump up segment count
					3363	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					3365 *	*/-----/	
					3366 *	/* Check valid segment length (1-8)	*/
					3367 *	*/-----/	
00086C	F910	D065	C889	00065 02889	3368	CP KSEGCHAR,=P'+1'	Seg length < 1?
000872	4740	B7FC		017FC	3369	BL ERR4000	YES. Invalid seg length, FALSE
000876	F910	D065	C88A	00065 0288A	3370	CP KSEGCHAR,=P'+8'	Seg length > 8?
00087C	4720	B7FC		017FC	3371	BH ERR4000	YES. Invalid seg length, FALSE
000880	F810	D065	C888	00065 02888	3372	ZAP KSEGCHAR,=P'+0'	Reset segment character count
000886	47F0	A8FE		008FE	3373	B PIDSNCX	Bump to next char
				0088A	3374 DODSNSEG EQU *		
					3375 *	*/-----/	
					3376 *	/* Test notional characters	*/
					3377 *	*/-----/	
00088A	955B	2000		00000	3378	CLI 0(R2),C'\$'	Byte = #? 5B
00088E	4780	A8F4		008F4	3379	BE CHKSEG	Valid char, check segment start
000892	9560	2000		00000	3380	CLI 0(R2),C'-'	Byte = #? 60
000896	4780	A8F4		008F4	3381	BE CHKSEG	Valid char, check segment start
00089A	957B	2000		00000	3382	CLI 0(R2),C'#'	Byte = #? 7B
00089E	4780	A8F4		008F4	3383	BE CHKSEG	Valid char, check segment start
0008A2	957C	2000		00000	3384	CLI 0(R2),C'@'	Byte = #? 7C
0008A6	4780	A8F4		008F4	3385	BE CHKSEG	Valid char, check segment start
					3386 *	*/-----/	
					3387 *	/* Test alphabetic characters	*/
					3388 *	*/-----/	
					3389 *	STM R1,R2,DW	SAVE R1,R2 to DW before TPUT
					3390 *	LA R1,=C'-- Test alpha	
					3391 *	LA R2,20	
					3392 *	TPUT (R1),(R2)	
					3393 *	LM R1,R2,DW	RESTORE R1,R2 from DW after TPUT
0008AA	95C1	2000		00000	3394	CLI 0(R2),C'A'	Byte < A? C1
0008AE	4740	B7FC		017FC	3395	BL ERR4000	Not alphabetic, return FALSE
0008B2	95C9	2000		00000	3396	CLI 0(R2),C'I'	Byte <=I? C9
0008B6	47D0	A8F4		008F4	3397	BNH CHKSEG	Valid char, check segment start
0008BA	95D1	2000		00000	3398	CLI 0(R2),C'J'	Byte < J? D1
0008BE	4740	B7FC		017FC	3399	BL ERR4000	Not alphabetic, return FALSE
0008C2	95D9	2000		00000	3400	CLI 0(R2),C'R'	Byte <=R? D9
0008C6	47D0	A8F4		008F4	3401	BNH CHKSEG	Valid char, check segment start
0008CA	95E2	2000		00000	3402	CLI 0(R2),C'S'	Byte < S? E2
0008CE	4740	B7FC		017FC	3403	BL ERR4000	Not alphabetic, return FALSE
0008D2	95E9	2000		00000	3404	CLI 0(R2),C'Z'	Byte <=Z? E9
0008D6	47D0	A8F4		008F4	3405	BNH CHKSEG	Valid char, check segment start
					3406 *	*/-----/	
					3407 *	/* Byte 1 of segment is invalid at this point	*/
					3408 *	*/-----/	
0008DA	F910	D065	C888	00065 02888	3409	CP KSEGCHAR,=P'+0'	Seg length = 0?
0008E0	4780	B7FC		017FC	3410	BE ERR4000	YES, invalid char, return FALSE
					3411 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3413	*	*/-----/	
					3414	*	/* Test numerics	*/
					3415	*	*/-----/	
0008E4	95F0	2000	00000		3416		CLI 0(R2),C'0'	Byte < 0? F0
0008E8	4740	B7FC	017FC		3417		BL ERR4000	Not numeric, return FALSE
0008EC	95F9	2000	00000		3418		CLI 0(R2),C'9'	Byte <= 9? F9
0008F0	47D0	A8F4	008F4		3419		BNH CHKSEG	Valid char, check segment start
				008F4	3420	CHKSEG	EQU *	
0008F4	FA10	D065	C889	00065	02889	3421	AP KSEGCHAR,=P'+1'	Bump up segment char count
0008FA	47F0	A8FE	008FE		3422		B PIDSNCX	Valid char, bump to next char
				008FE	3423	PIDSNCX	EQU *	
0008FE	4122	0001	00001		3424		LA R2,1(R2)	Bump up to next byte of VAR1
000902	4670	A85E	0085E		3425		BCT R7,PISDSNC	LOOP for alphabetic
					3426	*		End of Loop...
000906	F910	D065	C888	00065	02888	3427	CP KSEGCHAR,=P'+0'	Seg length = 0?
00090C	4780	B7FC	017FC		3428		BE ERR4000	YES, invalid char, return FALSE
				00910	3429	PISDSNX	EQU *	
000910	47F0	B80A	0180A		3430		B ERR4001	All pass, TRUE
					3431	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20	
					3433	*	*/*****	*/	
					3434	*13	/* Function: ECHO	REQ NO	
					3435	*14	/* ECHOQ	REQ NO	
					3436	*	*/*****	*/	
					3437	*	/* Display var1 without or with single quotes	*/	
					3438	*	/* - R1, R2, R5	Working Register	
					3439	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/	
					3440	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/	
					3441	*	*/*****	*/	
					3442	*	/* - RC=0	Successful Request	
					3443	*	/* - RC=4022	VAR2 not required	
					3444	*	*/*****	*/	
			00914		3445	PECHO	EQU *	Process ECHO	
					3446	*	/*-----	*/	
					3447	*	/* No	VAR2 content in position 1 needed	
					3448	*	/*-----	*/	
000914	D507	DB19	C7C0	00B19	027C0	3449	CLC	VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
00091A	4780	A926		00926		3450	BE	PECHO#	YES, continue...
00091E	9540	DB19		00B19		3451	CLI	VAR2,C' '	VAR2 name captured?
000922	4770	B8F8		018F8		3452	BNE	ERR4022	YES, VAR2 not needed
				00926	3453	PECHO#	EQU *		
000926	4550	BA3E		01A3E		3454	BAL	R5,S#GVAR1	Get VAR1
00092A	9012	D218		00218		3455	STM	R1,R2,DW	SAVE R1,R2 to DW before TPUT
00092E	95D8	DB0D		00B0D		3456	CLI	FUNCT+4,C'Q'	Is it ECHOQ?
000932	4770	A95A		0095A		3457	BNE	DOECHO	NO, DOECHO
				00936	3458	DOECHOQ	EQU *		YES... DOECHOQ
000936	927D	D4E0		004E0		3459	MVI	VAL1Q,C''''	Put start quote around string
00093A	4110	D4E0		004E0		3460	LA	R1,VAL1Q	
00093E	5A10	D280		00280		3461	A	R1,VALL1	
000942	4111	0001		00001		3462	LA	R1,1(R1)	
000946	927D	1000		00000		3463	MVI	0(R1),C''''	Put end quote around string
00094A	4110	D4E0		004E0		3464	LA	R1,VAL1Q	
00094E	5820	D280		00280		3465	L	R2,VALL1	
000952	4122	0002		00002		3466	LA	R2,2(R2)	
000956	47F0	A962		00962		3467	B	DODISPLY	
					3468	*			

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				0095A	3470	DOECHO	EQU *	
00095A	4110	D4E1	004E1		3471		LA R1,VAL1	
00095E	5820	D280	00280		3472		L R2,VALL1	
				00962	3473	DODISPLY	EQU *	Echo VAR01 content
000962	4020	D0A4	000A4		3474		STH R2,OUTTXT	Set length of VAL1
000966	4920	C860	02860		3475		CH R2,=X'00FF'	GT 256?
00096A	47D0	A974	00974		3476		BNH DOPLADJ	No, jump to adjust
00096E	D201	D0A4	C860 000A4	02860	3477		MVC OUTTXT(2),=X'00FF'	Yes, set length of VAL1 to 256
000974	4122	0004	00004		3478	DOPLADJ	LA R2,4(R2)	Adj length for header (+4)
000978	4020	D0A4	000A4		3479		STH R2,OUTTXT	
00097C	D201	D0A6	C85E 000A6	0285E	3480		MVC OUTTXT+2(2),=X'0000'	Init reserved fld to x'0'
000982	4450	A992	00992		3481		EX R5,MVCPUTL	EX move VAL1 to INFOLINE
000986	4550	BDF0	01DF0		3482		BAL R5,S#PUTL	Do PUTLINE
					3483	*	TPUT (R1),(R2)	
00098A	9812	D218	00218		3484		LM R1,R2,DW	RESTORE R1,R2 from DW after TPUT
				0098E	3485	PECHOX	EQU *	
00098E	47F0	B7E2	017E2		3486		B ERRDONE	Successful Request
					3487	*		
000992	D200	D0A8	1000 000A8	00000	3488	MVCPUTL	MVC INFOLINE(0),0(R1)	Move VAL1 to INFOLINE
					3489	*MVCPUTL	MVC INFOLINE(0),0(R1)	Move VAL1 to INFOLINE

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3491	*	*/*****	
					3492	*15	/* Function: REVRS REQ OPT	
					3493	*	*/*****	
					3494	*	/* Reverse letters and numbers.	
					3495	*	/* A-Z > Z-A, 0-9 > 9-0, others to blanks.	
					3496	*	/* - R2, R5 Working Register	
					3497	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					3498	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					3499	*	*/*****	
					3500	*	/* - RC=0 Successful Request	
					3501	*	*/*****	
				00998	3502	PREVRS	EQU *	Process REVRs
000998	4550	BA3E	01A3E		3503		BAL R5,S#GVAR1	Get VAR1
00099C	4550	BC4A	01C4A		3504		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
0009A0	0670				3505		BCTR R7,0	Adjust for EXecute
0009A2	4470	DCBA	00CBA		3506		EX R7,MVCVARWK	Execute VARWORK value move
					3507	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
0009A6	4470	DCDE	00CDE		3508		EX R7,REVRs	Execute XOR
					3509	*REVRs	TR VARWORK(0),TREVRs	Reverse letters and numbers
0009AA	4177	0001	00001		3510		LA R7,1(R7)	Adjust length to original
0009AE	4550	BBC4	01BC4		3511		BAL R5,S#UVAR2	Update VAR2
				009B2	3512	PREVRsX	EQU *	
0009B2	47F0	B7E2	017E2		3513		B ERRDONE	Successful Request
					3514	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3516	*	*/*****	*/
					3517	*16	/* Function: UPPERC	REQ OPT */
					3518	*17	/* Function: LOWERC	REQ OPT */
					3519	*	*/*****	*/
					3520	*	/* UPPERC Convert string to uppercase	*/
					3521	*	/* LOWERC Convert string to lowercase	*/
					3522	*	/* - R2, R5	Working Register */
					3523	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3524	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3525	*	*/*****	*/
					3526	*	/* - RC=0	Successful Request */
					3527	*	*/*****	*/
			009B6		3528	PUPPERC	EQU *	Process UPPERC
			009B6		3529	PLOWERC	EQU *	Process LOWERC
0009B6	4550	BA3E	01A3E		3530		BAL R5,S#GVAR1	Get VAR1
0009BA	4550	BC4A	01C4A		3531		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
0009BE	0670				3532		BCTR R7,0	Adjust for EXecute
0009C0	4470	DCBA	00CBA		3533		EX R7,MVCVARWK	Execute VARWORK value move
					3534	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
0009C4	4120	D6E4	006E4		3535		LA R2,VARWORK	
0009C8	D507	DB09	C648	00B09	02648		CLC FUNCT,=C'UPPERC	
0009CE	4780	A9DA	009DA		3537		BE DOUPPER	
				009D2	3538	DOLOWER	EQU *	
0009D2	4470	DCD2	00CD2		3539		EX R7,EXTRLC	Execute Lowercase translation
					3540	*EXTRLC	TR 0(0,R2),TRTBL	EBCDIC Upper-Lower Case Translate
0009D6	47F0	A9DE	009DE		3541		B PLOWERCX	Adjust and return
				009DA	3542	DOUPPER	EQU *	
0009DA	4470	DCC6	00CC6		3543		EX R7,UPPERCS	Execute Uppercase translation
					3544	*UPPERCS	OC 0(0,R2),BLANKS	EBCDIC Lower-Upper Case Translate
				009DE	3545	PUPPERCX	EQU *	
				009DE	3546	PLOWERCX	EQU *	
0009DE	4177	0001	00001		3547		LA R7,1(R7)	Adjust length to original
0009E2	4550	BBC4	01BC4		3548		BAL R5,S#UVAR2	Update VAR2
0009E6	47F0	B7E2	017E2		3549		B ERRDONE	Successful Request
					3550	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				3552 *	*/*****	*/
				3553 *18	*/ Function: COUNT REQ REQ	*/
				3554 *18	*/ Function: FINDALL (Alias COUNT) REQ REQ	*/
				3555 *19	*/ Function: FIND REQ REQ	*/
				3556 *05	*/ Function: INDEX (Alias FIND) REQ REQ	*/
				3557 *20	*/ Function: FINDL REQ REQ	*/
				3558 *06	*/ Function: INDEXB (Alias FINDL) REQ REQ	*/
				3559 *	*/*****	*/
				3560 *	*/ COUNT Number of search string occurrences	*/
				3561 *	*/ FIND Position of first search string occurrence	*/
				3562 *	*/ FINDL Position of last search string occurrence	*/
				3563 *	*/	*/
				3564 *	*/ VAR2 contains Search Params -----	*/
				3565 *	*/ Format: "sssssss",bbb,eee	*/
				3566 *	*/ where;	*/
				3567 *	*/ s - search string in double quotes	*/
				3568 *	*/ b - optional, begining search position	*/
				3569 *	*/ 1-3 numeric digits, default=1	*/
				3570 *	*/ e - optional, ending search position	*/
				3571 *	*/ 1-3 numeric digits, default=end	*/
				3572 *	*/ Params separated by commas (,):	*/
				3573 *	*/ **If no VAR2 specified, BLANK is search string	*/
				3574 *	*/ including default start and end positions	*/
				3575 *	*/	*/
				3576 *	*/ - R0, R2, R3, R4, R5, R6 Working Register	*/
				3577 *	*/ - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
				3578 *	*/ - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
				3579 *	*/*****	*/
				3580 *	*/ - RC=0 Search string NOT found	*/
				3581 *	*/ - RC>0 COUNT: number of occurrences	*/
				3582 *	*/ - FUNDF: pos of occurrence, relative to 1	*/
				3583 *	*/ - FUNDL: pos of occurrence, relative to 1	*/
				3584 *	*/ - RC=4016 VAR2 not found	*/
				3585 *	*/ - RC=4032 Start > End Location	*/
				3586 *	*/*****	*/
0009EA	4550 BA3E	01A3E	009EA	3587 PCOUNT	EQU * Process COUNT	
				3588	BAL R5,S#GVAR1 Get VAR1	
				3589 *	*/-----	*/
				3590 *	*/ Initialize Search Lengths	*/
				3591 *	*/-----	*/
0009EE	4120 0000	00000		3592	LA R2,0 Initialize Search Lengths	
0009F2	5020 D1F0	001F0		3593	ST R2,SRCHFND	
0009F6	5020 D1FC	001FC		3594	ST R2,SRCHSTRL	
0009FA	5020 D200	00200		3595	ST R2,SRCHST2L	
0009FE	5020 D1F4	001F4		3596	ST R2,SRCHBGL	
000A02	5020 D1F8	001F8		3597	ST R2,SRCHENDL	
				3598 *	*	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20	
					3600 *	*/-----/		
					3601 *	/* Copy VAR1 to VARWORK	*/	
					3602 *	*/-----/		
000A06	0670				3603	BCTR R7,0	Adjust for EXecute	
000A08	4470	DCBA	00CBA		3604	EX R7,MVCVARWK	Execute VARWORK value move	
					3605	*MVCVARWK MVC VARWORK(0),0(R8)	Save VARWORK value	
000A0C	4177	0001	00001		3606	LA R7,1(R7)	Reset R7 to original value	
000A10	4120	D6E4	006E4		3607	LA R2,VARWORK	Start addr - VARWORK	
					3608 *	*/-----/		
					3609 *	/* Check for absence of VAR2 (set default search)	*/	
					3610 *	*/-----/		
000A14	9540	DB19	00B19		3611	CLI VAR2,C' '	VAR2 name captured?	
000A18	4780	AA2A	00A2A		3612	BE LOADDFLT	No, use search default	
000A1C	D507	DB19	C7C0	00B19	027C0	3613	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
000A22	4780	AA2A	00A2A		3614	BE LOADDFLT	Yes, use search default	
000A26	47F0	AA3A	00A3A		3615	B LOADVAR2	No, get VAR2	
					3616 *	*/-----/		
					3617 *	/* Default search values	*/	
					3618 *	*/-----/		
				00A2A	3619	LOADDFLT EQU *		
000A2A	D202	D5E3	C88B	005E3	0288B	3620	MVC VAL2(3),=C'" "'	Search default, BLANK
000A30	D203	D290	C7DC	00290	027DC	3621	MVC VALL2,=F'3'	Search default length, 3
000A36	47F0	AA48	00A48		3622	B LOADINSS	Start process	
					3623 *	*/-----/		
					3624 *	/* Get VAR2	*/	
					3625 *	*/-----/		
				00A3A	3626	LOADVAR2 EQU *		
000A3A	4550	BB50	01B50		3627	BAL R5,S#GVAR2	Get VAR2	
000A3E	58F0	D04C	0004C		3628	L R15,RC#441	Restore R15	
000A42	12FF				3629	LTR R15,R15	Successful IKJCT441?	
000A44	4770	B8A4	018A4		3630	BNZ ERR4016	No, VAR2 not found	
					3631 *	*/-----/		
					3632 *	/* Load Registers with VAR2	*/	
					3633 *	*/-----/		
				00A48	3634	LOADINSS EQU *	Start loading search string	
000A48	5850	D294	00294		3635	L R5,VALP2	VAR2 address	
000A4C	5860	D290	00290		3636	L R6,VALL2	VAR2 length	
					3637 *	*/-----/		
					3638 *	/* Get search parms	*/	
					3639 *	*/-----/		
000A50	4510	C386	02386		3640	BAL R1,S#SRCHP	Get Search Parms	
					3641 *			

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3643	*	*/-----/	
					3644	*	/* Initialize and check Search Start and End Locations */	
					3645	*	*/-----/	
					3646	*		
000A54	5830	D204	00204		3647	L	R3,SRCHS	Search String
000A58	5840	D208	00208		3648	L	R4,SRCHE	Search Start Location
000A5C	5850	D1FC	001FC		3649	L	R5,SRCHSTRL	Search End Location
000A60	0650				3650	BCTR	R5,0	Search String Length
000A62	4110	0000	00000		3651	LA	R1,0	Adjust for EXecute CLC
000A66	4120	D6E4	006E4		3652	LA	R2,VARWORK	Find COUNTER
				00A6A	3653	PCNTC0	EQU *	Address of VAR1 copy
000A6A	1934				3654	CR	R3,R4	Check Start LOC value
000A6C	4720	B930	01930		3655	BH	ERR4032	Start > End location?
				00A70	3656	PCNTC1	EQU *	YES, start end loc error
000A70	1233				3657	LTR	R3,R3	Check Start LOC value
000A72	4780	AA80	00A80		3658	BZ	PCNTC2	Start LOC specified?
					3659	*		NO, check End LOC
000A76	1937				3660	CR	R3,R7	YES.
000A78	4720	AA80	00A80		3661	BH	PCNTC2	StartLoc > Length?
					3662	*		Yes, Ignore, check End LOC,
000A7C	1A23				3663	AR	R2,R3	NO.
000A7E	0620				3664	BCTR	R2,0	R2 = VAR1 addr + Start LOC
				00A80	3665	PCNTC2	EQU *	R2 = R2 - 1
000A80	1244				3666	LTR	R4,R4	Check End LOC value
000A82	4780	AA9A	00A9A		3667	BZ	PCOUNTL	END LOC specified?
					3668	*		NO, process...
000A86	1947				3669	CR	R4,R7	YES.
000A88	4720	AA9A	00A9A		3670	BH	PCOUNTL	StartLoc > Length?
					3671	*		Yes, Ignore, process...
000A8C	1804				3672	LR	R0,R4	NO.
000A8E	1B03				3673	SR	R0,R3	R0 = End Loc
000A90	5A00	C7F4	027F4		3674	A	R0,=F'1'	R0 = R0 - Start LOC
000A94	1870				3675	LR	R7,R0	R0 = R0 + 1
					3676	*		R7 = VAR1 LEN adjusted

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3678	*	*/-----/	
					3679	*	/* Count of substring	*/
					3680	*	*/-----/	
000A96	5800	D1FC	001FC		3681		L R0,SRCHSTRL	Search String Length
				00A9A	3682	PCOUNTL	EQU *	Process COUNT
000A9A	4450	DCD8	00CD8		3683		EX R5,SRCHCLC	Substring found?
					3684	*SRCHCLC	CLC 0(0,R2),SRCHSTR	
000A9E	4780	AAAE	00AAE		3685		BE PCOUNTF	YES,
000AA2	4122	0001	00001		3686		LA R2,1(R2)	NO, bump to next byte of VAR1
000AA6	4670	AA9A	00A9A		3687		BCT R7,PCOUNTL	Check again...
000AAA	47F0	AAE0	00AE0		3688		B PCOUNTX	Done
				00AAE	3689	PCOUNTF	EQU *	
000AAE	5070	D1F0	001F0		3690		ST R7,SRCHFND	
000AB2	D507	DB09	C670	00B09	02670		CLC FUNCT,=C'FIND	' FIND First function?
000AB8	4780	AAE0	00AE0		3692		BE PCOUNTX	YES. Return Position.
000ABC	D507	DB09	C5F0	00B09	025F0		CLC FUNCT,=C'INDEX	' Alias FIND First function?
000AC2	4780	AAE0	00AE0		3694		BE PCOUNTX	YES. Return Position.
000AC6	1970				3695		CR R7,R0	Enough for another search?
000AC8	4740	AAE0	00AE0		3696		BL PCOUNTX	NO, Done...
					3697	*		YES, count FIND.
000ACC	4111	0001	00001		3698		LA R1,1(R1)	Add 1 to Find COUNT
000AD0	1A25				3699		AR R2,R5	Bump up VAR1 position by SRCHSTRL
000AD2	4122	0001	00001		3700		LA R2,1(R2)	...adjust to next byte of VAR1
000AD6	1B75				3701		SR R7,R5	Adjust loop count by SRCHSTRL
000AD8	4670	AA9A	00A9A		3702		BCT R7,PCOUNTL	Adjust loop count
000ADC	47F0	AAE0	00AE0		3703		B PCOUNTX	Done
				00AE0	3704	PCOUNTX	EQU *	
000AE0	D507	DB09	C658	00B09	02658		CLC FUNCT,=C'COUNT	' COUNT function?
000AE6	4780	AB12	00B12		3706		BE PCOUNTXZ	Yes, return count (R1)
000AEA	D507	DB09	C660	00B09	02660		CLC FUNCT,=C'FINDALL	' Alias COUNT function?
000AF0	4780	AB12	00B12		3708		BE PCOUNTXZ	Yes, return count (R1)
					3709	*	must be FIND FINDL	No, return found position (R1)
					3710	*	INDEX INDEXB	
000AF4	D503	D1F0	C7D8	001F0	027D8		CLC SRCHFND,=F'0'	
000AFA	4780	AB0E	00B0E		3712		BE PCOUNTXY	
000AFE	5810	D280	00280		3713		L R1,VALL1	
000B02	5B10	D1F0	001F0		3714		S R1,SRCHFND	
000B06	5A10	C7F4	027F4		3715		A R1,=F'1'	
000B0A	47F0	AB12	00B12		3716		B PCOUNTXZ	
				00B0E	3717	PCOUNTXY	EQU *	
000B0E	5810	D1F0	001F0		3718		L R1,SRCHFND	
				00B12	3719	PCOUNTXZ	EQU *	
000B12	18F1				3720		LR R15,R1	Return POSITION or COUNT
000B14	47F0	B7EC	017EC		3721		B ERRDONEF	Successful Request - function
					3722	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3724	*	*/*****	*/
					3725	*21	/* Function: CENTER	REQ OPT */
					3726	*	*/*****	*/
					3727	*	/* Center string content	*/
					3728	*	/* - R3, R4, R5, R6	Working Register */
					3729	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3730	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3731	*	*/*****	*/
					3732	*	/* - RC=0	Successful Request */
					3733	*	*/*****	*/
000B18	4550	BA3E	01A3E	00B18	3734	PCENTER	EQU *	Process CENTER
000B1C	4550	BC4A	01C4A		3735		BAL R5,S#GVAR1	Get VAR1
					3736		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
					3737	*	*/-----	*/
					3738	*	/* Determine content start	*/
					3739	*	*/-----	*/
000B20	9540	8000	00000	00B20	3740	PCTRL	EQU *	
000B24	4770	AB34	00B34		3741		CLI 0(R8),C' '	Byte = SPACE?
000B28	4188	0001	00001		3742		BNE CTRBEG	NO, found begin position
000B2C	4670	AB20	00B20		3743		LA R8,1(R8)	Bump up to next byte of VAR1
					3744		BCT R7,PCTRL	LOOP for blanks
					3745	*		all blank, nothing to center
000B30	47F0	AB88	00B88		3746		B PCENTERX	Done...
				00B34	3747	CTRBEG	EQU *	
000B34	5840	D280	00280		3748		L R4,VALL1	VAR1 Length
000B38	1B47				3749		SR R4,R7	
000B3A	5A40	C7F4	027F4		3750		A R4,=F'1'	Starting position
					3751	*	*/-----	*/
					3752	*	/* Determine content end	*/
					3753	*	*/-----	*/
000B3E	5870	D280	00280		3754		L R7,VALL1	VAR1 Length
000B42	5880	D284	00284		3755		L R8,VALP1	VAR1 Address
000B46	1A87				3756		AR R8,R7	Add VAR1 length
000B48	5B80	C7F4	027F4		3757		S R8,=F'1'	VAR1 end address
				00B4C	3758	PCTRE	EQU *	
000B4C	9540	8000	00000		3759		CLI 0(R8),C' '	Byte = SPACE?
000B50	4770	AB5E	00B5E		3760		BNE CTREND	NO, found end position
000B54	0680				3761		BCTR R8,0	Bump down byte position of VAR1
000B56	4670	AB4C	00B4C		3762		BCT R7,PCTRE	LOOP for blanks
					3763	*		all blank, nothing to center
000B5A	47F0	AB88	00B88		3764		B PCENTERX	Done...
					3765	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				00B5E	3767	CTREND	EQU *	
000B5E	1857				3768		LR R5,R7	Ending position
					3769	*	/*	/-----/
					3770	*	/*	Determine content length */
					3771	*	/*	/-----/
000B60	1B54				3772		SR R5,R4	End - Start Pos
000B62	5A50	C7F4	027F4		3773		A R5,=F'1'	adj for content length
					3774	*	/*	/-----/
					3775	*	/*	Determine content starting address */
					3776	*	/*	/-----/
000B66	5870	D280	00280		3777		L R7,VALL1	VAR1 Length
000B6A	5880	D284	00284		3778		L R8,VALP1	VAR1 Address
000B6E	1A84				3779		AR R8,R4	
000B70	0680				3780		BCTR R8,0	content starting address
					3781	*	/*	/-----/
					3782	*	/*	Determine center starting position */
					3783	*	/*	/-----/
000B72	5830	D280	00280		3784		L R3,VALL1	VAR1 Length
000B76	1B35				3785		SR R3,R5	- content length
000B78	8830	0001	00001		3786		SRL R3,1	Divide by 2 for centering
					3787	*	/*	/-----/
					3788	*	/*	Determine center starting address */
					3789	*	/*	/-----/
000B7C	4160	D6E4	006E4		3790		LA R6,VARWORK	Address of VARWORK
000B80	1A63				3791		AR R6,R3	adjust address
					3792	*	/*	/-----/
					3793	*	/*	Move content to center of VAR2 */
					3794	*	/*	/-----/
000B82	0650				3795		BCTR R5,0	Adjust for EXecute
000B84	4450	DCCC	00CCC		3796		EX R5,MVCR8\$6	Execute R8-R6 MVC
					3797	*MVCR8\$6	MVC 0(0,R6),0(R8)	
				00B88	3798	PCENTERX	EQU *	
000B88	4550	BBC4	01BC4		3799		BAL R5,S#UVAR2	Update VAR2
000B8C	47F0	B7E2	017E2		3800		B ERRDONE	Successful Request
					3801	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3803	*	/* ***** */	
					3804	*22	/* Function: LJUST	REQ OPT
					3805	*	/* ***** */	
					3806	*	/* Left justify string content	
					3807	*	/* - R5, R6	Working Register
					3808	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					3809	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					3810	*	/* ***** */	
					3811	*	/* - RC=0	Successful Request
					3812	*	/* ***** */	
000B90	4550	BA3E	01A3E	00B90	3813	PLJUST	EQU *	Process LJUST
000B94	4550	BC4A	01C4A		3814		BAL R5,S#GVAR1	Get VAR1
					3815		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
					3816	*	/* ----- */	
					3817	*	/* Determine content begin	
					3818	*	/* ----- */	
000B98	9540	8000	00000	00B98	3819	PLJSTL	EQU *	
000B9C	4770	ABB4	00BB4		3820		CLI 0(R8),C' '	Byte = SPACE?
000BA0	4188	0001	00001		3821		BNE PLJBEG	NO, found begin position
000BA4	4670	AB98	00B98		3822		LA R8,1(R8)	Bump up byte position of VAR1
000BA8	5870	D280	00280		3823		BCT R7,PLJSTL	LOOP for blanks
					3824		L R7,VALL1	Reset VAR1 length
					3825	*		all blank, nothing to justify
000BAC	4550	BBC4	01BC4		3826		BAL R5,S#UVAR2	Update VAR2
000BB0	47F0	B7E2	017E2		3827		B ERRDONE	Successful Request
				00BB4	3828	PLJBEG	EQU *	
					3829	*		R8=address of content start
					3830	*		R5=start of content position
					3831	*		R7=length of remaining content
000BB4	5850	D280	00280		3832		L R5,VALL1	VAR1 Length
000BB8	1B57				3833		SR R5,R7	- current position
000BBA	5A50	C7F4	027F4		3834		A R5,=F'1'	Adjust
					3835	*	/* ----- */	
					3836	*	/* Determine left justify starting address	
					3837	*	/* ----- */	
000BBE	4160	D6E4	006E4		3838		LA R6,VARWORK	Address of VARWORK
					3839	*	/* ----- */	
					3840	*	/* Right justify content into VAR2	
					3841	*	/* ----- */	
					3842	*	L R8,VALP1	Address of VAR1 content
000BC2	0670				3843		BCTR R7,0	Adjust for EXecute
000BC4	4470	DCCC	00CCC		3844		EX R7,MVCR8\$6	Execute R8-R6 MVC
					3845	*MVCR8\$6	MVC 0(0,R6),0(R8)	
000BC8	5870	D280	00280		3846		L R7,VALL1	Reset VAR1 length
				00BCC	3847	PLJUSTX	EQU *	
000BCC	4550	BBC4	01BC4		3848		BAL R5,S#UVAR2	Update VAR2
000BD0	47F0	B7E2	017E2		3849		B ERRDONE	Successful Request
					3850	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					3852 *	*/*****	*/
					3853 *23	/* Function: RJUST	REQ OPT
					3854 *24	/* Function: ZFILL	REQ OPT
					3855 *	*/*****	*/
					3856 *	/* RJUST Right justify string content	*/
					3857 *	/* ZFILL Zero Fill string content after right justify	*/
					3858 *	/* - R5, R6	Working Register
					3859 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3860 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3861 *	*/*****	*/
					3862 *	/* - RC=0	Successful Request
					3863 *	*/*****	*/
				00BD4	3864 PRJUST	EQU *	Process RJUST
000BD4	4550	BA3E	01A3E		3865	BAL R5,S#GVAR1	Get VAR1
000BD8	4550	BC4A	01C4A		3866	BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
					3867 *	*/-----	*/
					3868 *	/* Determine content end	*/
					3869 *	*/-----	*/
000BDC	1A87				3870	AR R8,R7	Add VAR1 length
000BDE	0680				3871	BCTR R8,0	VAR1 end address
				00BE0	3872 PRJSTL	EQU *	
000BE0	9540	8000	00000		3873	CLI 0(R8),C' '	Byte = SPACE?
000BE4	4770	ABFA	00BFA		3874	BNE PRJEND	NO, found end position
000BE8	0680				3875	BCTR R8,0	Bump down byte position of VAR1
000BEA	4670	ABE0	00BE0		3876	BCT R7,PRJSTL	LOOP for blanks
000BEE	5870	D280	00280		3877	L R7,VALL1	Reset VAR1 length
					3878 *		all blank, nothing to move
000BF2	4550	BBC4	01BC4		3879	BAL R5,S#UVAR2	Update VAR2
000BF6	47F0	B7E2	017E2		3880	B ERRDONE	Successful Request
				00BFA	3881 PRJEND	EQU *	
					3882 *		R7=length of content
					3883 *	*/-----	*/
					3884 *	/* Determine right justify starting address	*/
					3885 *	*/-----	*/
000BFA	4160	D6E4	006E4		3886	LA R6,VARWORK	Address of VARWORK
000BFE	5A60	D280	00280		3887	A R6,VALL1	+ Length of VAR1
000C02	1B67				3888	SR R6,R7	- RJUST content
					3889 *	*/-----	*/
					3890 *	/* Right justify content into VAR2	*/
					3891 *	*/-----	*/
000C04	5880	D284	00284		3892	L R8,VALP1	Address of VAR1 content
000C08	0670				3893	BCTR R7,0	Adjust for EXecute
000C0A	4470	DCCC	00CCC		3894	EX R7,MVCR8\$6	Execute R8-R6 MVC
					3895 *MVCR8\$6	MVC 0(0,R6),0(R8)	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3897	*	*/-----/	
					3898	*	/* Zero Fill Request?	*/
					3899	*	*/-----/	
000C0E	D507	DB09	C690	00B09	02690	3900	CLC	FUNCT,=C'ZFILL ' ZFILL?
000C14	4770	AC34		00C34		3901	BNE	PRJX NO, done.
000C18	5870	D280		00280		3902	L	R7,VALL1 YES. Reset VAR1 length
000C1C	4160	D6E4		006E4		3903	LA	R6,VARWORK
				00C20	3904	PRJZF	EQU	*
000C20	9540	6000		00000		3905	CLI	0(R6),C' ' Byte = SPACE?
000C24	4770	AC34		00C34		3906	BNE	PRJX NO, Done.
000C28	92F0	6000		00000		3907	MVI	0(R6),C'0' YES, Zero Fill
000C2C	4166	0001		00001		3908	LA	R6,1(R6) Bump up
000C30	4670	AC20		00C20		3909	BCT	R7,PRJZF LOOP for blanks
				00C34	3910	PRJX	EQU	*
000C34	5870	D280		00280		3911	L	R7,VALL1 Reset VAR1 length
				00C38	3912	PRJUSTX	EQU	*
000C38	4550	BBC4		01BC4		3913	BAL	R5,S#UVAR2 Update VAR2
000C3C	47F0	B7E2		017E2		3914	B	ERRDONE Successful Request
					3915	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3917	*	/* *****/	
					3918	*25	/* Function: WORDS REQ NO	
					3919	*	/* *****/	
					3920	*	/* Count number of contiguous content separated by	
					3921	*	/* blanks in a string. Count is returned in return	
					3922	*	/* code.	
					3923	*	/* - R5 Working Register	
					3924	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					3925	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					3926	*	/* *****/	
					3927	*	/* - RC=# Number of WORDS	
					3928	*	/* - RC=4022 VAR2 not required	
					3929	*	/* *****/	
			00C40		3930	PWORDS	EQU * Process WORDS	
					3931	*	/* *****/	
					3932	*	/* No VAR2 content in position 1 needed	
					3933	*	/* *****/	
000C40	D507	DB19	C7C0	00B19	027C0	3934	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
000C46	4780	AC52		00C52		3935	BE PWORDS#	YES, continue...
000C4A	9540	DB19		00B19		3936	CLI VAR2,C' '	VAR2 name captured?
000C4E	4770	B8F8		018F8		3937	BNE ERR4022	YES, VAR2 not needed
				00C52	3938	PWORDS#	EQU *	
000C52	4550	BA3E		01A3E		3939	BAL R5,S#GVAR1	Get VAR1
000C56	4150	0000		00000		3940	LA R5,0	Word Count
					3941	*	/* *****/	
					3942	*	/* Count WORDS, non-blank contig blocks	
					3943	*	/* *****/	
				00C5A	3944	PWORDBLK	EQU *	
000C5A	9540	8000		00000		3945	CLI 0(R8),C' '	BLANK Character ?
000C5E	4770	AC6E		00C6E		3946	BNE PWORDFND	NO, found begin of WORD
				00C62	3947	PWORD000	EQU *	
000C62	4188	0001		00001		3948	LA R8,1(R8)	YES, Bump up position of VAR1
000C66	4670	AC5A		00C5A		3949	BCT R7,PWORDBLK	Keep looking until no BLANK
000C6A	47F0	AC8A		00C8A		3950	B PWORDSX	Done... Return
				00C6E	3951	PWORDFND	EQU *	WORD found
000C6E	4155	0001		00001		3952	LA R5,1(R5)	Add 1 to count
000C72	47F0	AC7E		00C7E		3953	B PWORD002	Go find next BLANK
				00C76	3954	PWORDNXT	EQU *	
000C76	9540	8000		00000		3955	CLI 0(R8),C' '	BLANK Character ?
000C7A	4780	AC62		00C62		3956	BE PWORD000	YES, look for next word
				00C7E	3957	PWORD002	EQU *	
000C7E	4188	0001		00001		3958	LA R8,1(R8)	NO, bump up position of VAR1
000C82	4670	AC76		00C76		3959	BCT R7,PWORDNXT	Keep looking until BLANK
000C86	47F0	AC8A		00C8A		3960	B PWORDSX	Done... Return
					3961	*	/* *****/	
					3962	*	/* Return WORD count	
					3963	*	/* *****/	
				00C8A	3964	PWORDSX	EQU *	
000C8A	18F5				3965	LR R15,R5		Return WORD count
000C8C	47F0	B7EC		017EC	3966	B ERRDONEF		Successful Request - function
					3967	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					3969	*	*/*****	*/
					3970	*26	/* Function: GEN#	REQ NO
					3971	*65	/* Function: TDSN	REQ NO
					3972	*	*/*****	*/
					3973	*	/* GEN# Generate a number (000-999)	*/
					3974	*	/* TDSN Generate temp DSN nodes	*/
					3975	*	/* - R0, R1, R5	Working Register
					3976	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
					3977	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
					3978	*	*/*****	*/
					3979	*	/* - RC=0	Successful Request
					3980	*	/* - RC=4022	VAR2 not required
					3981	*	*/*****	*/
			00C90		3982	PGEN#	EQU *	Process GEN#
			00C90		3983	PTDSN	EQU *	Process TSDN
			00C90		3984	PNOW	EQU *	Process PNOW
					3985	*	/*-----	*/
					3986	*	/* No VAR2 content in position 1 needed	*/
					3987	*	/*-----	*/
000C90	D507	DB19	C7C0	00B19	027C0	3988	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
000C96	4780	ACA2		00CA2		3989	BE PGEN##	YES, continue...
000C9A	9540	DB19		00B19		3990	CLI VAR2,C' '	VAR2 name captured?
000C9E	4770	B8F8		018F8		3991	BNE ERR4022	YES, VAR2 not needed
				00CA2	3992	PGEN##	EQU *	
000CA2	4550	BA3A		01A3A	3993	BAL R5,S#GVAR1N		Get VAR1, do not check for NULL
					3994	TIME DEC		Time in R0 HH_MM_SS_TH
					3999	*		Date in R1 00_YY_DD_DC
000CAC	5000	D1D8		001D8	4000	ST R0,WORKAREA		Store R0 to fullword
000CB0	F143	D218	D1D8	00218	001D8	4001	MVO DW(5),WORKAREA	Pack via MVO into 5 bytes
000CB6	960F	D21C		0021C		4002	OI DW+4,X'0F'	... with an F zone
000CBA	F384	D1DC	D218	001DC	00218	4003	UNPK UNPKAREA(9),DW(5)	Unpack 9 bytes 0HHMSSTH
					4004	*		
000CC0	D507	DB09	C7A8	00B09	027A8	4005	CLC FUNCT,=C'TDSN '	TDSN function ?
000CC6	4780	ACE8		00CE8		4006	BE PTDSNC	Yes, go to TDSNC
000CCA	D507	DB09	C7B0	00B09	027B0	4007	CLC FUNCT,=C'NOW '	NOW function ?
000CD0	4780	AD4A		00D4A		4008	BE PNOWC	Yes, go to PNOWC
					4009	*		No, must be GEN#
000CD4	D202	D6E4	D1E2	006E4	001E2	4010	MVC VARWORK(3),UNPKAREA+6	Use STH portion for gen number
000CDA	D203	D280	C7DC	00280	027DC	4011	MVC VALL1,=F'3'	VAR LENGTH = 3
000CE0	4170	0003		00003		4012	LA R7,3	VAR1 value len = 3
000CE4	47F0	ADF6		00DF6		4013	B PGEN#X	Done...
					4014	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4016	*	*/-----/	
					4017	*	/* VARWORK --> VAR1 + Dyyjjj.Thhmmss */	
					4018	*	*/-----/	
			00CE8		4019	PTDSNC	EQU *	
000CE8	0670				4020		BCTR R7,0	Adjust for EXecute
000CEA	4470	DCBA	00CBA		4021		EX R7,MVCVARWK	Execute VARWORK value move
					4022	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
000CEE	4177	0001	00001		4023		LA R7,1(R7)	Reset R7 to original value
000CF2	4150	D6E4	006E4		4024		LA R5,VARWORK	Start addr - VARWORK
000CF6	1A57				4025		AR R5,R7	Point to end of VAR1 in VARWORK
000CF8	1277				4026		LTR R7,R7	VAR1 data?
000CFA	4780	AD18	00D18		4027		BZ PTDSN1	No, go move time date to DSN
000CFE	0650				4028		BCTR R5,0	Yes, back up and check for '.'
000D00	954B	5000	00000		4029		CLI 0(R5),C'.'	Node at end of VAR1?
000D04	4780	AD14	00D14		4030		BE PTDSN0	Yes, go move time date to DSN
000D08	4155	0001	00001		4031		LA R5,1(R5)	No, add node to DSN
000D0C	4177	0001	00001		4032		LA R7,1(R7)	and adjust VAR1L
000D10	924B	5000	00000		4033		MVI 0(R5),C'.'	
			00D14		4034	PTDSN0	EQU *	
000D14	4155	0001	00001		4035		LA R5,1(R5)	Point to end of VAR1 in VARWORK
			00D18		4036	PTDSN1	EQU *	
000D18	92E3	5007	00007		4037		MVI 7(R5),C'T'	'Thhmmss'
000D1C	D205	5008	D1DD 00008	001DD	4038		MVC 8(6,R5),UNPKAREA+1	
000D22	5010	D1D8	001D8		4039		ST R1,WORKAREA	Date in R1 00_YY_DD_DC
000D26	F342	D1DC	D1D9 001DC	001D9	4040		UNPK UNPKAREA(5),WORKAREA+1(3)	Unpack YY_DD_DC
000D2C	96F0	D1E0	001E0		4041		OI UNPKAREA+4,X'F0'	F-zone
000D30	92C4	5000	00000		4042		MVI 0(R5),C'D'	'Dyyjjj.'
000D34	D204	5001	D1DC 00001	001DC	4043		MVC 1(5,R5),UNPKAREA	
000D3A	924B	5006	00006		4044		MVI 6(R5),C'.'	
000D3E	4177	000E	0000E		4045		LA R7,14(R7)	Adjust to new length
000D42	5070	D280	00280		4046		ST R7,VALL1	
000D46	47F0	ADF6	00DF6		4047		B PTDSNX	Done...
					4048	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4050	* * /-----/	
					4051	* * /* 0....5....0....5....0....5....0....5...*/	
					4052	* * /* VARWORK --> MM/DD/CCYY.JJJ HH:MM:SS.TT 6 DAYOFWEEK*/	
					4053	* * /-----/	
			00D4A		4054	PNOWC EQU *	
000D4A	D201	D6F3	D1DD	006F3	001DD	4055 MVC VARWORK+15(02), UNPKAREA+1	'hh:mm:ss.tt'
000D50	927A	D6F5		006F5		4056 MVI VARWORK+17,C'.'	
000D54	D201	D6F6	D1DF	006F6	001DF	4057 MVC VARWORK+18(02), UNPKAREA+3	
000D5A	927A	D6F8		006F8		4058 MVI VARWORK+20,C'.'	
000D5E	D201	D6F9	D1E1	006F9	001E1	4059 MVC VARWORK+21(02), UNPKAREA+5	
000D64	924B	D6FB		006FB		4060 MVI VARWORK+23,C'.'	
000D68	D201	D6FC	D1E3	006FC	001E3	4061 MVC VARWORK+24(02), UNPKAREA+7	
000D6E	5010	D1D8		001D8		4062 ST R1,WORKAREA Date in R1 II_YY_DD_DC	
000D72	F363	D1DC	D1D8	001DC	001D8	4063 UNPK UNPKAREA(7),WORKAREA+0(4)	Unpack IIYYDDD
000D78	96F0	D1E2		001E2		4064 OI UNPKAREA+6,X'F0'	F-zone IIYYDDD
000D7C	D206	D6E4	D1DC	006E4	001DC	4065 MVC VARWORK+00(07), UNPKAREA	IIYYDDD
000D82	F211	D220	D1DC	00220	001DC	4066 PACK FW(2), UNPKAREA(2)	II
000D88	FA11	D220	C862	00220	02862	4067 AP FW(2),=P'19'	CC = II + 19
000D8E	F311	D1DC	D220	001DC	00220	4068 UNPK UNPKAREA(2),FW(2)	CC
000D94	96F0	D1DD		001DD		4069 OI UNPKAREA+1,X'F0'	F-zone CC
000D98	924B	D6EE		006EE		4070 MVI VARWORK+10,C'.'	
000D9C	D202	D6EF	D1E0	006EF	001E0	4071 MVC VARWORK+11(03), UNPKAREA+4	'JJJ'
000DA2	9261	D6E9		006E9		4072 MVI VARWORK+05,C'/'	
000DA6	D203	D6EA	D1DC	006EA	001DC	4073 MVC VARWORK+06(04), UNPKAREA	'CCYY'
000DAC	92F0	D23C		0023C		4074 MVI DCCCCYY,C'0'	0CCYY
000DB0	D203	D23D	D1DC	0023D	001DC	4075 MVC DCCCCYY+1(4), UNPKAREA	
000DB6	D202	D247	D1E0	00247	001E0	4076 MVC DCJJJ, UNPKAREA+4	DCJJJ
					4077	* * /-----/	
					4078	* * /* Determine if CCYY is a leap year */	
					4079	* * /-----/	
					4080	* input DCCCCYY 0CCYY	
					4081	* output DC#LEAP 0=common year 1=leap year	
					4082	* DC#YDAYS 365=common year 366=leap year	
000DBC	4550	BE8E		01E8E		4083 BAL R5,S#ISLPY Determine Leap Year	
					4084	* * /-----/	
					4085	* * /* Determine MM and DD from JJJ */	
					4086	* * /-----/	
					4087	* input DC#LEAP PKJJJ	
					4088	* Work FW, R3, R4,	
					4089	* output DC#TRM DCMMM DCDDD	
					4090	* R4=addr of MONTHNAME R6=Length of MONTHNAME	
000DC0	F222	D254	D247	00254	00247	4091 PACK PKJJJ,DCJJJ	
000DC6	4550	C148		02148		4092 BAL R5,S#J2MD Determine MM DD from JJJ	
000DCA	D201	D6E4	D245	006E4	00245	4093 MVC VARWORK+00(02), DCMMM+1	'MM'
000DD0	9261	D6E6		006E6		4094 MVI VARWORK+02,C'/'	
000DD4	D201	D6E7	D242	006E7	00242	4095 MVC VARWORK+03(02), DCDDD+1	'DD'
					4096	* * /-----/	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4098	*	*/-----/	
					4099	*	/* Call S#CDOW to calculate DOW	*/
					4100	*	*/-----/	
					4101	*	input DCDDD DCCCCCY DCM MM DC#TRM	
					4102	*	output DCCDOW DCCWDAY	
					4103	*	Working fields DC#YR4 DC#DD DC#YR100 DC#YR400 DC#DOW	
000DDA	4550	BF50		01F50	4104		BAL R5,S#CDOW	
000DDE	D200	D6FF	D24A	006FF	0024A	4105	MVC VARWORK+27(01),DCCDOW	'3'
000DE4	D208	D701	D24B	00701	0024B	4106	MVC VARWORK+29(09),DCCWDAY	'Wednesday'
000DEA	4170	0026		00026		4107	LA R7,38	VAR1 value len = 38
000DEE	5070	D280		00280		4108	ST R7,VALL1	
000DF2	47F0	ADF6		00DF6		4109	B PNOWX	Done...
					4110	*		
				00DF6	4111	PGEN#X	EQU *	
				00DF6	4112	PTDSNX	EQU *	
				00DF6	4113	PNOWX	EQU *	
000DF6	9240	DB19		00B19	4114		MVI VAR2,C' '	Force use of VAR1
000DFA	4550	BBC4		01BC4	4115		BAL R5,S#UVAR2	Update VAR2
000DFE	47F0	B7E2		017E2	4116		B ERRDONE	Successful Request
					4117	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4119	*	/* ***** */	
					4120	*27	/* Function: DD2DSN REQ REQ */	
					4121	*	/* ***** */	
					4122	*	/* Obtain 1st DSN from allocated DD */	
					4123	*	/* - R3, R4, R5, R6 Working Register */	
					4124	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1) */	
					4125	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1) */	
					4126	*	/* ***** */	
					4127	*	/* - RC=0 Successful Request */	
					4128	*	/* - RC=8 DD not found */	
					4129	*	/* - RC=4015 VAR2 is required */	
					4130	*	/* ***** */	
000E02	4550	BA3E	01A3E	00E02	4131	PDD2DSN	EQU *	Process DD2DSN
					4132		BAL R5,S#GVAR1	Get VAR1
				00E06	4133	PDDFIND	EQU *	
000E06	D503	D280	C808	00280	02808	4134	CLC VALL1,=F'8'	DDNAME > 8 charaters?
000E0C	4720	AE14		00E14	4135		BH PDDF8	Yes, override to 8 charaters
000E10	47F0	AE1A		00E1A	4136		B PDDCHKV2	No, check for var2
				00E14	4137	PDDF8	EQU *	
000E14	D203	D280	C808	00280	02808	4138	MVC VALL1,=F'8'	
				00E1A	4139	PDDCHKV2	EQU *	
000E1A	9540	DB19		00B19	4140		CLI VAR2,C' '	Do I have a VAR2?
000E1E	4780	B896		01896	4141		BE ERR4015	No, VAR2 required error
				00E22	4142	FINDDDN	EQU *	YES, find DDN in TIOT
000E22	5840	0010		00010	4143		L R4,CVTPTR	xAddress of CVT
				00000	4144		USING CVT,R4	xTell Assembler, CVT
000E26	5830	4000		00000	4145		L R3,CVTTCPB	7Addr (2 FW) pointing to TCB
					4146		DROP R4	x
000E2A	D207	D218	3000	00218	00000	4147	MVC DW,0(R3)	7Save to DW
000E30	5830	D21C		0021C	4148		L R3,DW+4	7Take 2nd FW as TCB Addr
				00020	4149		USING TCB,R3	7Tell Assembler, TCB
000E34	5840	300C		0002C	4150		L R4,TCBTIO	xLoad TIOT Address
					4151		DROP R3	7
				00000	4152		USING TIOT,R4	xTell Assembler, TIOT
000E38	4130	4018		00018	4153		LA R3,TIOENTRY	7Load TIOENTRY address
				00018	4154		USING TIOENTRY,R3	7Tell Assembler, TIOENTRY
000E3C	5860	D280		00280	4155		L R6,VALL1	
000E40	0660				4156		BCTR R6,0	Adjust for EXecute CLC
				00E42	4157	LOOPDDN	EQU *	YES, find DDN in TIOT
000E42	9500	3000		00018	4158		CLI TIOELNGH,0	End of TIOT?
000E46	4780	AEA8		00EA8	4159		BE DSNNOFND	Yes, DDN not found, DSN=blanks
					4160	*		No, keep looking for DDN...
000E4A	4460	AE5E		00E5E	4161		EX R6,CHKDDN	DDNAME = TIOT entry?
					4162	*CHKDDN	CLC 0(0,R8),TIOEDDNM	DDNAME = TIOT entry?
000E4E	4780	AE64		00E64	4163		BE GOODDDN	Yes, found it!
000E52	1B55				4164		SR R5,R5	Clear R5
000E54	4350	3000		00018	4165		IC R5,TIOELNGH	TIOT Entry Length
000E58	1A35				4166		AR R3,R5	Bump to next TIOT entry
000E5A	47F0	AE42		00E42	4167		B LOOPDDN	
					4168	*HKDDN	CLC 0(0,R8),TIOEDDNM	DDNAME = TIOT entry?
000E5E	D500	8000	3004	00000	00004	4169	CHKDDN CLC 0(0,R8),4(R3)	DDNAME = TIOT entry?

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				00E64	4171	GOODDDN	EQU *	Found DDN
000E64	D703	D220	D220	00220	00220	4172	XC FW,FW	Clear FW (fullword)
000E6A	D202	D221	300C	00221	00024	4173	MVC FW+1(3),TIOEJFCB	FW contains addr of JFCB for DDN
000E70	5850	D220		00220		4174	L R5,FW	R8 = JFCB addrs
000E74	5A50	C80C		0280C		4175	A R5,=F'16'	R8 = JFCB+16 (for offset)
				00000	4176		USING JFCB,R5	Tell Assembler, JFCB
000E78	D22B	D6E4	5000	006E4	00000	4177	MVC VARWORK(44),JFCBDSNM	Move associated dataset name
					4178		DROP R4	
					4179		DROP R3	
					4180		DROP R5	
				00E7E	4181	CALCINIT	EQU *	Calculate DDSN Init
000E7E	4140	0000		00000		4182	LA R4,0	DSN Length
000E82	4130	002C		0002C		4183	LA R3,44	Max Length, loop control
000E86	4150	D6E4		006E4		4184	LA R5,VARWORK	DSN in VARWORK
				00E8A	4185	CALCDSNL	EQU *	Calculate DDSN length
000E8A	9540	5000		00000		4186	CLI 0(R5),C' '	Is it a BLANK?
000E8E	4780	AE9E		00E9E		4187	BE CALCEXIT	Yes, we have a length
000E92	4144	0001		00001		4188	LA R4,1(R4)	No, look at next byte
000E96	4155	0001		00001		4189	LA R5,1(R5)	
000E9A	4630	AE8A		00E8A		4190	BCT R3,CALCDSNL	Look again...
				00E9E	4191	CALCEXIT	EQU *	
000E9E	1874					4192	LR R7,R4	Store new length
000EA0	4550	BBC4		01BC4		4193	BAL R5,S#UVAR2	Update VAR2
000EA4	47F0	AEBA		00EBA		4194	B PDD2DSNX	Return
				00EA8	4195	DSNNOFND	EQU *	
000EA8	D20B	D6E4	C810	006E4	02810	4196	MVC VARWORK(12),=C'DD.NOT.FOUND'	
000EAE	5870	C81C		0281C		4197	L R7,=F'12'	VAR2 Length
000EB2	4550	BBC4		01BC4		4198	BAL R5,S#UVAR2	Update VAR2
000EB6	41F0	0008		00008		4199	LA R15,8	SET RC to 8, DD not found
				00EBA	4200	PDD2DSNX	EQU *	
000EBA	47F0	B7E2		017E2		4201	B ERRDONE	Successful Request

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4203	*	/* *****/	
					4204	*28	/* Function: JOBINFO REQ NO	
					4205	*	/* *****/	
					4206	*	/* JOB Information from TIOT in string	
					4207	*	/* - JOB NAME 8-bytes	
					4208	*	/* - PROC NAME 8-bytes	
					4209	*	/* - STEP NAME 8-bytes	
					4210	*	/* - R3, R4, R5 Working Register	
					4211	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					4212	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					4213	*	/* *****/	
					4214	*	/* - RC=0 Successful Request	
					4215	*	/* - RC=8 DD not found	
					4216	*	/* - RC=4022 VAR2 not required	
					4217	*	/* *****/	
			00EBE		4218	PJOBINFO	EQU * Process JOBINFO	
					4219	*	/* -----/	
					4220	*	/* No VAR2 content in position 1 needed	
					4221	*	/* -----/	
000EBE	D507	DB19	C7C0	00B19	027C0	4222	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
000EC4	4780	AED0		00ED0		4223	BE PJOBINF#	YES, continue...
000EC8	9540	DB19		00B19		4224	CLI VAR2,C' '	VAR2 name captured?
000ECC	4770	B8F8		018F8		4225	BNE ERR4022	YES, VAR2 not needed
				00ED0	4226	PJOBINF#	EQU *	
000ED0	4550	BA3A		01A3A		4227	BAL R5,S#GVAR1N	Get VAR1, do not check for NULL
000ED4	5840	0010		00010		4228	L R4,CVTPTR	Address of CVT
				00000	4229	USING	CVT,R4	Tell Assembler, CVT
000ED8	5830	4000		00000		4230	L R3,CVTTCBP	Addr (2 FW) pointing to TCB
					4231		DROP R4	
000EDC	D207	D218	3000	00218	00000	4232	MVC DW,0(R3)	Save to DW
000EE2	5830	D21C		0021C		4233	L R3,DW+4	Take 2nd FW as TCB Addr
				00020	4234	USING	TCB,R3	Tell Assembler, TCB
000EE6	5840	300C		0002C		4235	L R4,TCBTIO	Load TIOT Address
				00000	4236		DROP R3	
					4237	USING	TIOT,R4	Tell Assembler, TIOT
000EEA	4130	4018		00018		4238	LA R3,TIOENTRY	Load TIOENTRY address
				00018	4239	USING	TIOENTRY,R3	Tell Assembler, TIOENTRY
					4240	*		
000EEE	D217	D6E4	4000	006E4	00000	4241	MVC VARWORK(24),TIOCNJOB	JOB NUMBER and other info
000EF4	D203	D280	C820	00280	02820	4242	MVC VALL1,=F'24'	
000EFA	5870	C820		02820		4243	L R7,=F'24'	
000EFE	9240	DB19		00B19		4244	MVI VAR2,C' '	
				00F02	4245	PJOBINF#	EQU *	
000F02	4550	BBC4		01BC4		4246	BAL R5,S#UVAR2	Update VAR2
000F06	47F0	B7E2		017E2		4247	B ERRDONE	Successful Request
					4248	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4250	*	/* *****/	
					4251	*	/* Function: SETVAR	
					4252	*	/* *****/	
					4253	*	/* SET VAR1 via PARM data	
					4254	*	/* - R5 Working register	
					4255	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					4256	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					4257	*	/* *****/	
000F0A	4550	BA3E	01A3E	00F0A	4258	PSETVAR	EQU * Process SETVAR	
					4259	BAL	R5,S#GVAR1 Get VAR1	
					4260	*		
000F0E	D2FF	D6E4	DA09	006E4	00A09	4261	MVC VARWORK,VARSET1	SETVAR into VARWORK
					4262	*	MVC VAL1,VARSET1	SETVAR into VAR1
000F14	D203	D280	DB3C	00280	00B3C	4263	MVC VALL1,VARSET1L	SETVAR length
					4264	*	L R7,VARSET1L	
000F1A	5870	DB3C		00B3C	4265		L R7,VARSET1L	
000F1E	1277				4266		LTR R7,R7	
000F20	4780	B85E		0185E	4267	BZ	ERR4010	No SETVAR content
000F24	9240	DB19		00B19	4268	MVI	VAR2,C' '	
000F28	4550	BBC4		01BC4	4269	BAL	R5,S#UVAR2	Update VAR2
000F2C	47F0	B7E2		017E2	4270	B	ERRDONE	Successful Request
					4271	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4273	*	/* *****/	
					4274	*29	/* Function: DAYSMM	REQ NO
					4275	*	/* *****/	
					4276	*	/* Number of DAYS in specified MMCCYY	
					4277	*	/* - R3, R4, R5	Working Register
					4278	*	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					4279	*	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					4280	*	/* *****/	
					4281	*	/* - RC=0	Successful Request
					4282	*	/* - RC=4022	VAR2 not required
					4283	*	/* - RC=4040	Invalid length (VAR1)
					4284	*	/* - RC=4034	Not numeric (VAR1)
					4285	*	/* - RC=4049	Invalid MM (VAR1)
					4286	*	/* *****/	
			00F30		4287	PDAYSMM	EQU *	Process DAYSMM
					4288	*	/* -----*/	
					4289	*	/* No	VAR2 content in position 1 needed
					4290	*	/* -----*/	
000F30	D507	DB19	C7C0	00B19	027C0	4291	CLC	VAR2,=C'\$NOVAR2 ' VAR2 none keyword?
000F36	4780	AF42		00F42	4292	BE	PDAYSMM#	YES, continue...
000F3A	9540	DB19		00B19	4293	CLI	VAR2,C' '	VAR2 name captured?
000F3E	4770	B8F8		018F8	4294	BNE	ERR4022	YES, VAR2 not needed
				00F42	4295	PDAYSMM#	EQU *	
000F42	4550	BA3E		01A3E	4296	BAL	R5,S#GVAR1	Get VAR1
000F46	5070	D05C		0005C	4297	ST	R7,R7HOLD	Save VAR1 value Length
000F4A	5080	D060		00060	4298	ST	R8,R8HOLD	Save VAR1 value Address
000F4E	4150	0000		00000	4299	LA	R5,0	
				00F52	4300	PDMML	EQU *	
000F52	9540	8000		00000	4301	CLI	0(R8),C' '	Blank delimiter?
000F56	4780	AF66		00F66	4302	BE	PDMMLX	Yes, go process
000F5A	4188	0001		00001	4303	LA	R8,1(R8)	Bump up VAR1
000F5E	4155	0001		00001	4304	LA	R5,1(R5)	Bump up VAR1 length
000F62	4670	AF52		00F52	4305	BCT	R7,PDMML	...do it again
				00F66	4306	PDMMLX	EQU *	
000F66	1875				4307	LR	R7,R5	Implied VAR1 length
					4308	*	/* -----*/	
					4309	*	/* Check VAR1 for proper length	
					4310	*	/* -----*/	
000F68	5970	C824		02824	4311	C	R7,=F'6'	VAR1 length = 6 ?
000F6C	4770	B9A0		019A0	4312	BNE	ERR4040	No, Error- VAR1 wrong length
000F70	5880	D060		00060	4313	L	R8,R8HOLD	Restore VAR1 value Address
					4314	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4316	*	*/-----/	
					4317	*	/* Check VAR1 for numerics based on LENGTH (R7)	*/
					4318	*	*/-----/	
000F74	4550	BE60	01E60		4319		BAL R5,S#ISNUM	Numeric Test
000F78	59F0	C800	02800		4320		C R15,=F'4000'	Numeric?
000F7C	4780	B94C	0194C		4321		BE ERR4034	NO, error
					4322	*		YES, Continue...
000F80	5870	D05C	0005C		4323		L R7,R7HOLD	Restore VAR1 value Length
000F84	5880	D060	00060		4324		L R8,R8HOLD	Restore VAR1 value Address
					4325	*	*/-----/	
					4326	*	/* Store VAR1 date (R8=MMCCYY)	*/
					4327	*	*/-----/	
000F88	92F0	D244	00244		4328		MVI DCMMM,C'0'	Save MM from VAR1
000F8C	D201	D245	8000	00245	00000	4329	MVC DCMMM+1(2),0(R8)	... to 0MM field
000F92	92F0	D23C	0023C		4330		MVI DCCCCYY,C'0'	Save CCYY from VAR1
000F96	D203	D23D	8002	0023D	00002	4331	MVC DCCCCYY+1(4),2(R8)	... to 0CCYY field
					4332	*	*/-----/	
					4333	*	/* Find Days in MM	*/
					4334	*	*/-----/	
000F9C	4130	000C	0000C		4335		LA R3,MMTBLN	R3=12 Months
000FA0	4140	CD1B	02D1B		4336		LA R4,MMTABLE	R4=Address of MonthTable
				00000	4337		USING MPT,R4	Tell Assembler
				00FA4	4338	MMDAYS	EQU *	
000FA4	D501	D245	4000	00245	00000	4339	CLC DCMMM+1(2),MPTMM	VAR1 MM = MPT MM ?
000FAA	4780	AFBA	00FBA		4340		BE YMMDAYS	Yes, R4=MMTABLE entry
000FAE	4144	0011	00011		4341		LA R4,MPTLE(R4)	No, Move to next MMTABLE entry
000FB2	4630	AFA4	00FA4		4342		BCT R3,MMDAYS	Check again...
000FB6	47F0	B9BC	019BC		4343		B ERR4049	Error, Invalid MM
				00FBA	4344	YMMDAYS	EQU *	
000FBA	F271	D218	4002	00218	00002	4345	PACK DW,MPTMMAXD	Pack number of days for month
000FC0	D501	D245	C864	00245	02864	4346	CLC DCMMM+1(2),=C'02'	MM = FEB ?
000FC6	4770	AFD4	00FD4		4347		BNE PDAYSMMX	NO, go to PDAYSMMX
					4348	*		YES, check for leap year
					4349	*	*/-----/	
					4350	*	/* Determine if CCYY is a leap year	*/
					4351	*	*/-----/	
					4352	*	input DCCCCYY 0CCYY	
					4353	*	output DC#LEAP 0=common year 1=leap year	
					4354	*	DC#YDAYS 365=common year 366=leap year	
000FCA	4550	BE8E	01E8E		4355		BAL R5,S#ISLPY	Determine Leap Year
000FCE	FA70	D218	D239	00218	00239	4356	AP DW,DC#LEAP	Adjust for leap year
				00FD4	4357	PDAYSMMX	EQU *	
000FD4	4FF0	D218	00218		4358		CVB R15,DW	Convert to binary (R15)
000FD8	47F0	B7EC	017EC		4359		B ERRDONEF	Successful Request - function
					4360	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				4362 *	/* *****	*/
				4363 *30	/* Function: DAYSYY	REQ NO
				4364 *31	/* Function: ISLEAP	REQ NO
				4365 *	/* *****	*/
				4366 *	/* DAYSYY Number of days for year in CCYY	*/
				4367 *	/* ISLEAP Determine if CCYY is a leap year	*/
				4368 *	/* - R3, R4, R5	Working Register
				4369 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
				4370 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
				4371 *	/* *****	*/
				4372 *	/* - RC=0	Successful Request
				4373 *	/* - RC=4000	FALSE
				4374 *	/* - RC=4001	TRUE
				4375 *	/* - RC=4022	VAR2 not required
				4376 *	/* - RC=4040	Invalid length (VAR1)
				4377 *	/* - RC=4034	Not numeric (VAR1)
				4378 *	/* *****	*/
		00FDC		4379 PDAYSYY	EQU *	Process DAYSYY
				4380 *	/* -----	*/
				4381 *	/* No	VAR2 content in position 1 needed
				4382 *	/* -----	*/
000FDC	D507 DB19 C7C0	00B19	027C0	4383	CLC	VAR2,=C'\$NOVAR2 ' VAR2 none keyword?
000FE2	4780 AFEE	00FEE		4384	BE	PDAYSYY# YES, continue...
000FE6	9540 DB19	00B19		4385	CLI	VAR2,C' ' VAR2 name captured?
000FEA	4770 B8F8	018F8		4386	BNE	ERR4022 YES, VAR2 not needed
		00FEE		4387 PDAYSYY#	EQU *	
000FEE	4550 BA3E	01A3E		4388	BAL	R5,S#GVAR1 Get VAR1
000FF2	5070 D05C	0005C		4389	ST	R7,R7HOLD Save VAR1 value Length
000FF6	5080 D060	00060		4390	ST	R8,R8HOLD Save VAR1 value Address
000FFA	4150 0000	00000		4391	LA	R5,0
		00FFE		4392 PDYYL	EQU *	
000FFE	9540 8000	00000		4393	CLI	0(R8),C' ' Blank delimiter?
001002	4780 B012	01012		4394	BE	PDYYLX Yes, go process
001006	4188 0001	00001		4395	LA	R8,1(R8) Bump up VAR1
00100A	4155 0001	00001		4396	LA	R5,1(R5) Bump up VAR1 length
00100E	4670 AFFE	00FFE		4397	BCT	R7,PDYYL ...do it again
		01012		4398 PDYYLX	EQU *	
001012	1875			4399	LR	R7,R5 Implied VAR1 length
				4400 *	/* -----	*/
				4401 *	/* Check VAR1 for proper length	*/
				4402 *	/* -----	*/
001014	5970 C828	02828		4403	C	R7,=F'4' VAR1 length = 2 ?
001018	4770 B9A0	019A0		4404	BNE	ERR4040 No, Error- VAR1 wrong length
00101C	5880 D060	00060		4405	L	R8,R8HOLD Restore VAR1 value Address
				4406 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4408 *	*/-----/	
					4409 *	/* Check VAR1 for numerics based on LENGTH (R7)	*/
					4410 *	*/-----/	
001020	4550	BE60	01E60		4411	BAL R5,S#ISNUM	Numeric Test
001024	59F0	C800	02800		4412	C R15,=F'4000'	Numeric?
001028	4780	B94C	0194C		4413	BE ERR4034	NO, Error
					4414 *		YES, Continue...
00102C	5870	D05C	0005C		4415	L R7,R7HOLD	Restore VAR1 value Length
001030	5880	D060	00060		4416	L R8,R8HOLD	Restore VAR1 value Address
					4417 *	*/-----/	
					4418 *	/* Store VAR1 date (R8=CCYY)	*/
					4419 *	*/-----/	
001034	92F0	D23C	0023C		4420	MVI DCCCCYY,C'0'	Save CCYY from VAR1
001038	D203	D23D	8000	0023D 00000	4421	MVC DCCCCYY+1(4),0(R8)	... to 0CCYY field
					4422 *	*/-----/	
					4423 *	/* Determine if CCYY is a leap year	*/
					4424 *	*/-----/	
					4425 *	input DCCCCYY 0CCYY	
					4426 *	output DC#LEAP 0=not leap year 1=leap year	
					4427 *	DC#YDAYS 365=common year 366=leap year	
00103E	4550	BE8E	01E8E		4428	BAL R5,S#ISLPY	Determine Leap Year
001042	D507	DB09	C6C8	00B09 026C8	4429	CLC FUNCT,=C'ISLEAP '	ISLEAP?
001048	4780	B05A	0105A		4430	BE LEAPTF	Yes, return TRUE/FALSE
00104C	F871	D218	D23A	00218 0023A	4431	ZAP DW,DC#YDAYS	No, return number of days in year
001052	4FF0	D218	00218		4432	CVB R15,DW	Convert to binary (R15)
001056	47F0	B7EC	017EC		4433	B ERRDONEF	Successful Request - function
			0105A		4434	LEAPTF EQU *	
00105A	F900	D239	C88E	00239 0288E	4435	CP DC#LEAP,=P'1'	Leap Year?
001060	4780	B80A	0180A		4436	BE ERR4001	Yes, TRUE
001064	47F0	B7FC	017FC		4437	B ERR4000	NO, FALSE
					4438 *		

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
4440 *      * /*****  
4441 *32    * /* Function: CYJ-D8          REQ OPT      */  
4442 *33    * /* Function: CYJ-DAY        REQ OPT      */  
4443 *34    * /* Function: CYJ-DOW        REQ OPT      */  
4444 *35    * /* Function: CYJ-mdcy       REQ OPT      */  
4445 *36    * /* Function: JCY-D8          REQ OPT      */  
4446 *37    * /* Function: JCY-DAY        REQ OPT      */  
4447 *38    * /* Function: JCY-DOW        REQ OPT      */  
4448 *39    * /* Function: JCY-mdcy       REQ OPT      */  
4449 *      * /*  
4450 *      * /*          Date Conversion          */  
4451 *      * /*****  
4452 *      * /*      -D8      converts to monthname DD, CCYY      */  
4453 *      * /*      -DAY     converts to day name                */  
4454 *      * /*      -DOW     converts to day of week number          */  
4455 *      * /* CYJ-*      converts to MMDDCCYY (any order)          */  
4456 *      * /* JCY-*      converts to MMDDCCYY (any order)          */  
4457 *      * /*  
4458 *      * /*      Date input form specified by function bytes 1-3:  */  
4459 *      * /*      CYJ  --> CCYYJJJ      */  
4460 *      * /*      JCY  --> JJJCCYY      */  
4461 *      * /*****  
4462 *      * /* - R2, R3, R4, R5, R6      Working Register      */  
4463 *      * /* - R7 Length of VAR1 value (after BAL S#GVAR1)      */  
4464 *      * /* - R8 Address of VAR1 value (after BAL S#GVAR1)      */  
4465 *      * /*****  
4466 *      * /* - RC=0      Successful Request      */  
4467 *      * /* - RC=4011   Invalid FUNCTION      */  
4468 *      * /* - RC=4040   Invalid length (VAR1)      */  
4469 *      * /* - RC=4034   Not numeric (VAR1)      */  
4470 *      * /* - RC=4051   Invalid JJJ (VAR1)      */  
4471 *      * /*****
```

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				01068	4473	PJMDCY	EQU *	Process CYJ-*
					4474	*		Process JCJ-*
					4475	*	*/-----/	
					4476	*	/* Check for valid FUNCT in last 4 bytes and including */	
					4477	*	/* valid letters in MDCY result build */	
					4478	*	*/-----/	
001068	4120	DB0D	00B0D		4479		LA R2,FUNCT+4	Starting address
00106C	4130	0004	00004		4480		LA R3,4	Number of times
001070	D503	DB0D	DBBA 00B0D	00BBA	4481		CLC FUNCT+4(4),BLANKS	BLANK ?
001076	4780	B86C	0186C		4482		BE ERR4011	Yes, invalid function
				0107A	4483	PJCHKF	EQU *	
00107A	95D4	2000	00000		4484		CLI 0(R2),C'M'	Valid character?
00107E	4780	B0A6	010A6		4485		BE PJCHKFNX	Yes, check next
001082	95C4	2000	00000		4486		CLI 0(R2),C'D'	Valid character?
001086	4780	B0A6	010A6		4487		BE PJCHKFNX	Yes, check next
00108A	95C3	2000	00000		4488		CLI 0(R2),C'C'	Valid character?
00108E	4780	B0A6	010A6		4489		BE PJCHKFNX	Yes, check next
001092	95E8	2000	00000		4490		CLI 0(R2),C'Y'	Valid character?
001096	4780	B0A6	010A6		4491		BE PJCHKFNX	Yes, check next
00109A	9540	2000	00000		4492		CLI 0(R2),C' '	Blank delimiter?
00109E	4780	B0AE	010AE		4493		BE PJDATE	Yes, done, go process
0010A2	47F0	B86C	0186C		4494		B ERR4011	no, must be invalid function
				010A6	4495	PJCHKFNX	EQU *	
0010A6	4122	0001	00001		4496		LA R2,1(R2)	
0010AA	4630	B07A	0107A		4497		BCT R3,PJCHKF	Check next character...
					4498	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				010AE	4500	PJDATE	EQU *	Process CYJ-D8, JCY-D8
				010AE	4501	PJDAY	EQU *	Process CYJ-DAY, JCY-DAY
				010AE	4502	PJDOW	EQU *	Process CYJ-DOW, JCY-DOW
0010AE	4550	BA3E	01A3E		4503		BAL R5,S#GVAR1	Get VAR1
0010B2	4550	BC4A	01C4A		4504		BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2
0010B6	5070	D05C	0005C		4505		ST R7,R7HOLD	Save VAR1 value Length
0010BA	5080	D060	00060		4506		ST R8,R8HOLD	Save VAR1 value Address
0010BE	4150	0000	00000		4507		LA R5,0	
				010C2	4508	PJDLL	EQU *	
0010C2	9540	8000	00000		4509		CLI 0(R8),C' '	Blank delimiter?
0010C6	4780	B0D6	010D6		4510		BE PJDLLX	Yes, go process
0010CA	4188	0001	00001		4511		LA R8,1(R8)	Bump up VAR1
0010CE	4155	0001	00001		4512		LA R5,1(R5)	Bump up VAR1 length
0010D2	4670	B0C2	010C2		4513		BCT R7,PJDLL	...do it again
				010D6	4514	PJDLLX	EQU *	
0010D6	1875				4515		LR R7,R5	Implied VAR1 length
					4516	*	*/-----/	
					4517	*	/* Check VAR1 for proper length */	
					4518	*	*/-----/	
0010D8	5970	C82C	0282C		4519		C R7,=F'7'	VAR1 length = 7 ?
0010DC	4770	B9A0	019A0		4520		BNE ERR4040	No, Error- VAR1 wrong length
0010E0	5880	D060	00060		4521		L R8,R8HOLD	Restore VAR1 value Address
					4522	*	*/-----/	
					4523	*	/* Check VAR1 for numerics based on LENGTH (R7) */	
					4524	*	*/-----/	
0010E4	4550	BE60	01E60		4525		BAL R5,S#ISNUM	Numeric Test
0010E8	59F0	C800	02800		4526		C R15,=F'4000'	Numeric?
0010EC	4780	B94C	0194C		4527		BE ERR4034	NO, Error
					4528	*		YES, Continue...
0010F0	5870	D05C	0005C		4529		L R7,R7HOLD	Restore VAR1 value Length
0010F4	5880	D060	00060		4530		L R8,R8HOLD	Restore VAR1 value Address
					4531	*	*/-----/	
					4532	*	/* Store VAR1 date (R8=CCYYJJJ or JJJCCYY) */	
					4533	*	*/-----/	
					4534	*	input VAR1	
					4535	*	output DCCCCYY DCJJJ	
0010F8	4550	BFE8	01FE8		4536		BAL R5,S#ICYJ	Store VAR1 date
0010FC	F222	D254 D247	00254 00247		4537		PACK PKJJJ,DCJJJ	PACK JJJ of VAR1
					4538	*	*/-----/	
					4539	*	/* Determine if CCYY is leap year */	
					4540	*	*/-----/	
					4541	*	input DCCCCYY 0CCYY	
					4542	*	output DC#LEAP 0=not leap year 1=leap year	
					4543	*	DC#YDAYS 365=common year 366=leap year	
001102	4550	BE8E	01E8E		4544		BAL R5,S#ISLPY	Determine Leap Year
					4545	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4547	*	*/-----/	
					4548	*	/* Check VAR1 for valid JJJ	*/
					4549	*	*/-----/	
001106	F921	D254	D23A	00254	0023A	4550	CP PKJJJ,DC#YDAYS	Valid MAX JJJ for Year
00110C	4720	B9D8		019D8		4551	BH ERR4051	No, VAR1 invalid JJJ
001110	F920	D254	C88E	00254	0288E	4552	CP PKJJJ,=P'1'	Valid MIN JJJ for Year
001116	4740	B9D8		019D8		4553	BL ERR4051	No, VAR1 invalid JJJ
					4554	*	*/-----/	
					4555	*	/* Determine MM and DD from JJJ	*/
					4556	*	*/-----/	
					4557	*	input DC#LEAP PKJJJ	
					4558	*	Work FW, R3, R4, R5, R6	
					4559	*	output DC#TRM DCM MM DCDDD	
					4560	*	R4=addr of MONTHNAME R6=Length of MONTHNAME	
00111A	4550	C148		02148		4561	BAL R5,S#J2MD	Determine MM and DD from JJJ
					4562	*	*/-----/	
					4563	*	/* Start of result content build ??LB	*/
					4564	*	*/-----/	
00111E	4120	D6E4		006E4		4565	LA R2,VARWORK	Addr of VARWORK
001122	4130	0000		00000		4566	LA R3,0	Length of Result
					4567	*	*/-----/	
					4568	*	/* Process D8, DAY, DOW or MDCY result	*/
					4569	*	*/-----/	
001126	D503	DB0D	C830	00B0D	02830	4570	CLC FUNCT+4(4),=C'D8	' Date ? DECEMBER 31, 2020
00112C	4780	B1A4		011A4		4571	BE DO#DATE	Yes, do DATE
001130	D503	DB0D	C834	00B0D	02834	4572	CLC FUNCT+4(4),=C'DAY	' Day ? MONDAY
001136	4780	B1C2		011C2		4573	BE DO#DOWDY	Yes, do Weekday Name
00113A	D503	DB0D	C838	00B0D	02838	4574	CLC FUNCT+4(4),=C'DOW	' DOW ? 3
001140	4780	B1AC		011AC		4575	BE DO#DOW	Yes, do DOW
					4576	*		No, must be MDCY

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4578	* * /-----/	
					4579	* * /* Build M D C Y J content based on FUNCT */	
					4580	* * /-----/	
			01144		4581	DO#MDCY EQU *	Build MDCY
001144	D504	DB09	C874	00B09	02874	4582 CLC FUNCT(5),=C'MDCY-'	MDCY- Function ?
00114A	4780	B184		01184	4583 BE DO#MDCY3	Yes, apply MDCY- controls	
00114E	D504	DB09	C879	00B09	02879	4584 CLC FUNCT(5),=C'DMCY-'	DMCY- Function ?
001154	4780	B184		01184	4585 BE DO#MDCY3	Yes, apply MDCY- controls	
001158	D504	DB09	C87E	00B09	0287E	4586 CLC FUNCT(5),=C'CYMD-'	CYMD- Function ?
00115E	4780	B184		01184	4587 BE DO#MDCY3	Yes, apply MDCY- controls	
001162	D504	DB09	C883	00B09	02883	4588 CLC FUNCT(5),=C'CYDM-'	CYDM- Function ?
001168	4780	B184		01184	4589 BE DO#MDCY3	Yes, apply MDCY- controls	
00116C	D503	DB09	C7F8	00B09	027F8	4590 CLC FUNCT(4),=C'CYJ-'	CYJ- Function?
001172	4780	B190		01190	4591 BE DO#MDCY4	Yes, apply CYJ- controls	
001176	D503	DB09	C7FC	00B09	027FC	4592 CLC FUNCT(4),=C'JCY-'	JCY- Function?
00117C	4780	B190		01190	4593 BE DO#MDCY4	Yes, apply CYJ- controls	
001180	47F0	B86C		0186C	4594 B ERR4011	no, must be invalid function	
			01184		4595	DO#MDCY3 EQU *	
001184	4140	DB0E		00B0E	4596 LA R4,FUNCT+5	MDCY- Function start addr	
001188	4160	0003		00003	4597 LA R6,3	Limit to 3	
00118C	47F0	B19C		0119C	4598 B DO#MDCY9		
			01190		4599	DO#MDCY4 EQU *	
001190	4140	DB0D		00B0D	4600 LA R4,FUNCT+4	CYJ- Function start addr	
001194	4160	0004		00004	4601 LA R6,4	Limit to 4	
001198	47F0	B19C		0119C	4602 B DO#MDCY9		
					4603	* * /-----/	
					4604	* * /* Call S#BMDCY to build M D C Y J content */	
					4605	* * /-----/	
					4606	* * /* R2 - addr VARWORK */	
					4607	* * /* R3 - Length of content in VARWORK */	
					4608	* * /* R4 - Addr of FUNCT+4 */	
					4609	* * /* R6 - Loop control for monthname */	
					4610	* * /-----/	
			0119C		4611	DO#MDCY9 EQU *	
00119C	4550	C0C8		020C8	4612 BAL R5,S#BMDCY		
0011A0	47F0	B1F0		011F0	4613 B PJMDCYX	Done	
					4614	* * /-----/	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				011A4	4616	DO#DATE	EQU *	Build MONTHNAME
					4617	*	*	/-----/
					4618	*	*	/* Call S#BD8 to build DATE (monthname dd, ccy) */
					4619	*	*	/-----/
					4620	*	*	/* R2 - addr VARWORK */
					4621	*	*	/* R3 - Length of content in VARWORK */
					4622	*	*	/* R4 - Addr of MONTHNAME */
					4623	*	*	/* R6 - Loop control for monthname */
					4624	*	*	/-----/
0011A4	4550	BEF2	01EF2		4625		BAL R5,S#BD8	
0011A8	47F0	B1F0	011F0		4626		B PJMDCYX	Done
				011AC	4627	DO#DOW	EQU *	Build DOW
					4628	*	*	/-----/
					4629	*	*	/* Call S#CDOW to build DOW 3 */
					4630	*	*	/-----/
					4631	*		input DCDDD DCCCCCY DCMMD DC#TRM
					4632	*		output DCCDOW DCCWDAY
					4633	*		Working fields DC#YR4 DC#DD DC#YR100 DC#YR400 DC#DOW
0011AC	4550	BF50	01F50		4634		BAL R5,S#CDOW	
0011B0	D200	2000	D24A	00000	0024A	4635	MVC 0(1,R2),DCCDOW	Move N
0011B6	4122	0001	00001		4636		LA R2,1(R2)	Bump up VARWORK by 1 for 'n'
0011BA	4133	0001	00001		4637		LA R3,1(R3)	Bump up VARWORKL by 1 for 'n'
0011BE	47F0	B1F0	011F0		4638		B PJMDCYX	Done
				011C2	4639	DO#DOWDY	EQU *	Build DOW Day Name
					4640	*	*	/-----/
					4641	*	*	/* Call S#CDOW to build weekday Wednesday */
					4642	*	*	/-----/
					4643	*		input DCDDD DCCCCCY DCMMD DC#TRM
					4644	*		output DCCDOW DCCWDAY
					4645	*		Working fields DC#YR4 DC#DD DC#YR100 DC#YR400 DC#DOW
0011C2	4550	BF50	01F50		4646		BAL R5,S#CDOW	
0011C6	4140	D24B	0024B		4647		LA R4,DCCWDAY	R4=Day of Week name begin
0011CA	4160	0009	00009		4648		LA R6,L'DCCWDAY	R6=Length of Week Day Name
				011CE	4649	DO#DOWNM	EQU *	
0011CE	9540	4000	00000		4650		CLI 0(R4),C' '	Blank character?
0011D2	4780	B1F0	011F0		4651		BE PJMDCYX	Yes, done...
0011D6	D200	2000	4000	00000	00000	4652	MVC 0(1,R2),0(R4)	Move day of week name character
0011DC	4122	0001	00001		4653		LA R2,1(R2)	Bump up VARWORK
0011E0	4144	0001	00001		4654		LA R4,1(R4)	Bump up day of week name
0011E4	4133	0001	00001		4655		LA R3,1(R3)	Bump up VARWORKL
0011E8	4660	B1CE	011CE		4656		BCT R6,DO#DOWNM	Do next DOW name character
0011EC	47F0	B1F0	011F0		4657		B PJMDCYX	Done
				011F0	4658	PJMDYX	EQU *	
0011F0	1873				4659		LR R7,R3	Result length in R3
0011F2	4550	BBC4	01BC4		4660		BAL R5,S#UVAR2	Update VAR2
0011F6	5870	D05C	0005C		4661		L R7,R7HOLD	Restore VAR1 value Length
0011FA	47F0	B7E2	017E2		4662		B ERRDONE	Successful Request
					4663	*		

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
4665 *      * /*****  
4666 *40    * /* Function: MDCY-D8          REQ OPT      */  
4667 *41    * /* Function: MDCY-DAY        REQ OPT      */  
4668 *42    * /* Function: MDCY-DOW        REQ OPT      */  
4669 *43    * /* Function: MDCY-cyj        REQ OPT      */  
4670 *44    * /* Function: DMCY-D8          REQ OPT      */  
4671 *45    * /* Function: DMCY-DAY        REQ OPT      */  
4672 *46    * /* Function: DMCY-DOW        REQ OPT      */  
4673 *47    * /* Function: DMCY-cyj        REQ OPT      */  
4674 *48    * /* Function: CYMD-D8          REQ OPT      */  
4675 *49    * /* Function: CYMD-DAY        REQ OPT      */  
4676 *50    * /* Function: CYMD-DOW        REQ OPT      */  
4677 *51    * /* Function: CYMD-cyj        REQ OPT      */  
4678 *52    * /* Function: CYDM-D8          REQ OPT      */  
4679 *53    * /* Function: CYDM-DAY        REQ OPT      */  
4680 *54    * /* Function: CYDM-DOW        REQ OPT      */  
4681 *55    * /* Function: CYDM-cyj        REQ OPT      */  
4682 *      * /*  
4683 *      * /*          Date Conversion          */  
4684 *      * /*****  
4685 *      * /*          -D8 converts to monthname DD, CCYY      */  
4686 *      * /*          -DAY converts to day name          */  
4687 *      * /*          -DOW converts to day of week number      */  
4688 *      * /*          MDCY-* converts to YYCCJJJ (any order)      */  
4689 *      * /*          DMCY-* converts to YYCCJJJ (any order)      */  
4690 *      * /*          CYMD-* converts to YYCCJJJ (any order)      */  
4691 *      * /*          CYDM-* converts to YYCCJJJ (any order)      */  
4692 *      * /*  
4693 *      * /*          Date input form specified by function bytes 1-4:      */  
4694 *      * /*          MDCY --> MMDDCCYY      */  
4695 *      * /*          DMCY --> DDMMCCYY      */  
4696 *      * /*          CYMD --> CCYYMMDD      */  
4697 *      * /*          CYDM --> CCYYDDMM      */  
4698 *      * /*****  
4699 *      * /*          - R3, R4, R5, R6          Working Register      */  
4700 *      * /*          - R7 Length of VAR1 value (after BAL S#GVAR1)      */  
4701 *      * /*          - R8 Address of VAR1 value (after BAL S#GVAR1)      */  
4702 *      * /*****  
4703 *      * /*          - RC=0          Successful Request      */  
4704 *      * /*          - RC=4011 Invalid FUNCTION      */  
4705 *      * /*          - RC=4040 Invalid length (VAR1)      */  
4706 *      * /*          - RC=4047 Invalid DD (VAR1)      */  
4707 *      * /*          - RC=4034 Not numeric (VAR1)      */  
4708 *      * /*          - RC=4049 Invalid MM (VAR1)      */  
4709 *      * /*          - RC=4050 Invalid DD for MM (VAR1)      */  
4710 *      * /*          - RC=4051 Invalid JJJ (VAR1)      */  
4711 *      * /*****  
4712 *      *
```

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				011FE	4714	PMDCYJ	EQU *	Process MDCY-*
					4715	*		Process DMCY-*
					4716	*		Process CYMD-*
					4717	*		Process CYDM-*
					4718	*	* /-----/	
					4719	*	* /* Check for valid FUNCT in last 3 bytes and including */	
					4720	*	* /* valid letters in MDCY result build */	
					4721	*	* /-----/	
0011FE	4120	DB0E	00B0E		4722		LA R2,FUNCT+5	Starting address
001202	4130	0003	00003		4723		LA R3,3	Number of times
001206	D502	DB0E	DBBA 00B0E	00BBA	4724		CLC FUNCT+5(3),BLANKS	BLANK ?
00120C	4780	B86C	0186C		4725		BE ERR4011	Yes, invalid function
				01210	4726	PMCHKF	EQU *	
001210	95D1	2000	00000		4727		CLI 0(R2),C'J'	Valid character?
001214	4780	B234	01234		4728		BE PMCHKFNX	Yes, check next
001218	95C3	2000	00000		4729		CLI 0(R2),C'C'	Valid character?
00121C	4780	B234	01234		4730		BE PMCHKFNX	Yes, check next
001220	95E8	2000	00000		4731		CLI 0(R2),C'Y'	Valid character?
001224	4780	B234	01234		4732		BE PMCHKFNX	Yes, check next
001228	9540	2000	00000		4733		CLI 0(R2),C' '	Blank delimiter?
00122C	4780	B23C	0123C		4734		BE PMDATE	Yes, done, go process
001230	47F0	B86C	0186C		4735		B ERR4011	no, must be invalid function
				01234	4736	PMCHKFNX	EQU *	
001234	4122	0001	00001		4737		LA R2,1(R2)	
001238	4630	B210	01210		4738		BCT R3,PMCHKF	Check next character...
					4739	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				0123C	4741	PMDATE	EQU *	Process MDCY-D8 DMCY-D8
					4742	*		CYMD-D8 CYDM-D8
				0123C	4743	PMDAY	EQU *	Process MDCY-DAY DMCY-DAY
					4744	*		CYMD-DAY CYDM-DAY
				0123C	4745	PMDOW	EQU *	Process MDCY-DOW DMCY-DOW
					4746	*		CYMD-DOW CYDM-DOW
00123C	4550	BA3E	01A3E		4747	BAL	R5,S#GVAR1	Get VAR1
001240	4550	BC4A	01C4A		4748	BAL	R5,S#CHKV2	Reset VAR2 when \$NOVAR2
001244	5070	D05C	0005C		4749	ST	R7,R7HOLD	Save VAR1 value Length
001248	5080	D060	00060		4750	ST	R8,R8HOLD	Save VAR1 value Address
00124C	4150	0000	00000		4751	LA	R5,0	
				01250	4752	PMDDL	EQU *	
001250	9540	8000	00000		4753	CLI	0(R8),C' '	Blank delimiter?
001254	4780	B264	01264		4754	BE	PMDDLX	Yes, go process
001258	4188	0001	00001		4755	LA	R8,1(R8)	Bump up VAR1
00125C	4155	0001	00001		4756	LA	R5,1(R5)	Bump up VAR1 length
001260	4670	B250	01250		4757	BCT	R7,PMDDL	...do it again
				01264	4758	PMDDLX	EQU *	
001264	1875				4759	LR	R7,R5	Implied VAR1 length
					4760	*	*/-----/	
					4761	*	/* Check VAR1 for proper length */	
					4762	*	*/-----/	
001266	5970	C808	02808		4763	C	R7,=F'8'	VAR1 length = 8 ?
00126A	4770	B9A0	019A0		4764	BNE	ERR4040	No, Error- VAR1 wrong length
00126E	5880	D060	00060		4765	L	R8,R8HOLD	Restore VAR1 value Address
					4766	*	*/-----/	
					4767	*	/* Check VAR1 for numerics based on LENGTH (R7) */	
					4768	*	*/-----/	
001272	4550	BE60	01E60		4769	BAL	R5,S#ISNUM	Numeric Test
001276	59F0	C800	02800		4770	C	R15,=F'4000'	Numeric?
00127A	4780	B94C	0194C		4771	BE	ERR4034	NO, Error
					4772	*		YES, Continue...
00127E	5870	D05C	0005C		4773	L	R7,R7HOLD	Restore VAR1 value Length
001282	5880	D060	00060		4774	L	R8,R8HOLD	Restore VAR1 value Address
					4775	*	*/-----/	
					4776	*	/* Store VAR1 date (R8=MMDDCCYY or variation of MDCY) */	
					4777	*	*/-----/	
					4778	*	input VAR1	
					4779	*	output DCCCCYY DCJJJ	
001286	4550	C02E	0202E		4780	BAL	R5,S#IMDCY	Store VAR1 date
					4781	*	*/-----/	
					4782	*	/* Determine if CCYY is leap year */	
					4783	*	*/-----/	
					4784	*	input DCCCCYY 0CCYY	
					4785	*	output DC#LEAP 0=not leap year 1=leap year	
					4786	*	DC#YDAYS 365=common year 366=leap year	
00128A	4550	BE8E	01E8E		4787	BAL	R5,S#ISLPY	Determine Leap Year
					4788	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4790 *	* /-----/	
					4791 *	* /* Determine JJJ from MM DD	*/
					4792 *	* /-----/	
					4793 *	input DC#LEAP n packed	
					4794 *	DCDDD 0DDD	
					4795 *	DCMMM 0MMM	
					4796 *	output DC#JJJ JJJ packed	
					4797 *	DC#TRM n packed	
					4798 *	DC#DD DD packed	
					4799 *	DCJJJ JJJ	
					4800 *	R4=addr of MONTHNAME	
					4801 *	R6=Length of MONTHNAME	
00128E	4550	C1B0	021B0		4802 *	BAL R5,S#MD2J MMDD to JJJ	
					4803 *	* /-----/	
					4804 *	* /* Start of result content build	*/
					4805 *	* /-----/	
001292	4120	D6E4	006E4		4806 *	LA R2,VARWORK Addr of VARWORK	
001296	4130	0000	00000		4807 *	LA R3,0 Length of Result	
					4808 *	* /-----/	
					4809 *	* /* Process D8, DAY, DOW or CYJ result	*/
					4810 *	* /-----/	
00129A	D502	DB0E	C88F	00B0E	0288F	4811	CLC FUNCT+5(3),=C'D8 ' ? DECEMBER 31, 2020 MDCY-d
0012A0	4780	B1A4	011A4			4812	BE DO#DATE Yes, do DATE
0012A4	D502	DB0E	C892	00B0E	02892	4813	CLC FUNCT+5(3),=C'DAY' ? MONDAY
0012AA	4780	B1C2	011C2			4814	BE DO#DOWDY Yes, do Weekday Name
0012AE	D502	DB0E	C895	00B0E	02895	4815	CLC FUNCT+5(3),=C'DOW' ? 3
0012B4	4780	B1AC	011AC			4816	BE DO#DOW Yes, do DOW
0012B8	47F0	B144	01144			4817	B DO#MDCY No, Must be CYJ request
					4818 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				4820 *	/* *****	*/
				4821 *56	/* Function: FILL	REQ REQ
				4822 *67	/* Function: PAD	REQ REQ
				4823 *	/* *****	*/
				4824 *	/* Fill VAR1 with a 1-byte character.	*/
				4825 *	/* VAR2 contains 1-byte FILLing character and up to 3	*/
				4826 *	/* digits representing VAR1 length.	*/
				4827 *	/* e.g. "*33" FILL VAR1 with * for a length of 33	*/
				4828 *	/*	*/
				4829 *	/* Pad VAR1 with a 1-byte character.	*/
				4830 *	/* VAR2 contains 1-byte PADding character and up to 3	*/
				4831 *	/* digits representing VAR1 length.	*/
				4832 *	/* e.g. "*33" PAD VAR1 with * for a length of 33	*/
				4833 *	/*	*/
				4834 *	/* VAR2 contains Parm s -----	*/
				4835 *	/* Format: c n n n	*/
				4836 *	/* where;	*/
				4837 *	/* c - single byte character	*/
				4838 *	/* n - 1-3 digit length (1-256)	*/
				4839 *	/*	*/
				4840 *	/* - R2, R3, R4, R5 Working Register	*/
				4841 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
				4842 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
				4843 *	/* *****	*/
				4844 *	/* - RC=0 Successful Request	*/
				4845 *	/* - RC=4015 VAR2 is required	*/
				4846 *	/* - RC=4016 VAR2 not found	*/
				4847 *	/* - RC=4040 Invalid length (VAR2 content or Length)	*/
				4848 *	/* - RC=4034 Not numeric (Length)	*/
				4849 *	/* *****	*/
		012BC		4850 PFILL	EQU *	Process FILL
		012BC		4851 PPAD	EQU *	Process PAD
0012BC	4550 BA3A	01A3A		4852	BAL R5,S#GVAR1N	Get VAR1, do not check for NULL
0012C0	5070 D05C	0005C		4853	ST R7,R7HOLD	Save VAR1 value Length
0012C4	5080 D060	00060		4854	ST R8,R8HOLD	Save VAR1 value Address
0012C8	0670			4855	BCTR R7,0	Adjust for EXecute
0012CA	4470 DCBA	00CBA		4856	EX R7,MVCVARWK	Execute VARWORK value move
				4857 *MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
0012CE	4177 0001	00001		4858	LA R7,1(R7)	Reset after EXecute
				4859 *	/* -----	*/
				4860 *	/* Need VAR2 content in position 1	*/
				4861 *	/* -----	*/
0012D2	9540 DB19	00B19		4862	CLI VAR2,C' '	VAR2 name captured?
0012D6	4780 B896	01896		4863	BE ERR4015	NO, VAR2 required
				4864 *	/* -----	*/
				4865 *	/* Link to IKJCT441 to obtain VAR2	*/
				4866 *	/* -----	*/
0012DA	4550 BB50	01B50		4867	BAL R5,S#GVAR2	Get VAR2
0012DE	58F0 D04C	0004C		4868	L R15,RC#441	Restore R15
0012E2	12FF			4869	LTR R15,R15	Successful IKJCT441?
0012E4	4770 B8A4	018A4		4870	BNZ ERR4016	No, VAR2 not found
				4871 *	/* -----	*/
				4872 *	/* Update VAR2 Length	*/
				4873 *	/* -----	*/
0012E8	4550 BD58	01D58		4874	BAL R5,S#UV2L	Update VAR2L variable

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	16.49	04/12/20
-----	--------	------	-------	-------	------	--------	-----------	-----	------	-------	----------

4875 *

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4877 *	/* -----/	
					4878 *	/* Check VAR2 length limits VAR2 L = 1-4 */	
					4879 *	/* -----/	
0012EC	5820	D294	00294		4880	L R2,VALP2	R2 Value Ptr retrieved/created
0012F0	5840	D290	00290		4881	L R4,VALL2	R4 Value Len retrieved/created
0012F4	4130	0000	00000		4882	LA R3,0	R3 Length of VAR2
			012F8		4883 FILLLEN	EQU *	
0012F8	4122	0001	00001		4884	LA R2,1(R2)	Bump up VAR2
0012FC	4133	0001	00001		4885	LA R3,1(R3)	Bump up VAR2 Length
001300	9540	2000	00000		4886	CLI 0(R2),C' '	Blank delimiter?
001304	4780	B30C	0130C		4887	BE GOTFLEN	Yes, go process
001308	4640	B2F8	012F8		4888	BCT R4,FILLLEN	
			0130C		4889 GOTFLEN	EQU *	
00130C	5930	C828	02828		4890	C R3,=F'4'	Variable length > 4?
001310	4720	B9A0	019A0		4891	BH ERR4040	No, Error- VAR2 wrong length
001314	5930	C7F4	027F4		4892	C R3,=F'1'	Variable length = 0?
001318	4740	B9A0	019A0		4893	BL ERR4040	No, Error- VAR2 wrong length
					4894 *		must be 1 or greater
					4895 *		R3=Length of VAR2
00131C	5030	D290	00290		4896	ST R3,VALL2	Store length of VAR2
					4897 *	/* -----/	
					4898 *	/* Get 1-byte delimiter from VAR2 */	
					4899 *	/* -----/	
001320	5820	D294	00294		4900	L R2,VALP2	R2 Value Ptr retrieved/created
001324	D200	D9EA 2000	009EA 00000		4901	MVC DELIM(1),0(R2)	Save 1 byte delimiter
					4902 *	/* -----/	
					4903 *	/* Check VAR2 FILL Len for numerics based on LENGTH (R7)*/	
					4904 *	/* -----/	
00132A	5930	C7F4	027F4		4905	C R3,=F'1'	Only delimiter?
00132E	4780	B34A	0134A		4906	BE GETLFILL	Yes, done...
					4907 *		No, continue to edit
					4908 *	/* - R7 Length */	
					4909 *	/* - R8 Address of Content */	
					4910 *	/* - R15 Return Code */	
					4911 *	/* 4001-TRUE */	
					4912 *	/* 4000-FALSE */	
001332	1873				4913	LR R7,R3	R3=Length of FILL
001334	0670				4914	BCTR R7,0	Adjust for delimiter
001336	5880	D294	00294		4915	L R8,VALP2	Restore VAR2 value Address
00133A	4188	0001	00001		4916	LA R8,1(R8)	Adjust for delimiter
00133E	4550	BE60	01E60		4917	BAL R5,S#ISNUM	Numeric Test
001342	59F0	C800	02800		4918	C R15,=F'4000'	Numeric?
001346	4780	B94C	0194C		4919	BE ERR4034	NO, Error
					4920 *		We are numeric...
					4921 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					4923 *	*/-----/	
					4924 *	/* Get FILL length	*/
					4925 *	*/-----/	
			0134A		4926 GETLFILL	EQU *	
					4927 *	STM R1,R2,DW	SAVE R1,R2 to DW before TPUT
					4928 *	LA R1,=C' getlfill	
					4929 *	LA R2,20	
					4930 *	TPUT (R1),(R2)	
					4931 *	LM R1,R2,DW	RESTORE R1,R2 from DW after TPUT
00134A	5820	D294	00294		4932	L R2,VALP2	R2 Value Ptr retrieved/created
00134E	5930	C83C	0283C		4933	C R3,=F'2'	VAR2 Length = 2?
001352	4780	B36A	0136A		4934	BE LFILL1	Yes, len = 1
001356	5930	C7DC	027DC		4935	C R3,=F'3'	VAR2 Length = 3?
00135A	4780	B374	01374		4936	BE LFILL2	Yes, len = 2
00135E	5930	C828	02828		4937	C R3,=F'4'	VAR2 Length = 4?
001362	4780	B37E	0137E		4938	BE LFILL3	Yes, len = 3
001366	47F0	B388	01388		4939	B DFLTFILL	No, default to 1
			0136A		4940 LFILL1	EQU *	FILL len = 1 byte
00136A	F270	D218	2001 00218	00001	4941	PACK DW,1(1,R2)	Pack 1 digit after delim
001370	47F0	B38E	0138E		4942	B LFILLX	
			01374		4943 LFILL2	EQU *	FILL len = 2 bytes
001374	F271	D218	2001 00218	00001	4944	PACK DW,1(2,R2)	Pack 2 digits after delim
00137A	47F0	B38E	0138E		4945	B LFILLX	
			0137E		4946 LFILL3	EQU *	FILL len = 3 bytes
00137E	F272	D218	2001 00218	00001	4947	PACK DW,1(3,R2)	Pack 3 digits after delim
001384	47F0	B38E	0138E		4948	B LFILLX	
			01388		4949 DFLTFILL	EQU *	
001388	F270	D218	C898 00218	02898	4950	PACK DW,=C'1'	Default to len = 1
			0138E		4951 LFILLX	EQU *	
00138E	4F30	D218	00218		4952	CVB R3,DW	Convert to binary
					4953 *	*/-----/	
					4954 *	/* Check FILL length for variable limits	*/
					4955 *	*/-----/	
001392	5930	C840	02840		4956	C R3,=F'256'	Variable length > 256?
001396	4720	B9A0	019A0		4957	BH ERR4040	Cannot process. 256 limit.
00139A	5930	C7D8	027D8		4958	C R3,=F'0'	Variable length = 256?
00139E	4780	B9A0	019A0		4959	BE ERR4040	Yes, null value. Exit.
					4960 *		R3= Length of FILL
					4961 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					4963	*	*/-----/	
					4964	*	/* Fill VARWORK w delimiter value	*/
					4965	*	*/-----/	
0013A2	4120	D6E4		006E4	4966	LA	R2,VARWORK	Addr of VARWORK
0013A6	1843				4967	LR	R4,R3	Loop control - R3 times
0013A8	D507	DB09	C760	00B09	4968	CLC	FUNCT,=C'FILL	FILL Function ?
0013AE	4780	B3C0		013C0	4969	BE	FILLEM	Yes, go to FILLEM
					4970	*		No, must be PAD function
0013B2	5870	D05C		0005C	4971	L	R7,R7HOLD	Restore VAR1 value Length
0013B6	1974				4972	CR	R7,R4	VAR1 Length > Pad Length?
0013B8	47B0	B3D0		013D0	4973	BNL	PPADX	No, no need to pad VAR1
					4974	*		Yes, pad VAR1
0013BC	1A27				4975	AR	R2,R7	Position to end of VAR1
0013BE	1B47				4976	SR	R4,R7	Adjust loop control
				013C0	4977	FILLEM	EQU *	
0013C0	D200	2000	D9EA	00000	4978	MVC	0(1,R2),DELIM	Move DELIM to VARWORK
0013C6	4122	0001		00001	4979	LA	R2,1(R2)	Bump up VAR1
0013CA	4640	B3C0		013C0	4980	BCT	R4,FILLEM	Again...
					4981	*	*/-----/	
					4982	*	/* Update VAR1 w VARWORK value	*/
					4983	*	*/-----/	
				013CE	4984	PFILLX	EQU *	
0013CE	1873				4985	LR	R7,R3	Final Length as VAR1L
				013D0	4986	PPADX	EQU *	
0013D0	4120	D6E4		006E4	4987	LA	R2,VARWORK	VARWORK area as VAR1
0013D4	5020	D294		00294	4988	ST	R2,VALP2	
0013D8	9240	DB19		00B19	4989	MVI	VAR2,C' '	Force VAR1 to be updated
0013DC	4550	BBC4		01BC4	4990	BAL	R5,S#UVAR2	Update VAR1, no update to VAR2
0013E0	47F0	B7E2		017E2	4991	B	ERRDONE	
					4992	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				4994 *	*/*****	
				4995 *01	/* Function: LTRIM REQ OPT	
				4996 *57	/* Function: LSTRIP REQ OPT	
				4997 *	*/*****	
				4998 *	/* LTRIM : Strip leading spaces	
				4999 *	/* LSTRIP: Strip leading delimiter, default=BLANK	
				5000 *	/* - R2, R5 Working Register	
				5001 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
				5002 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
				5003 *	*/*****	
				5004 *	/* - RC=0 Successful Request	
				5005 *	*/*****	
		013E4		5006 PLSTRIP	EQU * Process LSTRIP	
0013E4	4550 BA3E	01A3E		5007	BAL R5,S#GVAR1 Get VAR1	
0013E8	4550 BC4A	01C4A		5008	BAL R5,S#CHKV2 Reset VAR2 when \$NOVAR2	
				5009 *	*/-----	
				5010 *	/* VAR2 content area initial value is blanks	
				5011 *	/* regardless of S#GVAR2 return	
				5012 *	*/-----	
0013EC	D507 DB09 C5D0	00B09	025D0	5013	CLC FUNCT,=C'LTRIM ' LTRIM Function ?	
0013F2	4780 B40C	0140C		5014	BE STRIPL Yes, DELIM = BLANK at init.	
				5015 *	*/-----	
				5016 *	/* Link to IKJCT441 to obtain VAR2	
				5017 *	*/-----	
0013F6	4550 BB50	01B50		5018	BAL R5,S#GVAR2 Get VAR2	
				5019 *	*/-----	
				5020 *	/* Update VAR2 Length	
				5021 *	*/-----	
0013FA	4550 BD58	01D58		5022	BAL R5,S#UV2L Update VAR2L variable	
				5023 *	*/-----	
				5024 *	/* Point to VAR2 value and length address	
				5025 *	*/-----	
0013FE	5820 D294	00294		5026	L R2,VALP2 R2 Value Ptr retrieved/created	
001402	5840 D290	00290		5027	L R4,VALL2 R4 Value Len retrieved/created	
				5028 *	*/-----	
				5029 *	/* Get 1-byte delimiter from VAR2	
				5030 *	/* BLANK assumed if not VAR2 specified	
				5031 *	*/-----	
001406	D200 D9EA 2000	009EA	00000	5032	MVC DELIM(1),0(R2) Save 1 byte delimiter	
				5033 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5035	*	*	/-----/
					5036	*	*	/* Skip LEADING delimiter character */
					5037	*	*	/-----/
				0140C	5038	STRIPL	EQU *	
00140C	D500	8000	D9EA	00000	009EA	5039	CLC 0(1,R8),DELIM	Strip character ?
001412	4770	B41E		0141E		5040	BNE STRIPLM	NO, move remaining data
001416	4188	0001		00001		5041	LA R8,1(R8)	YES, bump to next position
00141A	4670	B40C		0140C		5042	BCT R7,STRIPL	Decrement length while DELIM
					5043	*	*	/-----/
					5044	*	*	/* Move remaining data */
					5045	*	*	/-----/
				0141E	5046	STRIPLM	EQU *	
00141E	0670				5047	BCTR	R7,0	Adjust for EXecute
001420	4470	DCBA		00CBA	5048	EX	R7,MVCVARWK	Execute VARWORK value move
					5049	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
001424	4177	0001		00001	5050	LA	R7,1(R7)	Reset after EXecute
001428	D507	DB09	C5D0	00B09	025D0	5051	CLC FUNCT,=C'LTRIM '	LTRIM Function ?
00142E	4780	B43E		0143E	5052	BE	PLSTRIPX	Yes, update VAR2 and exit
					5053	*	*	No, update VAR1 and exit
					5054	*	*	/-----/
					5055	*	*	/* Force VAR1 update with VARWORK content */
					5056	*	*	/-----/
001432	4120	D6E4		006E4	5057	LA	R2,VARWORK	VARWORK area as VAR1
001436	5020	D294		00294	5058	ST	R2,VALP2	
00143A	9240	DB19		00B19	5059	MVI	VAR2,C' '	Force VAR1 to be updated
				0143E	5060	PLSTRIPX	EQU *	
00143E	4550	BBC4		01BC4	5061	BAL	R5,S#UVAR2	Update VAR2
001442	47F0	B7E2		017E2	5062	B	ERRDONE	
					5063	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				5065 *	*/*****	
				5066 *02 *	/* Function: RTRIM REQ OPT	
				5067 *58 *	/* Function: RSTRIP REQ OPT	
				5068 *	*/*****	
				5069 *	/* RTRIM : Strip trailing spaces	
				5070 *	/* RSTRIP: Strip trailing delimiter, default=BLANK	
				5071 *	/* - R2, R5 Working Register	
				5072 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
				5073 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
				5074 *	*/*****	
				5075 *	/* - RC=0 Successful Request	
				5076 *	*/*****	
		01446		5077 PRSTRIP EQU *	Process RSTRIP	
001446	4550 BA3E	01A3E		5078 BAL R5,S#GVAR1	Get VAR1	
00144A	4550 BC4A	01C4A		5079 BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2	
00144E	1A87			5080 AR R8,R7	Add VAR1 length to start addr	
001450	0680			5081 BCTR R8,0	..to determine VAR1 end addr	
				5082 *	/*-----	
				5083 *	/* VAR2 content area initial value is blanks	
				5084 *	/* regardless of S#GVAR2 return	
				5085 *	/*-----	
001452	D507 DB09 C5D8	00B09 025D8		5086 CLC FUNCT,=C'RTRIM	RTRIM Function ?	
001458	4780 B472	01472		5087 BE STRIPR	Yes, DELIM = BLANK at init.	
				5088 *	/*-----	
				5089 *	/* Link to IKJCT441 to obtain VAR2	
				5090 *	/*-----	
00145C	4550 BB50	01B50		5091 BAL R5,S#GVAR2	Get VAR2	
				5092 *	/*-----	
				5093 *	/* Update VAR2 Length	
				5094 *	/*-----	
001460	4550 BD58	01D58		5095 BAL R5,S#UV2L	Update VAR2L variable	
				5096 *	/*-----	
				5097 *	/* Point to VAR2 value and length address	
				5098 *	/*-----	
001464	5820 D294	00294		5099 L R2,VALP2	R2 Value Ptr retrieved/created	
001468	5840 D290	00290		5100 L R4,VALL2	R4 Value Len retrieved/created	
				5101 *	/*-----	
				5102 *	/* Get 1-byte delimiter from VAR2	
				5103 *	/* BLANK assumed if not VAR2 specified	
				5104 *	/*-----	
00146C	D200 D9EA 2000	009EA 00000		5105 MVC DELIM(1),0(R2)	Save 1 byte delimiter	
				5106 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5108	*	*/-----/	
					5109	*	/* Skip TRAILING delimiter character	*/
					5110	*	*/-----/	
				01472	5111	STRIPR	EQU *	
001472	D500	8000	D9EA	00000	009EA	5112	CLC 0(1,R8),DELIM	Strip character ?
001478	4770	B482		01482	5113		BNE STRIPRM	NO, move data
00147C	0680				5114		BCTR R8,0	YES, bump to prev position
00147E	4670	B472		01472	5115		BCT R7,STRIPR	Decrement length while DELIM
					5116	*	*/-----/	
					5117	*	/* Move remaining data	*/
					5118	*	*/-----/	
				01482	5119	STRIPRM	EQU *	
					5120	*		R7=VAR1 revised length
001482	5880	D284		00284	5121		L R8,VALP1	Reset to VAR1 value address
001486	0670				5122		BCTR R7,0	Adjust for EXecute
001488	4470	DCBA		00CBA	5123		EX R7,MVCVARWK	Execute VARWORK value move
					5124	*MVCVARWK	MVC VARWORK(0),0(R8)	Save VARWORK value
00148C	4177	0001		00001	5125		LA R7,1(R7)	Reset after EXecute
001490	D507	DB09	C5D8	00B09	025D8	5126	CLC FUNCT,=C'RTRIM '	RTRIM Function ?
001496	4780	B4A6		014A6	5127		BE PRSTRIPX	Yes, update VAR2 and exit
					5128	*		No, update VAR1 and exit
					5129	*	*/-----/	
					5130	*	/* Force VAR1 update with VARWORK content	*/
					5131	*	*/-----/	
00149A	4120	D6E4		006E4	5132		LA R2,VARWORK	VARWORK area as VAR1
00149E	5020	D294		00294	5133		ST R2,VALP2	
0014A2	9240	DB19		00B19	5134		MVI VAR2,C' '	Force VAR1 to be updated
				014A6	5135	PRSTRIPX	EQU *	
0014A6	4550	BBC4		01BC4	5136		BAL R5,S#UVAR2	Update VAR2
0014AA	47F0	B7E2		017E2	5137		B ERRDONE	
					5138	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				5140 *	*/*****	*/
				5141 *03	*/ * Function: TRIM	REQ OPT */
				5142 *59	*/ * Function: STRIP	REQ OPT */
				5143 *	*/*****	*/
				5144 *	*/ * TRIM : Strip leading and trailing spaces	*/
				5145 *	*/ * STRIP: Strip leading and trailing delimiter,	*/
				5146 *	*/ * default=BLANK	*/
				5147 *	*/ * - R2, R5 Working Register	*/
				5148 *	*/ * - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
				5149 *	*/ * - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
				5150 *	*/*****	*/
				5151 *	*/ * - RC=0 Successful Request	*/
				5152 *	*/*****	*/
0014AE	4550 BA3E	01A3E	014AE	5153 PSTRIP	EQU * Process STRIP	
				5154	BAL R5,S#GVAR1 Get VAR1	
				5155 *	ST R7,R7HOLD Save VAR1 value Length	
0014B2	4550 BC4A	01C4A		5156 *	ST R8,R8HOLD Save VAR1 value Address	
				5157	BAL R5,S#CHKV2 Reset VAR2 when \$NOVAR2	
				5158 *	*/-----	*/
				5159 *	*/ * VAR2 content area initial value is blanks	*/
				5160 *	*/ * regardless of S#GVAR2 return	*/
				5161 *	*/-----	*/
0014B6	D507 DB09 C5E0	00B09	025E0	5162	CLC FUNCT,=C'TRIM ' TRIM Function ?	
0014BC	4780 B4D6	014D6		5163	BE STRIPLL Yes, DELIM = BLANK at init.	
				5164 *	*/-----	*/
				5165 *	*/ * Link to IKJCT441 to obtain VAR2	*/
				5166 *	*/-----	*/
0014C0	4550 BB50	01B50		5167	BAL R5,S#GVAR2 Get VAR2	
				5168 *	*/-----	*/
				5169 *	*/ * Update VAR2 Length	*/
				5170 *	*/-----	*/
0014C4	4550 BD58	01D58		5171	BAL R5,S#UV2L Update VAR2L variable	
				5172 *	*/-----	*/
				5173 *	*/ * Check VAR2 length limits VAR2 L = 1	*/
				5174 *	*/-----	*/
0014C8	5820 D294	00294		5175	L R2,VALP2 R2 Value Ptr retrieved/created	
0014CC	5840 D290	00290		5176	L R4,VALL2 R4 Value Len retrieved/created	
				5177 *	*/-----	*/
				5178 *	*/ * Get 1-byte delimiter from VAR2	*/
				5179 *	*/-----	*/
0014D0	D200 D9EA 2000	009EA	00000	5180	MVC DELIM(1),0(R2) Save 1 byte delimiter	
				5181 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5183	* * /-----/	
					5184	* * /* Skip LEADING delimiter character */	
					5185	* * /-----/	
				014D6	5186	STRIPLL EQU *	
0014D6	D500	8000	D9EA	00000	009EA	5187 CLC 0(1,R8),DELIM	Strip character ?
0014DC	4770	B4EC		014EC		5188 BNE STRIPLMM	NO, move remaining data
0014E0	4188	0001		00001		5189 LA R8,1(R8)	YES, Bump to next position
0014E4	4670	B4D6		014D6		5190 BCT R7,STRIPLL	Decrement length while DELIM
0014E8	47F0	B510		01510		5191 B STRIPRMM	Done...
					5192	* * /-----/	
					5193	* * /* Move remainint data */	
					5194	* * /-----/	
				014EC	5195	STRIPLMM EQU *	
0014EC	0670				5196	BCTR R7,0	Adjust for EXecute
0014EE	4470	DCBA		00CBA	5197	EX R7,MVCVARWK	Execute VARWORK value move
					5198	*MVCVARWK MVC VARWORK(0),0(R8)	Save VARWORK value
0014F2	4177	0001		00001	5199	LA R7,1(R7)	Reset after EXecute
					5200	* * /-----/	
					5201	* * /* Move VARWORK to VAR1 */	
					5202	* * /-----/	
0014F6	D2FF	D4E1	D6E4	004E1	006E4	5203 MVC VAL1,VARWORK	VAR1 w/o leading delimiter
0014FC	1A87					5204 AR R8,R7	Add VAR1 length to start addr
0014FE	0680					5205 BCTR R8,0	..to determine VAR1 end addr
					5206	* * /-----/	
					5207	* * /* Skip TRAILING delimiter character */	
					5208	* * /-----/	
				01500	5209	STRIPRR EQU *	
001500	D500	8000	D9EA	00000	009EA	5210 CLC 0(1,R8),DELIM	Strip character ?
001506	4770	B510		01510		5211 BNE STRIPRMM	No, update data
00150A	0680					5212 BCTR R8,0	Yes, bump to prev position
00150C	4670	B500		01500		5213 BCT R7,STRIPRR	Decrement length while DELIM
				01510	5214	STRIPRMM EQU *	
					5215	*	R7=VAR1 revised length
001510	5880	D284		00284		5216 L R8,VALP1	Reset to VAR1 value address
001514	D507	DB09	C5E0	00B09	025E0	5217 CLC FUNCT,=C'TRIM	TRIM Function ?
00151A	4780	B52A		0152A		5218 BE PSTRIPX	Yes, update VAR2 and exit
					5219	*	No, update VAR1 and exit
					5220	* * /-----/	
					5221	* * /* Force VAR1 update with VARWORK content */	
					5222	* * /-----/	
00151E	4120	D6E4		006E4		5223 LA R2,VARWORK	VARWORK area as VAR1
001522	5020	D294		00294		5224 ST R2,VALP2	
001526	9240	DB19		00B19		5225 MVI VAR2,C' '	Force VAR1 to be updated
				0152A	5226	PSTRIPX EQU *	
00152A	4550	BBC4		01BC4		5227 BAL R5,S#UVAR2	Update VAR1, no update to VAR2
00152E	47F0	B7E2		017E2		5228 B ERDONE	
					5229	*	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5231 *	*/*****	
					5232 *60	/* Function: STRIP\$ REQ OPT	
					5233 *	*/*****	
					5234 *	/* Strip all delimiter.	
					5235 *	/* Default delimiter is BLANK.	
					5236 *	/* - R2, R5 Working Register	
					5237 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					5238 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					5239 *	*/*****	
					5240 *	/* - RC=0 Successful Request	
					5241 *	*/*****	
			01532		5242 PSTRIP\$ EQU *	Process STRIP\$	
001532	4550	BA3E	01A3E		5243 BAL R5,S#GVAR1	Get VAR1	
001536	5070	D05C	0005C		5244 ST R7,R7HOLD	Save VAR1 value Length	
00153A	5080	D060	00060		5245 ST R8,R8HOLD	Save VAR1 value Address	
00153E	4550	BC4A	01C4A		5246 BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2	
					5247 *	/*-----	
					5248 *	/* VAR2 content area initial value is blanks	
					5249 *	/* regardless of S#GVAR2 return	
					5250 *	/*-----	
					5251 *	/* Link to IKJCT441 to obtain VAR2	
					5252 *	/*-----	
001542	4550	BB50	01B50		5253 BAL R5,S#GVAR2	Get VAR2	
					5254 *	/*-----	
					5255 *	/* Update VAR2 Length	
					5256 *	/*-----	
001546	4550	BD58	01D58		5257 BAL R5,S#UV2L	Update VAR2L variable	
					5258 *	/*-----	
					5259 *	/* Point to VAR2 value and length address	
					5260 *	/*-----	
00154A	5820	D294	00294		5261 L R2,VALP2	R2 Value Ptr retrieved/created	
00154E	5840	D290	00290		5262 L R4,VALL2	R4 Value Len retrieved/created	
					5263 *	/*-----	
					5264 *	/* Get 1-byte delimiter from VAR2	
					5265 *	/* BLANK assumed if not VAR2 specified	
					5266 *	/*-----	
001552	D200	D9EA 2000	009EA 00000		5267 MVC DELIM(1),0(R2)	Save 1 byte delimiter	
					5268 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5270	*	*/-----/	
					5271	*	/* Link to IKJCT441 to obtain VAR2	*/
					5272	*	*/-----/	
001558	4550	BB50	01B50		5273		BAL R5,S#GVAR2	Get VAR2
					5274	*	*/-----/	
					5275	*	/* Skip all delimiter in string (VAR1)	*/
					5276	*	*/-----/	
00155C	4120	D6E4	006E4		5277		LA R2,VARWORK	
001560	1B55				5278		SR R5,R5	VARWORK Length
				01562	5279	STRIPALL	EQU *	
001562	D500	8000	D9EA	00000	5280		CLC 0(1,R8),DELIM	Strip character ?
001568	4780	B57A	0157A		5281		BE SKIPDLM	Yes, bypass
00156C	D200	2000	8000	00000	5282		MVC 0(1,R2),0(R8)	No, keep it...
001572	4155	0001	00001		5283		LA R5,1(R5)	Add 1 to length - VARWORK
001576	4122	0001	00001		5284		LA R2,1(R2)	Bump to next position - VARWORK
				0157A	5285	SKIPDLM	EQU *	
00157A	4188	0001	00001		5286		LA R8,1(R8)	Bump to next position - VAR1
00157E	4670	B562	01562		5287		BCT R7,STRIPALL	YES, skip it and check next pos
					5288	*	*/-----/	
					5289	*	/* Update VAR1 w VARWORK value	*/
					5290	*	*/-----/	
				01582	5291	PSTRIP\$X	EQU *	
001582	1875				5292		LR R7,R5	Fill Length as VAR1L
001584	4120	D6E4	006E4		5293		LA R2,VARWORK	VARWORK area as VAR1
001588	5020	D294	00294		5294		ST R2,VALP2	
00158C	9240	DB19	00B19		5295		MVI VAR2,C' '	Force VAR1 to be updated
001590	4550	BBC4	01BC4		5296		BAL R5,S#UVAR2	Update VAR1, no update to VAR2
001594	47F0	B7E2	017E2		5297		B ERRDONE	
					5298	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5300 *	*/*****	
					5301 *61	/* Function: CONCAT REQ REQ	
					5302 *	*/*****	
					5303 *	/* Concatenate VAR01 and VAR02 into VAR01	
					5304 *	/* VAR01 max length is 256 bytes	
					5305 *	/* - R2, R4, R5, R6 Working Register	
					5306 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1)	
					5307 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1)	
					5308 *	*/*****	
					5309 *	/* - RC=0 Successful Request	
					5310 *	/* - RC=4015 VAR2 is required	
					5311 *	/* - RC=4016 VAR2 not found	
					5312 *	*/*****	
			01598		5313 PCONCAT EQU *	Process CONCATenate	
001598	4550	BA3E	01A3E		5314 BAL R5,S#GVAR1	Get VAR1	
00159C	5070	D05C	0005C		5315 ST R7,R7HOLD	Save VAR1 value Length	
0015A0	5080	D060	00060		5316 ST R8,R8HOLD	Save VAR1 value Address	
					5317 *	/-----/	
					5318 *	/* Need VAR2 content in position 1	
					5319 *	/-----/	
0015A4	9540	DB19	00B19		5320 CLI VAR2,C' '	VAR2 name captured?	
0015A8	4780	B896	01896		5321 BE ERR4015	NO, VAR2 required	
					5322 *	/-----/	
					5323 *	/* Link to IKJCT441 to obtain VAR2	
					5324 *	/-----/	
0015AC	4550	BB50	01B50		5325 BAL R5,S#GVAR2	Get VAR2	
0015B0	58F0	D04C	0004C		5326 L R15,RC#441	Restore R15	
0015B4	12FF				5327 LTR R15,R15	Successful IKJCT441?	
0015B6	4770	B8A4	018A4		5328 BNZ ERR4016	No, VAR2 not found	
					5329 *	/-----/	
					5330 *	/* Update VAR2 Length	
					5331 *	/-----/	
0015BA	4550	BD58	01D58		5332 BAL R5,S#UV2L	Update VAR2L variable	
					5333 *	/-----/	
					5334 *	/* Check VAR2 length limits VAR2 L = 1	
					5335 *	/-----/	
0015BE	5820	D294	00294		5336 L R2,VALP2	R2 Value Ptr retrieved/created	
0015C2	5840	D290	00290		5337 L R4,VALL2	R4 Value Len retrieved/created	
					5338 *	/-----/	
					5339 *	/* Move VAR1 to VARWORK	
					5340 *	/-----/	
0015C6	0670				5341 BCTR R7,0	Adjust for EXecute	
0015C8	4470	DCBA	00CBA		5342 EX R7,MVCVARWK	Execute VARWORK value move	
					5343 *MVCVARWK MVC VARWORK(0),0(R8)	Save VARWORK value	
0015CC	4177	0001	00001		5344 LA R7,1(R7)	Adjust length to original	
					5345 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5347	*	*/-----/	
					5348	*	/* MOVE VAR2 to VARWORK (concatenate)	*/
					5349	*	*/-----/	
0015D0	4160	D6E4	006E4		5350		LA R6,VARWORK	Point to VARWORK
0015D4	1A67				5351		AR R6,R7	Point to end of VAR1 in VARWORK
0015D6	1882				5352		LR R8,R2	R8=Start addr of VAL2
0015D8	1857				5353		LR R5,R7	Init R5 with VAR1 Length
0015DA	1A54				5354		AR R5,R4	Add VAR2 Length
					5355	*	*/-----/	
					5356	*	/* Force total length to 256 if > 256	*/
					5357	*	*/-----/	
0015DC	5950	C840	02840		5358		C R5,=F'256'	Exceed 256 limit?
0015E0	47D0	B5EE	015EE		5359		BNH PCONCATC	No, continue
0015E4	5B50	C840	02840		5360		S R5,=F'256'	Yes, calc remaining length
0015E8	1845				5361		LR R4,R5	
0015EA	4150	0100	00100		5362		LA R5,256	Yes, force total length to 256
			015EE		5363	PCONCATC	EQU *	
0015EE	0640				5364	BCTR	R4,0	Ajust for EXecute R4=VALL2 - 1
0015F0	4440	DCCC	00CCC		5365	EX	R4,MVCR8\$6	Move remainder of VAR1
					5366	*MVCR8\$6	MVC 0(0,R6),0(R8)	MVC R8 to R6
0015F4	4144	0001	00001		5367		LA R4,1(R4)	Adjust length to original R4=VALL2
					5368	*	*/-----/	
					5369	*	/* Update VAR1 w VARWORK value	*/
					5370	*	*/-----/	
			015F8		5371	PCONCATX	EQU *	
0015F8	1875				5372	LR	R7,R5	Fill Length as VAR1L
0015FA	4120	D6E4	006E4		5373	LA	R2,VARWORK	VARWORK area as VAR1
0015FE	5020	D294	00294		5374	ST	R2,VALP2	
001602	9240	DB19	00B19		5375	MVI	VAR2,C' '	Force VAR1 to be updated
001606	4550	BBC4	01BC4		5376	BAL	R5,S#UVAR2	Update VAR1, no update to VAR2
00160A	47F0	B7E2	017E2		5377	B	ERRDONE	
					5378	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5380 *	*/*****	
					5381 *62 *	/* Function: UNSTR REQ OPT */	
					5382 *	*/*****	
					5383 *	/* Unstring VAR1 into separate VAR1nnn variables */	
					5384 *	/* based on 1-byte delimiter, if specified in VAR2. */	
					5385 *	/* - If no VAR2 is specified, blank delimiter assumed. */	
					5386 *	/* - VAR10 contains number of variables created. */	
					5387 *	/* - VAR10,1,2,3,4,5,6,7,8,9,10,11 to 99 are possible */	
					5388 *	/* variable names. */	
					5389 *	*/	
					5390 *	/* VAR2 contains Search Params ----- */	
					5391 *	/* Format: c */	
					5392 *	/* where; */	
					5393 *	/* c - single byte delimiter (default=BLANK) */	
					5394 *	*/	
					5395 *	/* - R2, R4, R5 Working Register */	
					5396 *	/* - R7 Length of VAR1 value (after BAL S#GVAR1) */	
					5397 *	/* - R8 Address of VAR1 value (after BAL S#GVAR1) */	
					5398 *	*/*****	
					5399 *	/* - RC=0 Successful Request */	
					5400 *	*/*****	
			0160E		5401 PUNSTR EQU *	Process UNSTRing	
00160E	4550	BA3E	01A3E		5402 BAL R5,S#GVAR1	Get VAR1	
001612	5070	D05C	0005C		5403 ST R7,R7HOLD	Save VAR1 value Length	
001616	5080	D060	00060		5404 ST R8,R8HOLD	Save VAR1 value Address	
00161A	4550	BC4A	01C4A		5405 BAL R5,S#CHKV2	Reset VAR2 when \$NOVAR2	
					5406 *	*/-----/	
					5407 *	/* VAR2 content area initial value is blanks */	
					5408 *	/* regardless of S#GVAR2 return */	
					5409 *	*/-----/	
					5410 *	/* Link to IKJCT441 to obtain VAR2 */	
					5411 *	*/-----/	
00161E	4550	BB50	01B50		5412 BAL R5,S#GVAR2	Get VAR2	
					5413 *	*/-----/	
					5414 *	/* Update VAR2 Length */	
					5415 *	*/-----/	
001622	4550	BD58	01D58		5416 BAL R5,S#UV2L	Update VAR2L variable	
					5417 *	*/-----/	
					5418 *	/* Point to VAR2 value and length address */	
					5419 *	*/-----/	
001626	5820	D294	00294		5420 L R2,VALP2	R2 Value Ptr retrieved/created	
00162A	5840	D290	00290		5421 L R4,VALL2	R4 Value Len retrieved/created	
					5422 *	*/-----/	
					5423 *	/* Get 1-byte delimiter from VAR2 */	
					5424 *	/* BLANK assumed if not VAR2 specified */	
					5425 *	*/-----/	
00162E	D200	D9EA	2000	009EA	00000	5426 MVC DELIM(1),0(R2)	Save 1 byte delimiter
					5427 *	*/-----/	
					5428 *	/* Variable Counter to append VAR1 variable name */	
					5429 *	*/-----/	
001634	F810	D1D4	C899	001D4	02899	5430 ZAP NNN,=P'0'	
					5431 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20	
					5433	*	*/-----/		
					5434	*	/* Copy string until first delimiter is found	*/	
					5435	*	*/-----/		
				0163A	5436	CR8VARS	EQU *		
00163A	4120	D6E4	006E4		5437	LA	R2,VARWORK	Variable content from VAR1	
00163E	1B55				5438	SR	R5,R5	VARWORK Length	
				01640	5439	CHK4VAR	EQU *		
001640	D500	8000	D9EA	00000	009EA	5440	CLC	0(1,R8),DELIM	Delim character ?
001646	4780	B670		01670	5441	BE	CR8VCHK	Yes, check for create of VAR1nnn	
00164A	D200	2000	8000	00000	00000	5442	MVC	0(1,R2),0(R8)	No, keep it...
001650	4155	0001		00001	5443	LA	R5,1(R5)	Add 1 to length - VARWORK	
001654	4122	0001		00001	5444	LA	R2,1(R2)	Bump to next position - VARWORK	
001658	4188	0001		00001	5445	LA	R8,1(R8)	Bump to next position - VAR1	
00165C	4670	B640		01640	5446	BCT	R7,CHK4VAR	YES, skip it and check next pos	
					5447	*	*/-----/		
					5448	*	/* Primary loop complete. Check if last var needs	*/	
					5449	*	/* processing, then quit.	*/	
					5450	*	*/-----/		
001660	1255				5451	LTR	R5,R5	Length = 0?	
001662	4780	B690		01690	5452	BZ	PUNSTRX	Yes, done... no process necessary	
001666	1835				5453	LR	R3,R5	No, process last one	
001668	4550	C248		02248	5454	BAL	R5,S#CVARN	Create/Update VAR1nnn and VAR10	
00166C	47F0	B690		01690	5455	B	PUNSTRX	done...	
					5456	*	*/-----/		
					5457	*	/* Delimiter found, create varwork length for	*/	
					5458	*	/* processing and continue to scan for delimiter	*/	
					5459	*	*/-----/		
				01670	5460	CR8VCHK	EQU *	Delimiter found, check length	
					5461	*	R5 is length of varwork		
001670	1255				5462	LTR	R5,R5	Length > 0?	
001672	4770	B682		01682	5463	BNZ	CR8VAR	Yes, create VAR1nnn	
001676	4188	0001		00001	5464	LA	R8,1(R8)	No, Bump to next position - VAR1	
00167A	4670	B640		01640	5465	BCT	R7,CHK4VAR	check delim again...	
00167E	47F0	B690		01690	5466	B	PUNSTRX	End of Loop. Done..	
				01682	5467	CR8VAR	EQU *		
001682	1835				5468	LR	R3,R5	Length of VARWORK in RC for sub	
001684	4550	C248		02248	5469	BAL	R5,S#CVARN	Create/Update VAR1nnn and VAR10	
001688	4188	0001		00001	5470	LA	R8,1(R8)	Bump to next position - VAR1	
00168C	4670	B63A		0163A	5471	BCT	R7,CR8VARS	Reset and check next VAR01 character	
					5472	*	*/-----/		
					5473	*	/* No more VAR1 content	*/	
					5474	*	*/-----/		
				01690	5475	PUNSTRX	EQU *		
001690	47F0	B7E2		017E2	5476	B	ERRDONE		
					5477	*			

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				5479 *	*/*****	*/
				5480 *63	*/ * Function: REPLACE REQ REQ	*/
				5481 *	*/*****	*/
				5482 *	*/ * Replace substring in VAR1.	*/
				5483 *	*/ * VAR2 has from, to, start, end	*/
				5484 *	*/ * - result in VAR1	*/
				5485 *	*/ *	*/
				5486 *	*/ * VAR2 contains Search Params -----	*/
				5487 *	*/ * Format: "sssssss", "rrrrrrr", bbb, eee	*/
				5488 *	*/ * where;	*/
				5489 *	*/ * s - search string in double quotes	*/
				5490 *	*/ * r - replace string in double quotes	*/
				5491 *	*/ * b - optional, beginning search position	*/
				5492 *	*/ * 1-3 numeric digits, default=1	*/
				5493 *	*/ * e - optional, ending search position	*/
				5494 *	*/ * 1-3 numeric digits, default=end	*/
				5495 *	*/ * Params separated by commas (,):	*/
				5496 *	*/ *	*/
				5497 *	*/ * - R1, R2, R4, R5, R6 Working Register	*/
				5498 *	*/ * - R7 Length of VAR1 value (after BAL S#GVAR1)	*/
				5499 *	*/ * - R8 Address of VAR1 value (after BAL S#GVAR1)	*/
				5500 *	*/*****	*/
				5501 *	*/ * - RC=0 Successful Request	*/
				5502 *	*/ * - RC=4015 VAR2 is required	*/
				5503 *	*/ * - RC=4023 Result > 256 bytes	*/
				5504 *	*/ * - RC=4030 No begin " (search / replace string)	*/
				5505 *	*/ * - RC=4031 No end " (search / replace string)	*/
				5506 *	*/ * - RC=4032 Start > End position	*/
				5507 *	*/ * - RC=4033 Invalid character (delimiter)	*/
				5508 *	*/ * - RC=4034 Not numeric (start / end position)	*/
				5509 *	*/ * - RC=4035 Too many digits (start / end position)	*/
				5510 *	*/ * - RC=4036 Search or replace string is null	*/
				5511 *	*/*****	*/
		01694		5512 PREPL	EQU * Process REPLace	
001694	4550 BA3E	01A3E		5513	BAL R5,S#GVAR1 Get VAR1	
001698	5070 D05C	0005C		5514	ST R7,R7HOLD Save VAR1 value Length	
00169C	5080 D060	00060		5515	ST R8,R8HOLD Save VAR1 value Address	
				5516 *	*/-----	*/
				5517 *	*/ * Need VAR2 content in position 1	*/
				5518 *	*/-----	*/
0016A0	9540 DB19	00B19		5519	CLI VAR2,C' ' VAR2 name captured?	
0016A4	4780 B896	01896		5520	BE ERR4015 NO, VAR2 required	
				5521 *	*/-----	*/
				5522 *	*/ * Link to IKJCT441 to obtain VAR2	*/
				5523 *	*/-----	*/
0016A8	4550 BB50	01B50		5524	BAL R5,S#GVAR2 Get VAR2	
0016AC	58F0 D04C	0004C		5525	L R15,RC#441 Restore R15	
0016B0	12FF			5526	LTR R15,R15 Successful IKJCT441?	
0016B2	4770 B896	01896		5527	BNZ ERR4015 No, VAR2 required	
				5528 *		
0016B6	4550 BD58	01D58		5529	BAL R5,S#UV2L Update VAR2L variable	
				5530 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5532	*	*/-----/	
					5533	*	/* Check VAR2 length limits	VAR2 L = 1
					5534	*	*/-----/	
0016BA	5850	D294	00294		5535		L R5,VALP2	R2 Value Ptr retrieved/created
0016BE	5860	D290	00290		5536		L R6,VALL2	R4 Value Len retrieved/created
0016C2	5960	C7D8	027D8		5537		C R6,=F'0'	Variable length = 0?
0016C6	4780	B896	01896		5538		BE ERR4015	Yes, VAR2 required
					5539	*	*/-----/	
					5540	*	/* Get Replace Parameters	
					5541	*	*/-----/	
0016CA	4510	C386	02386		5542		BAL R1,S#SRCHP	Get Search Parms
0016CE	5820	D294	00294		5543		L R2,VALP2	VAR2 address
0016D2	5840	D290	00290		5544		L R4,VALL2	VAR2 length
					5545	*	*/-----/	
					5546	*	/* Check Search Start and End Values	
					5547	*	*/-----/	
0016D6	D503	D204	D208	00204	00208	5548	CLC SRCHS,SRCHE	Start > End location?
0016DC	4720	B930		01930		5549	BH ERR4032	Yes, Error Start > End
0016E0	D503	D204	D280	00204	00280	5550	CLC SRCHS,VALL1	Start > VAR1 Length?
0016E6	4720	B6EE		016EE		5551	BH REPL000	Yes, default Start to 1
0016EA	47F0	B6F4		016F4		5552	B REPL001	
				016EE		5553	REPL000 EQU *	
0016EE	D203	D204	C7F4	00204	027F4	5554	MVC SRCHS,=F'1'	
				016F4		5555	REPL001 EQU *	
0016F4	D503	D208	D280	00208	00280	5556	CLC SRCHE,VALL1	End > VAR1 Length?
0016FA	4720	B702		01702		5557	BH REPL002	Yes, default End to VAR1 Length
0016FE	47F0	B708		01708		5558	B REPL003	
				01702		5559	REPL002 EQU *	
001702	D203	D208	D280	00208	00280	5560	MVC SRCHE,VALL1	
				01708		5561	REPL003 EQU *	
						5562	*	*/-----/
						5563	*	/* Initialize for Search and Replace
						5564	*	*/-----/
						5565	*	/* R8- VAR1 R7- VAR1 Length
						5566	*	/* R3- VARWORK R5- VARWORK Length
001708	5810	D204		00204		5567	L R1,SRCHS	R1=Search Start
00170C	0610					5568	BCTR R1,0	...Adj for 0-base
00170E	4410	CF2E		02F2E		5569	EX R1,MVCVARW#	Move portion of VAR1 to VARWORK
						5570	*MVCVARW# MVC VARWORK(0),0(R8)	Save VARWORK value
001712	1B71					5571	SR R7,R1	Adj VAR1 Length by SRCHS
001714	1A81					5572	AR R8,R1	Adj start by SRCHS
001716	4130	D6E4		006E4		5573	LA R3,VARWORK	R3=Addr VARWORK
00171A	1A31					5574	AR R3,R1	Adj VARWORK addr by SRCHS
00171C	4150	0000		00000		5575	LA R5,0	R5=VARWORK Length
001720	1A51					5576	AR R5,R1	Adj VARWORK Length by SRCHS
001722	1828					5577	LR R2,R8	R2=VAR1
001724	5810	D1FC		001FC		5578	L R1,SRCHSTRL	R1=Length of Search String
001728	0610					5579	BCTR R1,0	...Adj for CLC
					5580	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5582 *	/* ----- */	
					5583 *	/* Search VARWORK for 'str' */	
					5584 *	r8 var1 r7 var1L	
					5585 *	r3 varwork r5 varworkL	
					5586 *	/* ----- */	
00172A	4410	DCD8	00CD8	0172A	5587 REPL	EQU * Look for 'str' in VAR1	
					5588	EX R1,SRCHCLC Substring found?	
					5589 *SRCHCLC	CLC 0(0,R2),SRCHSTR	
00172E	4780	B754	01754		5590	BE REPLF YES, let us replace!	
001732	D200	3000	2000	00000	5591	MVC 0(1,R3),0(R2) Move VAR1 to VARWORK	
001738	4133	0001	00001		5592	LA R3,1(R3) Bump up VARWORK	
00173C	4155	0001	00001		5593	LA R5,1(R5) Bump up VARWORK length	
001740	5950	C840	02840		5594	C R5,=F'256' > 256 Limit?	
001744	4720	B906	01906		5595	BH ERR4023 Yes, ERROR.	
001748	4122	0001	00001		5596	LA R2,1(R2) Bump up to next byte of VAR1	
00174C	4670	B72A	0172A		5597	BCT R7,REPL Check again...	
001750	47F0	B790	01790		5598	B PREPLX Done...	
					5599 *	/* ----- */	
					5600 *	/* Replace 'str' with 'st2' */	
					5601 *	/* ----- */	
				01754	5602 REPLF	EQU * R1=Addr of Search String 2	
001754	4110	D8E4	008E4		5603	LA R1,SRCHST2 R6=Length of Search String 2	
001758	5860	D200	00200		5604	L R6,SRCHST2L	
				0175C	5605 REPLFL	EQU * MOVE REPL String to VARWORK	
00175C	D200	3000	1000	00000	5606	MVC 0(1,R3),0(R1) Bump up VARWORK	
001762	4133	0001	00001		5607	LA R3,1(R3) Bump up VARWORK length	
001766	4155	0001	00001		5608	LA R5,1(R5) > 256 Limit?	
00176A	5950	C840	02840		5609	C R5,=F'256' Yes, ERROR.	
00176E	4720	B906	01906		5610	BH ERR4023 Bump to next REPL String char	
001772	4111	0001	00001		5611	LA R1,1(R1) Move again...	
001776	4660	B75C	0175C		5612	BCT R6,REPLFL End of loop	
					5613 *	R1=Length of Search String	
00177A	5810	D1FC	001FC		5614	L R1,SRCHSTRL ...Adj for CLC	
00177E	0610				5615	BCTR R1,0 Bump up SRCHSTRL bytes of VAR1	
001780	5A20	D1FC	001FC		5616	A R2,SRCHSTRL Reset length of VAR1	
001784	5B70	D1FC	001FC		5617	S R7,SRCHSTRL ...Adj R7 Length	
001788	4177	0001	00001		5618	LA R7,1(R7) Check again...	
00178C	4670	B72A	0172A		5619	BCT R7,REPL Done...	
					5620 *	/* ----- */	
					5621 *	/* Update VAR1 w VARWORK value */	
					5622 *	/* ----- */	
				01790	5623	PREPLX EQU * Fill Length as VAR1L	
001790	1875				5625	LR R7,R5 VARWORK area as VAR1	
001792	4120	D6E4	006E4		5626	LA R2,VARWORK	
001796	5020	D294	00294		5627	ST R2,VALP2	
00179A	9240	DB19	00B19		5628	MVI VAR2,C' ' Force VAR1 to be updated	
00179E	4550	BBC4	01BC4		5629	BAL R5,S#UVAR2 Update VAR1, no update to VAR2	
0017A2	47F0	B7E2	017E2		5630	B ERRDONE	
					5631 *	/* ----- */	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5633	*	*/*****	
					5634	*64	/* Function: VEXIST REQ NO	
					5635	*	*/*****	
					5636	*	/* Check if VAR1 variable exists.	
					5637	*	/* Result in RC; RC=4000 FALSE or content length, nnn	
					5638	*	/* No content returned in VAR1	
					5639	*	/* - R5 Working Register	
					5640	*	/* - R15 for RC	
					5641	*	*/*****	
					5642	*	/* - RC=nnn VAR1 content length	
					5643	*	/* - RC=4000 FALSE, VAR1 NOT found	
					5644	*	/* - RC=4022 VAR2 not needed	
					5645	*	*/*****	
			017A6		5646	PVEXIST	EQU * Process VEXIST	
					5647	*	/-----/	
					5648	*	/* VAR2 position 1 must be blank	
					5649	*	/-----/	
0017A6	D507	DB19	C7C0	00B19	027C0	5650	CLC VAR2,=C'\$NOVAR2 '	VAR2 none keyword?
0017AC	4780	B7B8		017B8		5651	BE PVEXIST#	YES, continue...
0017B0	9540	DB19		00B19		5652	CLI VAR2,C' '	VAR2 name captured?
0017B4	4770	B8F8		018F8		5653	BNE ERR4022	NO, VAR2 NOT required
					5654	*	/-----/	
					5655	*	/* Link to IKJCT441 to obtain VAR2 (via VAR1)	
					5656	*	/-----/	
				017B8	5657	PVEXIST#	EQU * Overlay VAR2 with VAR1	
0017B8	D207	DB19	DB11	00B19	00B11	5658	MVC VAR2,VAR1	Overlay VAR2 with VAR1
0017BE	D203	DB34	DB30	00B34	00B30	5659	MVC VAR2L,VAR1L	Overlay VAR2 with VAR1
0017C4	4550	BB4C		01B4C		5660	BAL R5,S#GVAR2N	Get VAR2 as VAR1, no NULL check
0017C8	58F0	D04C		0004C		5661	L R15,RC#441	Restore R15
0017CC	12FF					5662	LTR R15,R15	Successful IKJCT441?
0017CE	4770	B7DA		017DA		5663	BNZ VFINDNO	No, VAR1 not found
				017D2	5664	VFIND	EQU * Length of VAR1	
0017D2	58F0	D290		00290		5665	L R15,VALL2	Done...
0017D6	47F0	B7DE		017DE		5666	B PVEXISTX	
				017DA	5667	VFINDNO	EQU * FALSE, NOT FOUND	
0017DA	47F0	B7FC		017FC		5668	B ERR4000	
					5669	*	/-----/	
					5670	*	/* Done...	
					5671	*	/-----/	
				017DE	5672	PVEXISTX	EQU * ERRDONE	
0017DE	47F0	B7E2		017E2		5673	B	
					5674	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5676	*	*/*****	
					5677	*	/* Error Setting	
					5678	*	*/*****	
				017E2	5679	ERRDONE	EQU *	
0017E2	D212	D9EB	CAAC	009EB	02AAC	5680	MVC	ERRMSG(MSGDONEL),MSGDONE
0017E8	47F0	BA02		01A02	5681		B	ENDME
				017EC	5682	ERRDONEF	EQU *	
0017EC	D212	D9EB	CAAC	009EB	02AAC	5683	MVC	ERRMSG(MSGDONEL),MSGDONE
0017F2	D207	D9F6	DB09	009F6	00B09	5684	MVC	ERRMSG+11(8),FUNCT
0017F8	47F0	BA02		01A02	5685		B	ENDME
				017FC	5686	ERR4000	EQU *	
0017FC	41F0	0FA0		00FA0	5687		LA	R15,4000 FALSE
001800	D204	D9EB	CABF	009EB	02ABF	5688	MVC	ERRMSG(MSG4000L),MSG4000
001806	47F0	BA02		01A02	5689		B	ENDME
				0180A	5690	ERR4001	EQU *	
00180A	41F0	0FA1		00FA1	5691		LA	R15,4001 TRUE
00180E	D203	D9EB	CAC4	009EB	02AC4	5692	MVC	ERRMSG(MSG4001L),MSG4001
001814	47F0	BA02		01A02	5693		B	ENDME
				01818	5694	ERR4002	EQU *	
001818	41F0	0FA2		00FA2	5695		LA	R15,4002 VAR1 not found
00181C	D20D	D9EB	CAC8	009EB	02AC8	5696	MVC	ERRMSG(MSG4002L),MSG4002
001822	47F0	BA02		01A02	5697		B	ENDME
				01826	5698	ERR4004	EQU *	
001826	41F0	0FA4		00FA4	5699		LA	R15,4004 No PARM
00182A	D206	D9EB	CAD6	009EB	02AD6	5700	MVC	ERRMSG(MSG4004L),MSG4004
001830	47F0	BA02		01A02	5701		B	ENDME
				01834	5702	ERR4005	EQU *	
001834	41F0	0FA5		00FA5	5703		LA	R15,4005 WORD GT 8 bytes
001838	D20E	D9EB	CADD	009EB	02ADD	5704	MVC	ERRMSG(MSG4005L),MSG4005
00183E	47F0	BA02		01A02	5705		B	ENDME
				01842	5706	ERR4008	EQU *	
001842	41F0	0FA8		00FA8	5707		LA	R15,4008 CPPL no Parm
001846	D20B	D9EB	CAEC	009EB	02AEC	5708	MVC	ERRMSG(MSG4008L),MSG4008
00184C	47F0	BA02		01A02	5709		B	ENDME
				01850	5710	ERR4009	EQU *	
001850	41F0	0FA9		00FA9	5711		LA	R15,4009 Too many WORDS
001854	D20D	D9EB	CAF8	009EB	02AF8	5712	MVC	ERRMSG(MSG4009L),MSG4009
00185A	47F0	BA02		01A02	5713		B	ENDME

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				0185E	5715	ERR4010	EQU *	
00185E	41F0	0FAA	00FAA		5716		LA R15,4010	SETVAR no content
001862	D210	D9EB	CB06	009EB	5717		MVC ERRMSG(MSG4010L),MSG4010	
001868	47F0	BA02		01A02	5718		B ENDME	
				0186C	5719	ERR4011	EQU *	
00186C	41F0	0FAB	00FAB		5720		LA R15,4011	Invalid Function
001870	D20F	D9EB	CB17	009EB	5721		MVC ERRMSG(MSG4011L),MSG4011	
001876	47F0	BA02		01A02	5722		B ENDME	
				0187A	5723	ERR4012	EQU *	
00187A	41F0	0FAC	00FAC		5724		LA R15,4012	VAR2 not updated
00187E	D20F	D9EB	CB27	009EB	5725		MVC ERRMSG(MSG4012L),MSG4012	
001884	47F0	BA02		01A02	5726		B ENDME	
				01888	5727	ERR4014	EQU *	
001888	41F0	0FAE	00FAE		5728		LA R15,4014	VAR1 GT 256
00188C	D210	D9EB	CB37	009EB	5729		MVC ERRMSG(MSG4014L),MSG4014	
001892	47F0	BA02		01A02	5730		B ENDME	
				01896	5731	ERR4015	EQU *	
001896	41F0	0FAF	00FAF		5732		LA R15,4015	VAR2 required
00189A	D20C	D9EB	CB48	009EB	5733		MVC ERRMSG(MSG4015L),MSG4015	
0018A0	47F0	BA02		01A02	5734		B ENDME	
				018A4	5735	ERR4016	EQU *	
0018A4	41F0	0FB0	00FB0		5736		LA R15,4016	VAR2 not found
0018A8	D20D	D9EB	CB55	009EB	5737		MVC ERRMSG(MSG4016L),MSG4016	
0018AE	47F0	BA02		01A02	5738		B ENDME	
				018B2	5739	ERR4017	EQU *	
0018B2	41F0	0FB1	00FB1		5740		LA R15,4017	VAR2 GT 256
0018B6	D210	D9EB	CB63	009EB	5741		MVC ERRMSG(MSG4017L),MSG4017	
0018BC	47F0	BA02		01A02	5742		B ENDME	
				018C0	5743	ERR4018	EQU *	
0018C0	41F0	0FB2	00FB2		5744		LA R15,4018	VAR2 is null
0018C4	D208	D9EB	CB74	009EB	5745		MVC ERRMSG(MSG4018L),MSG4018	
0018CA	47F0	BA02		01A02	5746		B ENDME	
				018CE	5747	ERR4019	EQU *	
0018CE	41F0	0FB3	00FB3		5748		LA R15,4019	VAR1 is null
0018D2	D208	D9EB	CB7D	009EB	5749		MVC ERRMSG(MSG4019L),MSG4019	
0018D8	47F0	BA02		01A02	5750		B ENDME	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				018DC	5752	ERR4020	EQU *	
0018DC	41F0	0FB4	00FB4		5753		LA R15,4020	ERRMSG cannot create updt
0018E0	D218	D9EB	CB86	009EB	02B86	5754	MVC	ERRMSG(MSG4020L),MSG4020
0018E6	47F0	BA02		01A02	5755		B	ENDME
				018EA	5756	ERR4021	EQU *	
0018EA	41F0	0FB5	00FB5		5757		LA R15,4021	VAR1 required
0018EE	D20C	D9EB	CB9F	009EB	02B9F	5758	MVC	ERRMSG(MSG4021L),MSG4021
0018F4	47F0	BA02		01A02	5759		B	ENDME
				018F8	5760	ERR4022	EQU *	
0018F8	41F0	0FB6	00FB6		5761		LA R15,4022	VAR2 not required
0018FC	D210	D9EB	CBAC	009EB	02BAC	5762	MVC	ERRMSG(MSG4022L),MSG4022
001902	47F0	BA02		01A02	5763		B	ENDME
				01906	5764	ERR4023	EQU *	
001906	41F0	0FB7	00FB7		5765		LA R15,4023	Result GT 256
00190A	D20C	D9EB	CBBD	009EB	02BBD	5766	MVC	ERRMSG(MSG4023L),MSG4023
001910	47F0	BA02		01A02	5767		B	ENDME

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				01914	5769	ERR4030	EQU *	
001914	41F0	0FBE	00FBE		5770		LA R15,4030	No begin "
001918	D209	D9EB	CBCA	009EB	5771		MVC ERRMSG(MSG4030L),MSG4030	
00191E	47F0	BA02	01A02		5772		B ENDME	
				01922	5773	ERR4031	EQU *	
001922	41F0	0FBF	00FBF		5774		LA R15,4031	No end "
001926	D207	D9EB	CBD4	009EB	5775		MVC ERRMSG(MSG4031L),MSG4031	
00192C	47F0	BA02	01A02		5776		B ENDME	
				01930	5777	ERR4032	EQU *	
001930	41F0	0FC0	00FC0		5778		LA R15,4032	Start GT End
001934	D20B	D9EB	CBDC	009EB	5779		MVC ERRMSG(MSG4032L),MSG4032	
00193A	47F0	BA02	01A02		5780		B ENDME	
				0193E	5781	ERR4033	EQU *	
00193E	41F0	0FC1	00FC1		5782		LA R15,4033	Invalid char
001942	D20B	D9EB	CBE8	009EB	5783		MVC ERRMSG(MSG4033L),MSG4033	
001948	47F0	BA02	01A02		5784		B ENDME	
				0194C	5785	ERR4034	EQU *	
00194C	41F0	0FC2	00FC2		5786		LA R15,4034	Not numeric
001950	D20A	D9EB	CBF4	009EB	5787		MVC ERRMSG(MSG4034L),MSG4034	
001956	47F0	BA02	01A02		5788		B ENDME	
				0195A	5789	ERR4035	EQU *	
00195A	41F0	0FC3	00FC3		5790		LA R15,4035	Too many digits
00195E	D20E	D9EB	CBFF	009EB	5791		MVC ERRMSG(MSG4035L),MSG4035	
001964	47F0	BA02	01A02		5792		B ENDME	
				01968	5793	ERR4036	EQU *	
001968	41F0	0FC4	00FC4		5794		LA R15,4036	Null value not allowed
00196C	D215	D9EB	CC0E	009EB	5795		MVC ERRMSG(MSG4036L),MSG4036	
001972	47F0	BA02	01A02		5796		B ENDME	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	16.49	04/12/20
				01976	5798	ERR4037	EQU	*			
001976	41F0	0FC5		00FC5	5799		LA	R15,4037	No begin QUOTE		
00197A	D20D	D9EB	CC24	009EB	02C24	5800	MVC	ERRMSG(MSG4037L),MSG4037			
001980	47F0	BA02		01A02	5801		B	ENDME			
				01984	5802	ERR4038	EQU	*			
001984	41F0	0FC6		00FC6	5803		LA	R15,4038	No end QUOTE		
001988	D20B	D9EB	CC32	009EB	02C32	5804	MVC	ERRMSG(MSG4038L),MSG4038			
00198E	47F0	BA02		01A02	5805		B	ENDME			
				01992	5806	ERR4039	EQU	*			
001992	41F0	0FC7		00FC7	5807		LA	R15,4039	No end QUOTE found		
001996	D211	D9EB	CC3E	009EB	02C3E	5808	MVC	ERRMSG(MSG4039L),MSG4039			
00199C	47F0	BA02		01A02	5809		B	ENDME			

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				019A0	5811	ERR4040	EQU *	
0019A0	41F0	0FC8	00FC8		5812		LA R15,4040	Invalid length
0019A4	D20D	D9EB	CC50	009EB	02C50	5813	MVC	ERRMSG(MSG4040L),MSG4040
0019AA	47F0	BA02		01A02	5814		B	ENDME
				019AE	5815	ERR4047	EQU *	
0019AE	41F0	0FCF	00FCF		5816		LA R15,4047	Invalid DD
0019B2	D209	D9EB	CC5E	009EB	02C5E	5817	MVC	ERRMSG(MSG4047L),MSG4047
0019B8	47F0	BA02		01A02	5818		B	ENDME
				019BC	5819	ERR4049	EQU *	
0019BC	41F0	0FD1	00FD1		5820		LA R15,4049	Invalid MM
0019C0	D209	D9EB	CC68	009EB	02C68	5821	MVC	ERRMSG(MSG4049L),MSG4049
0019C6	47F0	BA02		01A02	5822		B	ENDME

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				019CA	5824	ERR4050	EQU *	
0019CA	41F0	0FD2	00FD2		5825		LA R15,4050	Invalid DD for MM
0019CE	D210	D9EB	CC72	009EB	5826		MVC ERRMSG(MSG4050L),MSG4050	
0019D4	47F0	BA02	01A02		5827		B ENDME	
				019D8	5828	ERR4051	EQU *	
0019D8	41F0	0FD3	00FD3		5829		LA R15,4051	Invalid JJJ
0019DC	D20A	D9EB	CC83	009EB	5830		MVC ERRMSG(MSG4051L),MSG4051	
0019E2	47F0	BA02	01A02		5831		B ENDME	
				019E6	5832	ERR4098	EQU *	
0019E6	58F0	C844	02844		5833		L R15,=F'4098'	Cannot link to IKJCT441
0019EA	D216	D9EB	CC8E	009EB	5834		MVC ERRMSG(MSG4098L),MSG4098	
0019F0	47F0	BA02	01A02		5835		B ENDME	
				019F4	5836	ERR4099	EQU *	
0019F4	58F0	C848	02848		5837		L R15,=F'4099'	Must run under TSO
0019F8	D211	D9EB	CCA5	009EB	5838		MVC ERRMSG(MSG4099L),MSG4099	
0019FE	47F0	BA02	01A02		5839		B ENDME	
					5840	*	*	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				5842 *	*/*****	
				5843 *	/* Exit, return to caller	*/
				5844 *	/* Update ERRMSG, VAR1L, VAR2L, FREEMAIN, BR14	*/
				5845 *	*/*****	
001A02				5846 ENDME	DS 0H	Return....
001A02	50F0 D048	00048		5847	ST R15,SAVER15	SAVE R15, RC for exit
001A06	4550 BC62	01C62		5848	BAL R5,S#UERRM	Update ERRMSG variable
001A0A	4550 BCC8	01CC8		5849	BAL R5,S#UV1L	Update VAR1L variable
001A0E	4550 BD58	01D58		5850	BAL R5,S#UV2L	Update VAR2L variable
		01A12		5851 MYEXIT	EQU *	
				5852 *	*/*****	
				5853 *	/* - Restore savearea, FREEMAIN, and	*/
				5854 *	/* return to caller	*/
				5855 *	*/*****	
001A12	181D			5856	LR R1,R13	Working Storage Addr w Savearea
001A14	5850 D048	00048		5857	L R5,SAVER15	Load R5 with R15 return code
001A18	58DD 0004	00004		5858	L R13,4(R13)	Restore R13 w callers savearea
				5859 *	/* -----	*/
				5860 *	/* Working Storage FREEMAIN	*/
				5861 *	/* -----	*/
				5862	PRINT GEN	
				5863	FREEMAIN R, LV=WSAREAL, A=(R1) Free Storage	
				5864+*	OS/VS2 RELEASE 3 VERSION -- 10/25/74	00001603
001A1C				5865+	CNOP 0,4	00144002
001A1C	47F0 BA24	01A24		5866+	B *+8	BRANCH AROUND LENGTH 00145002
001A20	00000CE8			5867+	DC A(WSAREAL)	LENGTH 00147802
001A24	5800 BA20	01A20		5868+	L 0,*-4	LOAD SP AND LV 00148002
001A28	4110 1000	00000		5869+	LA 1,0(0,R1)	LOAD AREA ADDRESS 00164002
001A2C	0A0A			5870+	SVC 10	ISSUE FREEMAIN SVC 00311202
				5871	PRINT NOGEN	
				5872 *	/* -----	*/
				5873 *	/* Restore caller registers and return to caller	*/
				5874 *	/* -----	*/
001A2E	18F5			5875	LR R15,R5	R15 = RC for exit
		01A30		5876 NOMAINS	EQU *	
		01A30		5877 RETURN	EQU *	
001A30	58ED 000C	0000C		5878	L R14,12(R13)	Restore R14
001A34	980C D014	00014		5879	LM R0,R12,20(R13)	Restore R0 thru R12
001A38	07FE			5880	BR R14	Return to caller
				5881 *	/*	

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
5883 *      * /*****  
5884 *      * /* Subroutine - Get Variable 1 (R5) */  
5885 *      * /* Get Variable 1 (Source) */  
5886 *      * /* - R7 Length of VAR1 symbolic (after return) */  
5887 *      * /* - R8 Address of VAR1 symbolic (after return) */  
5888 *      * /* - R0, R1, R2, R6, R15 Working Register */  
5889 *      * /*****  
5890 *      * /* -,VALL1 VALUE Len retrieved/created */  
5891 *      * /* - VALP1 VALUE Ptr retrieved/created */  
5892 *      * /* - VAL1 VALUE content retrieved/created */  
5893 *      * /*****  
5894 *      * /* Return codes for IKJCT441 */  
5895 *      * /*  
5896 *      * /* 0 - SUCCESS */  
5897 *      * /* 4 - VARIABLE RETURNED SHOULDN'T BE RE-SCANNED */  
5898 *      * /* 8 - VARIABLE RETURNED REQUIRES EVALUATION */  
5899 *      * /* 12 - VARIABLE RETURNED IS A LABEL (CANNOT UPDATE) */  
5900 *      * /* 16 - SYSTEM VARIABLE - CAN'T BE UPDATED BY THE USER */  
5901 *      * /* 20 - FOR LOCATE - NO MORE VARIABLES TO RETURN */  
5902 *      * /* 32 - GETMAIN/FREEMAIN FAILURE */  
5903 *      * /* 36 - NAME LENGTH < 1 OR > 252 */  
5904 *      * /* -OR- VALUE LENGTH < 0 OR > 32767 */  
5905 *      * /* 40 - INCORRECT ENVIRONMENT - */  
5906 *      * /* PLIST ERROR OR NO CLIST ACTIVE */  
5907 *      * /* 44 - INVALID ENTRY CODE */  
5908 *      * /* 52 - UNDEFINED VARIABLE */  
5909 *      * /*  
5910 *      * /*****
```

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				01A3A	5912	S#GVAR1N	EQU *	
001A3A	92D5	DB40	00B40		5913	MVI	CHKV1NL,C'N'	Bypass VAR1 NULL test
				01A3E	5914	S#GVAR1	EQU *	
001A3E	5050	D070	00070		5915	ST	R5,SAVER5	
					5916	*	*/-----/	
					5917	*	/* Bypass VAR1 fetch	*/
					5918	*	*/-----/	
001A42	9540	DB11	00B11		5919	CLI	VAR1,C' '	Blank VAR1 name in position 1?
001A46	4780	B8EA	018EA		5920	BE	ERR4021	No, VAR1 missing
					5921	*	*/-----/	
					5922	*	/* Get provided VAR1 name	*/
					5923	*	*/-----/	
001A4A	D207	D4D0	DB11	004D0	00B11	5924	MVC	NAM1,VAR1 Init VARIABLE NAME
001A50	D203	D278	DB30	00278	00B30	5925	MVC	NAML1,VAR1L Init VARIABLE NAME LENGTH
					5926	*	*/-----/	
					5927	*	/* Initialize IKJCT441 Parm List area	*/
					5928	*	*/-----/	
001A56	4100	CAA4	02AA4		5929	LA	R0,ECNOIMP	Entry Code
001A5A	5000	D260	00260		5930	ST	R0,CT441P1	
001A5E	4100	D27C	0027C		5931	LA	R0,NAMP1	Name Pointer
001A62	5000	D264	00264		5932	ST	R0,CT441P2	NAMP1
001A66	4100	D278	00278		5933	LA	R0,NAML1	Name Length
001A6A	5000	D268	00268		5934	ST	R0,CT441P3	NAML1
001A6E	4100	D284	00284		5935	LA	R0,VALP1	Var Pointer
001A72	5000	D26C	0026C		5936	ST	R0,CT441P4	VALP1
001A76	4100	D280	00280		5937	LA	R0,VALL1	Var Length
001A7A	5000	D270	00270		5938	ST	R0,CT441P5	VALL1
001A7E	4100	D1EC	001EC		5939	LA	R0,TOKEN	Token
001A82	5000	D274	00274		5940	ST	R0,CT441P6	
001A86	9280	D274	00274		5941	MVI	CT441P6,B'10000000'	Last Parm designation
					5942	*	*/-----/	
					5943	*	/* Link to IKJCT441	*/
					5944	*	*/-----/	
001A8A	4110	D260	00260		5945	LA	R1,CT441LST	R1 = Parm List
001A8E	58F0	D25C	0025C		5946	L	R15,IKJCT441	
001A92	05EF				5947	BALR	R14,R15	
					5948	*		
001A94	50F0	D04C	0004C		5949	ST	R15,RC#441	Save R15 from IKJCT441
001A98	12FF				5950	LTR	R15,R15	Successful invokation?
001A9A	4770	B818	01818		5951	BNZ	ERR4002	No, VAR1 not found
					5952	*		Yes.
001A9E	5870	D280	00280		5953	L	R7,VALL1	R7 Value Len retrieved/created
001AA2	5880	D284	00284		5954	L	R8,VALP1	R8 Value Ptr retrieved/created
					5955	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					5957	*	*/-----/	
					5958	*	/* Check length of variable limits	*/
					5959	*	*/-----/	
001AA6	5970	C840	02840		5960	C	R7,=F'256'	Variable length > 256?
001AAA	4720	B888	01888		5961	BH	ERR4014	Cannot process. 256 limit.
001AAE	95D5	DB40	00B40		5962	CLI	CHKV1NL,C'N'	Check VAR1 for NULL?
001AB2	4780	BABE	01ABE		5963	BE	BYPSV1NL	No, bypass check
					5964	*		Yes, check for NULL
001AB6	5970	C7D8	027D8		5965	C	R7,=F'0'	Variable length = 0?
001ABA	4780	B8CE	018CE		5966	BE	ERR4019	Yes, null value. Exit.
			01ABE		5967	BYPSV1NL	EQU *	
					5968	*	*/-----/	
					5969	*	/* Copy value of CLIST value	*/
					5970	*	*/-----/	
001ABE	4470	DCC0	00CC0		5971	EX	R7,MVCVAR1	Execute VAR1 value move
					5972	*MVCVAR1	MVC VAL1(0),0(R8)	Save VAR1 value
					5973	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					5975 *	*/*****	*/
					5976 *	/* QUOTE, rebuild VAR1 based on QUOTE content	*/
					5977 *	/* - R6 Working Register	*/
					5978 *	*/*****	*/
001AC2	D507	DB21	C7C8	00B21	027C8	5979 CLC	SQUOTE,=C'QUOTE ' QUOTE Processing?
001AC8	4770	BB46		01B46		5980 BNE	S#GVAR1X No, branch to continue
001ACC	5080	D060		00060		5981 ST	R8,R8HOLD Yes, take content between quotes
001AD0	5070	D05C		0005C		5982 ST	R7,R7HOLD
					5983 *	VAR1 = 'I am in set of quotes '	
					5984 *	^-----^	
					5985 *	0-----1-----2-----3-----4	
					5986 *	1---+---0---+---0---+---0---+---0	
					5987 *	R7 = 30 (length) FW R8 = 5003 (address) FW	
001AD4	957D	8000		00000		5988 CLI	0(R8),C' ' ' VAR1 1st byte a QUOTE?
001AD8	4770	B976		01976		5989 BNE	ERR4037 NO, Error need quote at start!
001ADC	1A87					5990 AR	R8,R7 YES, Point to end of
001ADE	0680					5991 BCTR	R8,0 ... VAR1 and check/find end quote
001AE0	0670					5992 BCTR	R7,0 Adjust length of VAR1
				01AE2	5993 LOOKBBB	EQU *	Look for end quote ...
001AE2	9540	8000		00000		5994 CLI	0(R8),C' ' ' VAR1 LAST byte a BLANK?
001AE6	4770	BAF4		01AF4		5995 BNE	ENDVAR1 NO, Check for end QUOTE in VAR1
001AEA	0680					5996 BCTR	R8,0 YES, Move back one byte in VAR1
001AEC	4670	BAE2		01AE2		5997 BCT	R7,LOOKBBB ... and Look again.
001AF0	47F0	B992		01992		5998 B	ERR4039 Did not find ending quote
				01AF4	5999 ENDVAR1	EQU *	
001AF4	957D	8000		00000		6000 CLI	0(R8),C' ' ' ' Is LAST byte a quote?
001AF8	4770	B984		01984		6001 BNE	ERR4038 NO, Error need quote at end!
001AFC	0670					6002 BCTR	R7,0 R7=Revised VAR1 Length minus QUOTE
				01AFE	6003 MOVQTVAR	EQU *	
001AFE	4160	D6E4		006E4		6004 LA	R6,VARWORK R6=Addr of VARWORK
001B02	5880	D060		00060		6005 L	R8,R8HOLD Restore R8, start of VAR1 content
001B06	4188	0001		00001		6006 LA	R8,1(R8) Adjust R8 for begining QUOTE
001B0A	4470	DCCC		00CCC		6007 EX	R7,MVCR8\$6 Move VAR1 content to VARWORK
					6008 *MVCR8\$6	MVC 0(0,R6),0(R8)	
001B0E	4160	D4E1		004E1		6009 LA	R6,VAL1 R6=Addr of VAL1 (VAR1 content)
001B12	4180	D6E4		006E4		6010 LA	R8,VARWORK R8=Addr of VARWORK
001B16	4470	DCCC		00CCC		6011 EX	R7,MVCR8\$6 Move VARWORD to VAR1 (overlay)
					6012 *MVCR8\$6	MVC 0(0,R6),0(R8)	
001B1A	5880	D060		00060		6013 L	R8,R8HOLD Restore start of VAR1
001B1E	4188	0001		00001		6014 LA	R8,1(R8) Adjust R8 for begining QUOTE
					6015 *		
					6016 *		
001B22	5070	D280		00280		6017 ST	R7,VALL1 New VAR1 length and start addr
001B26	5080	D284		00284		6018 ST	R8,VALP1 Store adjusted length for VAR1
001B2A	9240	D6E4		006E4		6019 MVI	VARWORK,C' ' ' Clear VARWORK
001B2E	D2FF	D6E5	D6E4	006E5	006E4	6020 MVC	VARWORK+1(L'VARWORK),VARWORK
001B34	D507	DB19	C7C0	00B19	027C0	6021 CLC	VAR2,=C'\$NOVAR2 ' NOVAR2 keyword?
001B3A	4770	BB46		01B46		6022 BNE	S#GVAR1X No, use VAR2 for result
001B3E	9240	DB19		00B19		6023 MVI	VAR2,C' ' ' Yes, use VAR1 for result
001B42	47F0	BB46		01B46		6024 B	S#GVAR1X
				01B46	6025 S#GVAR1X	EQU *	
001B46	5850	D070		00070		6026 L	R5,SAVER5
001B4A	07F5					6027 BR	R5 Return to caller (R5)
					6028 *		

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

```
6030 *      * /*****
6031 *      * /* Subroutine - Get Variable 2 (R5) */
6032 *      * /* Get Variable 2 (Source) */
6033 *      * /* - R0, R1 Working Register */
6034 *      * /*****
6035 *      * /* - VALL2 VALUE Len retrieved/created */
6036 *      * /* - VALP2 VALUE Ptr retrieved/created */
6037 *      * /*****
6038 *      * /* Return codes for IKJCT441 */
6039 *      * /*
6040 *      * /* 0 - SUCCESS */
6041 *      * /* 4 - VARIABLE RETURNED SHOULDN'T BE RE-SCANNED */
6042 *      * /* 8 - VARIABLE RETURNED REQUIRES EVALUATION */
6043 *      * /* 12 - VARIABLE RETURNED IS A LABEL (CANNOT UPDATE) */
6044 *      * /* 16 - SYSTEM VARIABLE - CAN'T BE UPDATED BY THE USER */
6045 *      * /* 20 - FOR LOCATE - NO MORE VARIABLES TO RETURN */
6046 *      * /* 32 - GETMAIN/FREEMAIN FAILURE */
6047 *      * /* 36 - NAME LENGTH < 1 OR > 252 */
6048 *      * /* -OR- VALUE LENGTH < 0 OR > 32767 */
6049 *      * /* 40 - INCORRECT ENVIRONMENT - */
6050 *      * /* PLIST ERROR OR NO CLIST ACTIVE */
6051 *      * /* 44 - INVALID ENTRY CODE */
6052 *      * /* 52 - UNDEFINED VARIABLE */
6053 *      * /*
6054 *      * /*****
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
			01B4C	6056	S#GVAR2N EQU *	
001B4C	92D5 DB41	00B41		6057	MVI CHKV2NL,C'N'	Bypass VAR2 NULL test
			01B50	6058	S#GVAR2 EQU *	
001B50	5050 D070	00070		6059	ST R5,SAVER5	
001B54	D207 D4D8 DB19	004D8 00B19		6060	MVC NAM2,VAR2	Init VARIABLE NAME
001B5A	D203 D288 DB34	00288 00B34		6061	MVC NAML2,VAR2L	Init VARIABLE NAME LENGTH
				6062	* /* *****	*/
				6063	* /* Initialize IKJCT441 Parm List area	*/
				6064	* /* - R0 Working Register	*/
				6065	* /* - R15 RC from IKJCT441	*/
				6066	* /* *****	*/
001B60	4100 CAA4	02AA4		6067	LA R0,ECNOIMP	Entry Code
001B64	5000 D260	00260		6068	ST R0,CT441P1	
001B68	4100 D28C	0028C		6069	LA R0,NAMP2	Name Pointer
001B6C	5000 D264	00264		6070	ST R0,CT441P2	
001B70	4100 D288	00288		6071	LA R0,NAML2	Name Length
001B74	5000 D268	00268		6072	ST R0,CT441P3	
001B78	4100 D294	00294		6073	LA R0,VALP2	Var Pointer
001B7C	5000 D26C	0026C		6074	ST R0,CT441P4	
001B80	4100 D290	00290		6075	LA R0,VALL2	Var Length
001B84	5000 D270	00270		6076	ST R0,CT441P5	
001B88	4100 D1EC	001EC		6077	LA R0,TOKEN	Token
001B8C	5000 D274	00274		6078	ST R0,CT441P6	
001B90	9280 D274	00274		6079	MVI CT441P6,B'10000000'	Last Parm designation
				6080	* /* -----	*/
				6081	* /* Link to IKJCT441	*/
				6082	* /* -----	*/
001B94	4110 D260	00260		6083	LA R1,CT441LST	R1 = Parm List
001B98	58F0 D25C	0025C		6084	L R15,IKJCT441	
001B9C	05EF			6085	BALR R14,R15	
				6086	* /*	
001B9E	50F0 D04C	0004C		6087	ST R15,RC#441	Save R15 from IKJCT441
				6088	* /* -----	*/
				6089	* /* Check length of variable limits	*/
				6090	* /* -----	*/
001BA2	D503 D290 C840	00290 02840		6091	CLC VALL2,=F'256'	Variable length > 256?
001BA8	4720 B8B2	018B2		6092	BH ERR4017	Cannot process. 256 limit.
001BAC	95D5 DB41	00B41		6093	CLI CHKV2NL,C'N'	Check VAR1 for NULL?
001BB0	4780 BBBE	01BBE		6094	BE BYPSV2NL	No, bypass check
				6095	* Yes, check for NULL	
001BB4	D503 D290 C7D8	00290 027D8		6096	CLC VALL2,=F'0'	Variable length = 0?
001BBA	4780 B8C0	018C0		6097	BE ERR4018	Yes, null value. Exit.
		01BBE		6098	BYPSV2NL EQU *	
		01BBE		6099	S#GVAR2X EQU *	
001BBE	5850 D070	00070		6100	L R5,SAVER5	
001BC2	07F5			6101	BR R5	Return to caller (R5)
				6102	* /*	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6104	*	/* ***** */	
					6105	*	/* Subroutine - Update/Create Variable 2 (Result)(R5) */	
					6106	*	/*	
					6107	*	/* If no VAR2 specified, overlay VAR1 with result */	
					6108	*	/* If VAR2 specified, create/update VAR2 with result */	
					6109	*	/* - R7 Length of VAR1 symbolic */	
					6110	*	/* - R0, R1 Working Register */	
					6111	*	/* ***** */	
			01BC4		6112	S#UVAR2	EQU *	
001BC4	5050	D070	00070		6113		ST R5,SAVER5	
001BC8	D2FF	D5E3	D6E4	005E3	006E4	6114	MVC VAL2,VARWORK	Copy VARWORK to VAL2
001BCE	5070	D290		00290	6115		ST R7,VALL2	Init VARIABLE DATA LENGTH
001BD2	9540	DB19		00B19	6116		CLI VAR2,C' '	VAR2 symbolic name blank?
001BD6	4780	BBEA		01BEA	6117		BE USEVAR1	YES, Update VAR1 symbolic
				01BDA	6118	USEVAR2	EQU *	NO, Update VAR2 symbolic
001BDA	D207	D4D8	DB19	004D8	00B19	6119	MVC NAM2,VAR2	Init VARIABLE NAME
001BE0	D203	D288	DB34	00288	00B34	6120	MVC NAML2,VAR2L	Init VARIABLE NAME LENGTH
001BE6	47F0	BBFC		01BFC	6121		B USECONT	
				01BEA	6122	USEVAR1	EQU *	Update VAR2 symbolic
001BEA	D207	D4D8	DB11	004D8	00B11	6123	MVC NAM2,VAR1	Init VARIABLE NAME
001BF0	D203	D288	DB30	00288	00B30	6124	MVC NAML2,VAR1L	Init VARIABLE NAME LENGTH
001BF6	D203	D280	D290	00280	00290	6125	MVC VALL1,VALL2	Update VAR01 data length
				01BFC	6126	USECONT	EQU *	
					6127	*	/* ----- */	
					6128	*	/* Initialize IKJCT441 Parm List area */	
					6129	*	/* ----- */	
001BFC	4100	CAA8		02AA8	6130		LA R0,ECUPDT	Entry Code
001C00	5000	D260		00260	6131		ST R0,CT441P1	
001C04	4100	D28C		0028C	6132		LA R0,NAMP2	Name Pointer
001C08	5000	D264		00264	6133		ST R0,CT441P2	
001C0C	4100	D288		00288	6134		LA R0,NAML2	Name Length
001C10	5000	D268		00268	6135		ST R0,CT441P3	
001C14	4100	D294		00294	6136		LA R0,VALP2	Var Pointer
001C18	5000	D26C		0026C	6137		ST R0,CT441P4	
001C1C	4100	D290		00290	6138		LA R0,VALL2	Var Length
001C20	5000	D270		00270	6139		ST R0,CT441P5	
001C24	4100	D1EC		001EC	6140		LA R0,TOKEN	Token
001C28	5000	D274		00274	6141		ST R0,CT441P6	
001C2C	9280	D274		00274	6142		MVI CT441P6,B'10000000'	Last Parm designation

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6144	*	*/-----/	
					6145	*	/* Link to IKJCT441	*/
					6146	*	*/-----/	
001C30	4110	D260	00260		6147		LA R1,CT441LST	R1 = Parm List
001C34	58F0	D25C	0025C		6148		L R15,IKJCT441	
001C38	05EF				6149		BALR R14,R15	
					6150	*		
001C3A	50F0	D04C	0004C		6151		ST R15,RC#441	Save R15 from IKJCT441
001C3E	12FF				6152		LTR R15,R15	Successful invocation?
001C40	4770	B87A	0187A		6153		BNZ ERR4012	No, VAR2 not updated
				01C44	6154	S#UVAR2X	EQU *	
001C44	5850	D070	00070		6155		L R5,SAVER5	
001C48	07F5				6156		BR R5	Return to caller (R5)
					6157	*		

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

						6159	*	*	/*****	
						6160	*	*	/* Subroutine - Check for \$NOVAR2 in VAR2 Var (R5) */	
						6161	*	*	/*	
						6162	*	*	/* Set VAR2 value to SPACE if \$NOVAR2 */	
						6163	*	*	/*****	
				01C4A		6164	S#CHKV2	EQU	*	
001C4A	5050	D070		00070		6165		ST	R5,SAVER5	
001C4E	D507	DB19	C7C0	00B19	027C0	6166		CLC	VAR2,=C'\$NOVAR2 '\$NOVAR2 ?	
001C54	4770	BC5C		01C5C		6167		BNE	S#CHKV2X NO, return.	
001C58	9240	DB19		00B19		6168		MVI	VAR2,C' ' YES, set VAR2 name to BLANK	
				01C5C		6169	S#CHKV2X	EQU	*	
001C5C	5850	D070		00070		6170		L	R5,SAVER5	
001C60	07F5					6171		BR	R5 Return to caller (R5)	
						6172	*			

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6174	*	/* *****/	
					6175	*	/* Subroutine - Update/Create ERRMSG Variable (R5) */	
					6176	*	/*	
					6177	*	/* Update/Create ERRMSG Variable	
					6178	*	/* - Attempt to update ERRMSG, do not check R15	
					6179	*	/* - R0, R1, R7 Working Register	
					6180	*	/* *****/	
			01C62		6181	S#UERRM	EQU *	
001C62	5050	D070	00070		6182		ST R5,SAVER5	
001C66	D21D	D2AC	D9EB	002AC	009EB	6183	MVC VAL3(L'ERRMSG),ERRMSG	Copy VARWORK to VAL3
001C6C	4170	001E	0001E		6184		LA R7,L'ERRMSG	Length of ERRMSG
001C70	5070	D3AC	003AC		6185		ST R7,VALL3	Init VARIABLE DATA LENGTH
001C74	D208	D298	C7D0	00298	027D0	6186	MVC NAM3,=C'ERRMSG '	Init VARIABLE NAME
001C7A	D203	D2A4	C824	002A4	02824	6187	MVC NAML3,=F'6'	Init VARIABLE NAME LENGTH
					6188	*	/* -----/	
					6189	*	/* Initialize IKJCT441 Parm List area */	
					6190	*	/* -----/	
001C80	4100	CAA8	02AA8		6191		LA R0,ECUPDT	Entry Code
001C84	5000	D260	00260		6192		ST R0,CT441P1	
001C88	4100	D2A8	002A8		6193		LA R0,NAMP3	Name Pointer
001C8C	5000	D264	00264		6194		ST R0,CT441P2	
001C90	4100	D2A4	002A4		6195		LA R0,NAML3	Name Length
001C94	5000	D268	00268		6196		ST R0,CT441P3	
001C98	4100	D3B0	003B0		6197		LA R0,VALP3	Var Pointer
001C9C	5000	D26C	0026C		6198		ST R0,CT441P4	
001CA0	4100	D3AC	003AC		6199		LA R0,VALL3	Var Length
001CA4	5000	D270	00270		6200		ST R0,CT441P5	
001CA8	4100	D1EC	001EC		6201		LA R0,TOKEN	Token
001CAC	5000	D274	00274		6202		ST R0,CT441P6	
001CB0	9280	D274	00274		6203		MVI CT441P6,B'10000000'	Last Parm designation
					6204	*	/* -----/	
					6205	*	/* Link to IKJCT441 */	
					6206	*	/* -----/	
001CB4	4110	D260	00260		6207		LA R1,CT441LST	R1 = Parm List
001CB8	58F0	D25C	0025C		6208		L R15,IKJCT441	
001CBC	05EF				6209		BALR R14,R15	
001CBE	50F0	D04C	0004C		6210		ST R15,RC#441	Save R15 from IKJCT441
			01CC2		6211	S#UERRMX	EQU *	
001CC2	5850	D070	00070		6212		L R5,SAVER5	
001CC6	07F5				6213		BR R5	Return to caller (R5)
					6214	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6216	*	/* *****/	
					6217	*	/* Subroutine - Update/Create VAR1 Length Var (R5) */	
					6218	*	/*	
					6219	*	/* - Attempt to update var1L, do not check R15	
					6220	*	/* - R0, R1, R2 Working Register	
					6221	*	/* *****/	
			01CC8		6222	S#UV1L	EQU *	
001CC8	5050	D070	00070		6223		ST R5,SAVER5	
001CCC	9540	D4D0	004D0		6224		CLI NAM1,C' ' Bypass if no NAM1 exists	
001CD0	4780	BD52	01D52		6225		BE S#UV1LX	
001CD4	D207	D298	D4D0	00298	004D0	6226	MVC NAM3(8),NAM1	Copy NAM1 variable name
001CDA	9240	D2A0	002A0		6227		MVI NAM3+8,C' ' Blank position 9	
001CDE	4120	D298	00298		6228		LA R2,NAM3	Starting address of NAM3
001CE2	5A20	D278	00278		6229		A R2,NAML1	Adjust address to past of NAM3
001CE6	92D3	2000	00000		6230		MVI 0(R2),C'L'	Append 'L' to variable name
001CEA	5820	D278	00278		6231		L R2,NAML1	Get NAM1 variable length
001CEE	4122	0001	00001		6232		LA R2,1(R2)	Adjust by 1
001CF2	5020	D2A4	002A4		6233		ST R2,NAML3	Store Length
001CF6	4120	0005	00005		6234		LA R2,5	Length of value (5)
001CFA	5020	D3AC	003AC		6235		ST R2,VALL3	Store Length
001CFE	5820	D280	00280		6236		L R2,VALL1	Length of NAM1 Value
001D02	4E20	D218	00218		6237		CVD R2,DW	Convert to decimal
001D06	F342	D2AC	D21D	002AC	0021D	6238	UNPK VAL3(5),DW+5(3)	Unpack and
001D0C	96F0	D2B0	002B0		6239		OI VAL3+4,X'F0'	... force an F zone
					6240	*	/* -----/	
					6241	*	/* Initialize IKJCT441 Parm List area */	
					6242	*	/* -----/	
001D10	4100	CAA8	02AA8		6243		LA R0,ECUPDT	Entry Code
001D14	5000	D260	00260		6244		ST R0,CT441P1	
001D18	4100	D2A8	002A8		6245		LA R0,NAMP3	Name Pointer
001D1C	5000	D264	00264		6246		ST R0,CT441P2	
001D20	4100	D2A4	002A4		6247		LA R0,NAML3	Name Length
001D24	5000	D268	00268		6248		ST R0,CT441P3	
001D28	4100	D3B0	003B0		6249		LA R0,VALP3	Var Pointer
001D2C	5000	D26C	0026C		6250		ST R0,CT441P4	
001D30	4100	D3AC	003AC		6251		LA R0,VALL3	Var Length
001D34	5000	D270	00270		6252		ST R0,CT441P5	
001D38	4100	D1EC	001EC		6253		LA R0,TOKEN	Token
001D3C	5000	D274	00274		6254		ST R0,CT441P6	
001D40	9280	D274	00274		6255		MVI CT441P6,B'10000000'	Last Parm designation
					6256	*	/* -----/	
					6257	*	/* Link to IKJCT441 */	
					6258	*	/* -----/	
001D44	4110	D260	00260		6259		LA R1,CT441LST	R1 = Parm List
001D48	58F0	D25C	0025C		6260		L R15,IKJCT441	
001D4C	05EF				6261		BALR R14,R15	
001D4E	50F0	D04C	0004C		6262		ST R15,RC#441	Save R15 from IKJCT441
			01D52		6263	S#UV1LX	EQU *	
001D52	5850	D070	00070		6264		L R5,SAVER5	
001D56	07F5				6265		BR R5	Return to caller (R5)
					6266	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					6268 *	/* *****/	
					6269 *	/* Subroutine - Update/Create VAR2 Length Var (R5) */	
					6270 *	/*	
					6271 *	/* - Attempt to update var2L, do not check R15	
					6272 *	/* - R0, R1, R2 Working Register	
					6273 *	/* *****/	
			01D58		6274 S#UV2L	EQU *	
001D58	5050	D070	00070		6275	ST R5,SAVER5	
001D5C	9540	D4D8	004D8		6276	CLI NAM2,C' ' Bypass if no NAM2 exists	
001D60	4780	BDEA	01DEA		6277	BE S#UV2LX	
001D64	9540	DB19	00B19		6278	CLI VAR2,C' ' VAR2 symbolic name blank?	
001D68	4780	BDEA	01DEA		6279	BE S#UV2LX YES, go to var2l exit	
001D6C	D207	D298	D4D8 00298	004D8	6280	MVC NAM3(8),NAM2 Copy NAM1 variable name	
001D72	9240	D2A0	002A0		6281	MVI NAM3+8,C' ' Blank position 9	
001D76	4120	D298	00298		6282	LA R2,NAM3 Starting address of NAM3	
001D7A	5A20	D288	00288		6283	A R2,NAML2 Adjust address to past of NAM3	
001D7E	92D3	2000	00000		6284	MVI 0(R2),C'L' Append 'L' to variable name	
001D82	5820	D288	00288		6285	L R2,NAML2 Get NAM1 variable length	
001D86	4122	0001	00001		6286	LA R2,1(R2) Adjust by 1	
001D8A	5020	D2A4	002A4		6287	ST R2,NAML3 Store Length	
001D8E	4120	0005	00005		6288	LA R2,5 Length of value (5)	
001D92	5020	D3AC	003AC		6289	ST R2,VALL3 Store Length	
001D96	5820	D290	00290		6290	L R2,VALL2 Length of NAM2 Value	
001D9A	4E20	D218	00218		6291	CVD R2,DW Convert to decimal	
001D9E	F342	D2AC	D21D 002AC	0021D	6292	UNPK VAL3(5),DW+5(3) Unpack and	
001DA4	96F0	D2B0	002B0		6293	OI VAL3+4,X'F0' ... force an F zone	
					6294 *	/* -----/	
					6295 *	/* Initialize IKJCT441 Parm List area */	
					6296 *	/* -----/	
001DA8	4100	CAA8	02AA8		6297	LA R0,ECUPDT Entry Code	
001DAC	5000	D260	00260		6298	ST R0,CT441P1	
001DB0	4100	D2A8	002A8		6299	LA R0,NAMP3 Name Pointer	
001DB4	5000	D264	00264		6300	ST R0,CT441P2	
001DB8	4100	D2A4	002A4		6301	LA R0,NAML3 Name Length	
001DBC	5000	D268	00268		6302	ST R0,CT441P3	
001DC0	4100	D3B0	003B0		6303	LA R0,VALP3 Var Pointer	
001DC4	5000	D26C	0026C		6304	ST R0,CT441P4	
001DC8	4100	D3AC	003AC		6305	LA R0,VALL3 Var Length	
001DCC	5000	D270	00270		6306	ST R0,CT441P5	
001DD0	4100	D1EC	001EC		6307	LA R0,TOKEN Token	
001DD4	5000	D274	00274		6308	ST R0,CT441P6	
001DD8	9280	D274	00274		6309	MVI CT441P6,B'10000000' Last Parm designation	
					6310 *	/* -----/	
					6311 *	/* Link to IKJCT441 */	
					6312 *	/* -----/	
001DDC	4110	D260	00260		6313	LA R1,CT441LST R1 = Parm List	
001DE0	58F0	D25C	0025C		6314	L R15,IKJCT441	
001DE4	05EF				6315	BALR R14,R15	
001DE6	50F0	D04C	0004C		6316	ST R15,RC#441 Save R15 from IKJCT441	
			01DEA		6317 S#UV2LX	EQU *	
001DEA	5850	D070	00070		6318	L R5,SAVER5	
001DEE	07F5				6319	BR R5 Return to caller (R5)	
					6320 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	16.49	04/12/20
					6322	*	/* *****/				
					6323	*	/* Subroutine - Write text to screen PUTLINE (R5) */				
					6324	*	/* save/restore R7,R8 */				
					6325	*	/* - R7, R8 Working Register */				
					6326	*	/* *****/				
				01DF0	6327	S#PUTL	EQU *				
001DF0	5050	D070	00070		6328		ST R5,SAVER5				
001DF4	9078	D074	00074		6329		STM R7,R8,SAVER7R8	Save R7,R8			
001DF8	9878	D07C	0007C		6330		LM R7,R8,MYUPT	Load UPT and ECT			
					6331		PUTLINE PARM=PUTBLK,UPT=(R7),ECT=(R8),ECB=MYECB,			X	
							OUTPUT=(OUTTXT,TERM,SINGLE,DATA),MF=(E,IOPADS)				
001E56	9878	D074	00074		6357		LM R7,R8,SAVER7R8	Restore R7,R8			
				01E5A	6358	S#PUTLX	EQU *				
001E5A	5850	D070	00070		6359		L R5,SAVER5				
001E5E	07F5				6360		BR R5	Return to caller (R5)			
					6361	*					

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6363 *	/*	*****
				6364 *	/* Subroutine - Test Numeric (TRUE/FALSE)	(R5) */
				6365 *	/* - R7 Length	*/
				6366 *	/* - R8 Address of Content	*/
				6367 *	/* - R15 Return Code	*/
				6368 *	/* 4001-TRUE	*/
				6369 *	/* 4000-FALSE	*/
				6370 *	*****	*/
		01E60		6371 S#ISNUM	EQU *	
001E60	5050 D070	00070		6372	ST R5,SAVER5	
001E64	95F0 8000	00000		6373	CLI 0(R8),C'0'	Byte < 0?
001E68	4740 BE84	01E84		6374	BL S#ISNUMF	YES, not numeric, FALSE
001E6C	95F9 8000	00000		6375	CLI 0(R8),C'9'	Byte > 9?
001E70	4720 BE84	01E84		6376	BH S#ISNUMF	YES, not numeric, FALSE
001E74	4188 0001	00001		6377	LA R8,1(R8)	Bump up to next byte of VAR1
001E78	4670 BE60	01E60		6378	BCT R7,S#ISNUM	LOOP for numeric
				6379 *		At this point, all numeric!
		01E7C		6380 S#ISNUMT	EQU *	
001E7C	41F0 0FA1	00FA1		6381	LA R15,4001	TRUE, numeric
001E80	47F0 BE88	01E88		6382	B S#ISNUMX	
		01E84		6383 S#ISNUMF	EQU *	
001E84	41F0 0FA0	00FA0		6384	LA R15,4000	FALSE, not numeric
		01E88		6385 S#ISNUMX	EQU *	
001E88	5850 D070	00070		6386	L R5,SAVER5	
001E8C	07F5			6387	BR R5	Return to caller (R5)
				6388 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6390 *	/* *****	*/
				6391 *	/* Subroutine - Determine Leap Year	(R5) */
				6392 *	/*	*/
				6393 *	/* IF CCYY MOD 4 = 0	*/
				6394 *	/* IF CCYY MOD 100 = 0	*/
				6395 *	/* IF CCYY MOD 400 = 0	*/
				6396 *	/* LEAP = 1	*/
				6397 *	/* ELSE	*/
				6398 *	/* LEAP = 0	*/
				6399 *	/* ELSE	*/
				6400 *	/* LEAP = 1	*/
				6401 *	/* ELSE	*/
				6402 *	/* LEAP = 0	*/
				6403 *	/*	*/
				6404 *	/* - Input : DCCCCYY initialized to format 0CCYY	*/
				6405 *	/* - Output: DC#LEAP 1-LeapYear 0-CommonYear	*/
				6406 *	/* - DC#YDAYS 366-LeapYear 365-CommonYear	*/
				6407 *	/* *****	*/
		01E8E		6408 S#ISLPY	EQU *	
001E8E	5050 D070	00070		6409	ST R5,SAVER5	
001E92	F800 D239 C899	00239 02899		6410	ZAP DC#LEAP,=P'0'	Assume, not leap year
001E98	F811 D23A C866	0023A 02866		6411	ZAP DC#YDAYS,=P'365'	Assume, 365 days in year
001E9E	F234 D227 D23C	00227 0023C		6412	PACK DC#YR4,DCCCCYY	Seed YR4 with 0CCYY
				6413 *		
001EA4	F833 D22B D227	0022B 00227		6414	ZAP DC#YR100,DC#YR4	Seed YR100
001EAA	F833 D22F D227	0022F 00227		6415	ZAP DC#YR400,DC#YR4	Seed YR400
001EB0	FD30 D227 C89A	00227 0289A		6416	DP DC#YR4,=P'4'	CCYY / 4
001EB6	FD31 D22B C868	0022B 02868		6417	DP DC#YR100,=P'100'	CCYY / 100
001EBC	FD31 D22F C86A	0022F 0286A		6418	DP DC#YR400,=P'400'	CCYY / 400
001EC2	F900 D22A C899	0022A 02899		6419	CP DC#YR4+3(1),=P'0'	Remainder 0?
001EC8	4720 BEEC	01EEC		6420	BH S#ISLPYX	No, not a leap year
				6421 *		Yes, check further
001ECC	F910 D22D C899	0022D 02899		6422	CP DC#YR100+2(2),=P'0'	Remainder 0?
001ED2	4720 BEE0	01EE0		6423	BH S#ISLPYT	Yes, it's leap year
				6424 *		No, check further
001ED6	F910 D231 C899	00231 02899		6425	CP DC#YR400+2(2),=P'0'	Remainder 0?
001EDC	4720 BEEC	01EEC		6426	BH S#ISLPYX	No, not leap year
		01EE0		6427 S#ISLPYT	EQU *	
001EE0	FA00 D239 C88E	00239 0288E		6428	AP DC#LEAP,=P'1'	Yes, leap year
001EE6	FA10 D23A C88E	0023A 0288E		6429	AP DC#YDAYS,=P'1'	... 366 days in year
		01EEC		6430 S#ISLPYX	EQU *	
001EEC	5850 D070	00070		6431	L R5,SAVER5	
001EF0	07F5			6432	BR R5	Return to caller (R5)
				6433 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6435 *	/* *****	/*
				6436 *	/* Subroutine - Build DATE (monthname dd, ccyy) (R5)	*/
				6437 *	/* -----	/*
				6438 *	/* R2 - Addr to receive content	*/
				6439 *	/* R3 - Length of content	*/
				6440 *	/* R4 - Addr of MONTHNAME	*/
				6441 *	/* R6 - Loop control for monthname	*/
				6442 *	/* *****	/*
001EF2	5050 D070	00070	01EF2	6443 S#BD8	EQU *	
				6444	ST R5,SAVER5	
				6445 *	/* -----	/*
				6446 *	/* Build MONTHNAME JANUARY	*/
				6447 *	/* -----	/*
			01EF6	6448 S#BMN	EQU *	
001EF6	9540 4000	00000		6449	CLI 0(R4),C' ' BLANK in MONTHNAME	
001EFA	4780 BF14	01F14		6450	BE S#BDD Yes, continue...	
001EFE	D200 2000 4000	00000 00000		6451	MVC 0(1,R2),0(R4) Move MONTHNAME	
001F04	4122 0001	00001		6452	LA R2,1(R2) Bump up VARWORK	
001F08	4144 0001	00001		6453	LA R4,1(R4) Bump up MONTHNAME	
001F0C	4133 0001	00001		6454	LA R3,1(R3) Bump up VARWORKL	
001F10	4660 BEF6	01EF6		6455	BCT R6,S#BMN Again...	
				6456 *	/* -----	/*
				6457 *	/* Build DD JANUARY 10,	*/
				6458 *	/* -----	/*
			01F14	6459 S#BDD	EQU *	Build DD
001F14	9240 2000	00000		6460	MVI 0(R2),C' ' Leave a blank after MONTHNAME	
001F18	4122 0001	00001		6461	LA R2,1(R2) Bump up VARWORK	
001F1C	4133 0001	00001		6462	LA R3,1(R3) Bump up VARWORKL	
001F20	D201 2000 D242	00000 00242		6463	MVC 0(2,R2),DCDDD+1 MOVE DD	
001F26	4122 0002	00002		6464	LA R2,2(R2) Bump up VARWORK by 2 for DD	
001F2A	4133 0002	00002		6465	LA R3,2(R3) Bump up VARWORKL by 2 for DD	
001F2E	D201 2000 C86C	00000 0286C		6466	MVC 0(2,R2),=C', ' Place ', ' after DD	
001F34	4122 0002	00002		6467	LA R2,2(R2) Bump up VARWORK by 2 for ', '	
001F38	4133 0002	00002		6468	LA R3,2(R3) Bump up VARWORKL by 2 for ', '	
				6469 *	/* -----	/*
				6470 *	/* Build CCYY JANUARY 10, 2000	*/
				6471 *	/* -----	/*
			01F3C	6472 S#BCCYY	EQU *	Build DD
001F3C	D203 2000 D23D	00000 0023D		6473	MVC 0(4,R2),DCCCCYY+1 Move CCYY	
001F42	4122 0004	00004		6474	LA R2,4(R2) Bump up VARWORK by 4 for CCYY	
001F46	4133 0004	00004		6475	LA R3,4(R3) Bump up VARWORKL by 4 for CCYY	
			01F4A	6476 S#BD8X	EQU *	
001F4A	5850 D070	00070		6477	L R5,SAVER5	
001F4E	07F5			6478	BR R5 Return to caller (R5)	
				6479 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6481 *	/* *****/	
				6482 *	/* Subroutine - Calculate Day of Week (0-6) (R5) */	
				6483 *	/* and Weekday (Sunday-Saturday) */	
				6484 *	/* -----/	
				6485 *	/* - Input : DCCCCYY initialized to format 0CCYY */	
				6486 *	/* - DCDDD initialized to format 0DD */	
				6487 *	/* - DCM MM initialized to format 0MM */	
				6488 *	/* - DC#TRM initialized to month term (packed) */	
				6489 *	/* - Output: DCCDOW 0-6 */	
				6490 *	/* - DCCWDAY Sunday-Saturday */	
				6491 *	/* -----/	
				6492 *	/* - R4, R6 Working Register */	
				6493 *	/* *****/	
001F50	5050 D070	00070	01F50	6494 S#CDOW	EQU *	
				6495	ST R5,SAVER5	
				6496 *	/* -----/	
				6497 *	/* Calculate DOW (Day of Week Number) */	
				6498 *	/* */	
				6499 *	/* Using Sakamoto's method: */	
				6500 *	/* */	
				6501 *	/* DOW(y, m, d) /* 1 <= m <= 12, y > 1752 (in the U.K.) */	
				6502 *	/* { */	
				6503 *	/* static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4}; */	
				6504 *	/* y -= m < 3; */	
				6505 *	/* return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7; */	
				6506 *	/* } */	
				6507 *	/* */	
				6508 *	/* Posted by Tomohiko Sakamoto on the comp.lang.c */	
				6509 *	/* Usenet newsgroup in 1992, it is accurate for any */	
				6510 *	/* Gregorian date. */	
				6511 *	/* */	
				6512 *	/* See below https:// link */	
				6513 *	/* en.wikipedia.org/wiki/Determination_of_the_day_of_the_week */	
				6514 *	/* -----/	
001F54	F212 D237 D241	00237	00241	6515	PACK DC#DD,DCDDD	Pack 0DD
001F5A	F234 D227 D23C	00227	0023C	6516	PACK DC#YR4,DCCCCYY	Pack 0CCYY
001F60	D501 D245 C864	00245	02864	6517	CLC DCM MM+1(2),=C'02'	> FEB?
001F66	4720 BF70		01F70	6518	BH S#CDWNA	Yes, do not adjust Year
001F6A	FB30 D227 C88E	00227	0288E	6519	SP DC#YR4,=P'1'	OCCYY - 1
			01F70	6520 S#CDWNA	EQU *	
001F70	F833 D233 D227	00233	00227	6521	ZAP DC#DOW,DC#YR4	Seed adjusted CCYY
001F76	F833 D22B D227	0022B	00227	6522	ZAP DC#YR100,DC#YR4	Seed adjusted CCYY
001F7C	F833 D22F D227	0022F	00227	6523	ZAP DC#YR400,DC#YR4	Seed adjusted CCYY
001F82	FD30 D227 C89A	00227	0289A	6524	DP DC#YR4,=P'4'	CCYY / 4
001F88	FD31 D22B C868	0022B	02868	6525	DP DC#YR100,=P'100'	CCYY / 100
001F8E	FD31 D22F C86A	0022F	0286A	6526	DP DC#YR400,=P'400'	CCYY / 400
001F94	FA32 D233 D227	00233	00227	6528	AP DC#DOW,DC#YR4(3)	+ INT(y/4)
001F9A	FB31 D233 D22B	00233	0022B	6529	SP DC#DOW,DC#YR100(2)	- INT(y/100)
001FA0	FA31 D233 D22F	00233	0022F	6530	AP DC#DOW,DC#YR400(2)	+ INT(y/400)
001FA6	FA30 D233 D224	00233	00224	6531	AP DC#DOW,DC#TRM	+ T
001FAC	FA31 D233 D237	00233	00237	6532	AP DC#DOW,DC#DD	+ DD
001FB2	FD30 D233 C89B	00233	0289B	6533	DP DC#DOW,=P'7'	DOW = INT(DOW/7)
001FB8	F300 D24A D236	0024A	00236	6534	UNPK DCCDOW,DC#DOW+3(1)	Remainder of DOW
001FBE	96F0 D24A		0024A	6535	OI DCCDOW,X'F0'	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6537	*	*/-----/	
					6538	*	/* Determine Weekday */	
					6539	*	*/-----/	
001FC2	4140	CDE7	02DE7		6540	LA	R4,DOWTABLE	R4=Address of DOWTABLE
001FC6	4160	0007	00007		6541	LA	R6,DOWTBLN	R6=Number of days
			01FCA		6542	DO#DOWSR	EQU *	
001FCA	D500	4000	D24A	00000	0024A	6543	CLC	0(1,R4),DCCDOW Match DOW ?
001FD0	4780	BFDC	01FDC		6544	BE	DO#DOWFD	Yes, use Week Day Name
001FD4	4144	000A	0000A		6545	LA	R4,DOWTBLL(R4)	No, look at next DOWTABLE entry
001FD8	4660	BFCA	01FCA		6546	BCT	R6,DO#DOWSR	Check again
			01FDC		6547	DO#DOWFD	EQU *	
001FDC	D208	D24B	4001	0024B	00001	6548	MVC	DCCWDAY(DOWTBLL-1),1(R4) Get Weekday name
			01FE2		6550	S#CDOWX	EQU *	
001FE2	5850	D070	00070		6551	L	R5,SAVER5	
001FE6	07F5				6552	BR	R5	Return to caller (R5)
					6553	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM
					6555	*	/* *****/	
					6556	*	/* Subroutine - Build CYJ input content (R5) */	
					6557	*	/* -----*/	
					6558	*	/* Build C Y J input content based on FUNCT(4) */	
					6559	*	/* - Input : VAR1 via R8 */	
					6560	*	/* - FUNCT Function request */	
					6561	*	/* - Output: DCCCCYY 0CCYY */	
					6562	*	/* - DCJJJ JJJ */	
					6563	*	/* *****/	
			01FE8		6564	S#ICYJ	EQU *	
001FE8	5050	D070	00070		6565		ST R5,SAVER5	
001FEC	92F0	D23C	0023C		6566		MVI DCCCCYY,C'0'	Move 0 to first byte DCCCCYY
001FF0	D503	DB09	C7F8	00B09	027F8	6567	CLC FUNCT(4),=C'CYJ-'	CYJ- Function
001FF6	4780	C008	02008		6568		BE INCYJ	Yes, CCYYJJJ input
001FFA	D503	DB09	C7FC	00B09	027FC	6569	CLC FUNCT(4),=C'JCY-'	JCY- Function
002000	4780	C018	02018		6570		BE INJCY	Yes, JJJCCYY input
002004	47F0	B86C	0186C		6571		B ERR4011	No, invalid function
			02008		6573	INCYJ	EQU *	R8 CCYYJJJ
002008	D203	D23D	8000	0023D	00000	6574	MVC DCCCCYY+1(4),0(R8)	Save CCYY from VAR1
00200E	D202	D247	8004	00247	00004	6575	MVC DCJJJ,4(R8)	Save JJJ from VAR1
002014	47F0	C028	02028		6576		B S#ICYJX	
			02018		6577	INJCY	EQU *	R8 JJJCCYY
002018	D203	D23D	8003	0023D	00003	6578	MVC DCCCCYY+1(4),3(R8)	Save CCYY from VAR1
00201E	D202	D247	8000	00247	00000	6579	MVC DCJJJ,0(R8)	Save JJJ from VAR1
002024	47F0	C028	02028		6580		B S#ICYJX	
			02028		6581	S#ICYJX	EQU *	
002028	5850	D070	00070		6582		L R5,SAVER5	
00202C	07F5				6583		BR R5	Return to caller (R5)
					6584	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6586 *	/* *****	/*
				6587 *	/* Subroutine - Build MDCY input content	(R5) */
				6588 *	/* -----	/*
				6589 *	/* Build M D C Y input content based on FUNCT(5)	*/
				6590 *	/* - Input : VAR1	via R8 */
				6591 *	/* - FUNCT	Function request */
				6592 *	/* - Output: DCCCCYY	0CCYY */
				6593 *	/* - DCDDD	0DD */
				6594 *	/* - DCMMM	0MM */
				6595 *	/* *****	/*
		0202E		6596 S#IMDCY	EQU *	
00202E	5050 D070	00070		6597	ST R5,SAVER5	
002032	92F0 D23C	0023C		6598	MVI DCCCCYY,C'0'	Move 0 to first byte DCCCCYY
002036	92F0 D241	00241		6599	MVI DCDDD,C'0'	Move 0 to first byte DCDDD
00203A	92F0 D244	00244		6600	MVI DCMMM,C'0'	Move 0 to first byte DCMMM
00203E	D503 DB09 C84C	00B09 0284C		6601	CLC FUNCT(4),=C'MDCY'	MDCY- Function
002044	4780 C06A	0206A		6602	BE INMDCY	Yes, MMDDCCYY input
002048	D503 DB09 C850	00B09 02850		6603	CLC FUNCT(4),=C'DMCY'	DMCY- Function
00204E	4780 C080	02080		6604	BE INDMCY	Yes, DDM MCCYY input
002052	D503 DB09 C854	00B09 02854		6605	CLC FUNCT(4),=C'CYMD'	CYMD- Function
002058	4780 C096	02096		6606	BE INCYMD	Yes, CCYYMMDD input
00205C	D503 DB09 C858	00B09 02858		6607	CLC FUNCT(4),=C'CYDM'	CYDM- Function
002062	4780 C0AC	020AC		6608	BE INCYDM	Yes, CCYYDDMM input
002066	47F0 B86C	0186C		6609	B ERR4011	No, invalid function
		0206A		6610 INMDCY	EQU *	R8 MMDDCCYY
00206A	D203 D23D 8004	0023D 00004		6611	MVC DCCCCYY+1(4),4(R8)	Save CCYY from VAR1
002070	D201 D245 8000	00245 00000		6612	MVC DCMMM+1(2),0(R8)	Save MM from VAR1
002076	D201 D242 8002	00242 00002		6613	MVC DCDDD+1(2),2(R8)	Save DD from VAR1
00207C	47F0 C0C2	020C2		6614	B S#IMDCYX	
		02080		6615 INDMCY	EQU *	R8 DDM MCCYY
002080	D203 D23D 8004	0023D 00004		6616	MVC DCCCCYY+1(4),4(R8)	Save CCYY from VAR1
002086	D201 D245 8002	00245 00002		6617	MVC DCMMM+1(2),2(R8)	Save MM from VAR1
00208C	D201 D242 8000	00242 00000		6618	MVC DCDDD+1(2),0(R8)	Save DD from VAR1
002092	47F0 C0C2	020C2		6619	B S#IMDCYX	
		02096		6620 INCYMD	EQU *	R8 CCYYMMDD
002096	D203 D23D 8000	0023D 00000		6621	MVC DCCCCYY+1(4),0(R8)	Save CCYY from VAR1
00209C	D201 D245 8004	00245 00004		6622	MVC DCMMM+1(2),4(R8)	Save MM from VAR1
0020A2	D201 D242 8006	00242 00006		6623	MVC DCDDD+1(2),6(R8)	Save DD from VAR1
0020A8	47F0 C0C2	020C2		6624	B S#IMDCYX	
		020AC		6625 INCYDM	EQU *	R8 CCYYDDMM
0020AC	D203 D23D 8000	0023D 00000		6626	MVC DCCCCYY+1(4),0(R8)	Save CCYY from VAR1
0020B2	D201 D245 8006	00245 00006		6627	MVC DCMMM+1(2),6(R8)	Save MM from VAR1
0020B8	D201 D242 8004	00242 00004		6628	MVC DCDDD+1(2),4(R8)	Save DD from VAR1
0020BE	47F0 C0C2	020C2		6629	B S#IMDCYX	
		020C2		6630 S#IMDCYX	EQU *	
0020C2	5850 D070	00070		6631	L R5,SAVER5	
0020C6	07F5			6632	BR R5	Return to caller (R5)
				6633 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6635	*	/* ***** */	
					6636	*	/* Subroutine - Build M D C Y content (R5) */	
					6637	*	/* ----- */	
					6638	*	/* R2 - addr VARWORK */	
					6639	*	/* R3 - Length of content in VARWORK */	
					6640	*	/* R4 - Addr of FUNCT */	
					6641	*	/* R6 - Loop control for FUNCT test */	
					6642	*	/* ***** */	
			020C8		6643	S#BMDCY	EQU *	
0020C8	5050	D070	00070		6644		ST R5,SAVER5	
0020CC	95D4	4000	00000		6645		CLI 0(R4),C'M'	Format MM?
0020D0	4780	C0F8	020F8		6646		BE MOVMM	Yes, move MM
0020D4	95C4	4000	00000		6647		CLI 0(R4),C'D'	Format DD?
0020D8	4780	C102	02102		6648		BE MOVDD	Yes, move DD
0020DC	95C3	4000	00000		6649		CLI 0(R4),C'C'	Format CC?
0020E0	4780	C10C	0210C		6650		BE MOVCC	Yes, move CC
0020E4	95E8	4000	00000		6651		CLI 0(R4),C'Y'	Format YY?
0020E8	4780	C116	02116		6652		BE MOVYY	Yes, move YY
0020EC	95D1	4000	00000		6653		CLI 0(R4),C'J'	Format JJJ?
0020F0	4780	C120	02120		6654		BE MOVJJJ	Yes, move JJJ
0020F4	47F0	C13E	0213E		6655		B NXTC	No, check next...
			020F8		6656	MOVMM	EQU *	
0020F8	D201	2000	D245	00000	00245		MVC 0(2,R2),DCMMM+1	MM content
0020FE	47F0	C132	02132		6658		B MOVXX	Next one...
			02102		6659	MOVDD	EQU *	
002102	D201	2000	D242	00000	00242		MVC 0(2,R2),DCDDD+1	DD content
002108	47F0	C132	02132		6661		B MOVXX	Next one...
			0210C		6662	MOVCC	EQU *	
00210C	D201	2000	D23D	00000	0023D		MVC 0(2,R2),DCCCCYY+1	CC content
002112	47F0	C132	02132		6664		B MOVXX	Next one...
			02116		6665	MOVYY	EQU *	
002116	D201	2000	D23F	00000	0023F		MVC 0(2,R2),DCCCCYY+3	YY content
00211C	47F0	C132	02132		6667		B MOVXX	Next one...
			02120		6668	MOVJJJ	EQU *	
002120	D202	2000	D247	00000	00247		MVC 0(3,R2),DCJJJ	JJJ content
002126	4122	0001	00001		6670		LA R2,1(R2)	Bump up VARWORK
00212A	4133	0001	00001		6671		LA R3,1(R3)	Bump up VARWORKL
00212E	47F0	C132	02132		6672		B MOVXX	Next one...
			02132		6673	MOVXX	EQU *	
002132	4122	0002	00002		6674		LA R2,2(R2)	Bump up VARWORK
002136	4133	0002	00002		6675		LA R3,2(R3)	Bump up VARWORKL
00213A	4144	0001	00001		6676		LA R4,1(R4)	Bump up FUNCT
			0213E		6677	NXTC	EQU *	
00213E	4660	C0C8	020C8		6678		BCT R6,S#BMDCY	Check next value
			02142		6679	S#BMDCYX	EQU *	
002142	5850	D070	00070		6680		L R5,SAVER5	
002146	07F5				6681		BR R5	Return to caller (R5)
					6682	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6684 *	*/*****	*/
				6685 *	/* Subroutine - Determine MM DD from JJJ	(R5) */
				6686 *	/* -----	*/
				6687 *	/* Determine MM and DD from JJJ	*/
				6688 *	/* - Input : DC#LEAP L packed	*/
				6689 *	/* - PKJJJ JJJ packed	*/
				6690 *	/* - Work : FW	*/
				6691 *	/* - Output: DC#TRM T packed	*/
				6692 *	/* - DCDDD 0DD	*/
				6693 *	/* - DCMMM 0MM	*/
				6694 *	/* - R4=addr of MONTHNAME	*/
				6695 *	/* - R6=Length of MONTHNAME	*/
				6696 *	/* -----	*/
				6697 *	/* - R3, R4, R5, R6 Working Register	*/
				6698 *	*/*****	*/
			02148	6699 S#J2MD	EQU *	
002148	5050 D070	00070		6700	ST R5,SAVER5	
				6701 *	/* -----	*/
				6702 *	/* Search for MM using JJJ	*/
				6703 *	/* -----	*/
00214C	4140 CDE7	02DE7		6704	LA R4,MMTABLEE	R4=End Address of MonthTable
		00000		6705	USING MPT,R4	Use MPT DSECT
002150	4150 0011	00011		6706	LA R5,MMTBLEN	R3=Length of table entry
002154	1B45			6707	SR R4,R5	R4=Address of Last MonthTable Entry
002156	4130 000C	0000C		6708	LA R3,MMTBLN	R3=12 Months
		0215A		6709 GETMMJJJ	EQU *	
00215A	F212 D220 4004	00220 00004		6710	PACK FW(2),MPTDAT@1	Pack start month days from table
002160	D501 4000 C864	00000 02864		6711	CLC MPTMM,=C'02'	JAN or FEB?
002166	47D0 C170	02170		6712	BNH NOADJLY	Yes, bypass adjust for leap year
00216A	FA10 D220 D239	00220 00239		6713	AP FW(2),DC#LEAP	No, adjust for leap year when MM >1
		02170		6714 NOADJLY	EQU *	
002170	F912 D220 D254	00220 00254		6715	CP FW(2),PKJJJ	MM JJ < JJJ
002176	4740 C182	02182		6716	BL FNDMM	Yes, found month table entry MM
00217A	1B45			6717	SR R4,R5	No, Point to prev MonthTable Entry
00217C	4630 C15A	0215A		6718	BCT R3,GETMMJJJ	Check again...
				6719 ****	something wrong if this executes	
002180	0000			6720	DC X'0000'	
				6721 *	/* -----	*/
				6722 *	/* Found MM	*/
				6723 *	/* -----	*/
			02182	6724 FNDMM	EQU *	
				6725 **	DC C'02280313February '	
				6726 *	R4 points to MM table entry	
				6727 *	DD=FW(2)-JJJ	
002182	92F0 D244	00244		6728	MVI DCMMM,C'0'	Save MM from
002186	D201 D245 4000	00245 00000		6729	MVC DCMMM+1(2),MPTMM	... table lookup
00218C	F200 D224 4007	00224 00007		6730	PACK DC#TRM,MPTDWTRM	Pack TERM for DOW calculation
002192	FB21 D254 D220	00254 00220		6731	SP PKJJJ,FW(2)	Compute DD for MM from JJJ
002198	F322 D241 D254	00241 00254		6732	UNPK DCDDD,PKJJJ	Save DD
00219E	96F0 D243	00243		6733	OI DCDDD+2,X'F0'	F-Zone
				6734 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	16.49	04/12/20
					6736	*	* /-----/				
					6737	*	/* Load R4 and R6 with MONTHNAME and Length				
					6738	*	* /-----/				
0021A2	4140	4008	00008		6739		LA R4,MPTMNAME	Point to MONTHNAME			
0021A6	4160	0009	00009		6740		LA R6,MPTMNL	R6=Length of MONTHNAME			
					6741		DROP R4	Tell Assembler			
					6742	*					
0021AA	5850	D070	00070	021AA	6743	S#J2MDX	EQU *				
					6744		L R5,SAVER5				
0021AE	07F5				6745		BR R5	Return to caller (R5)			
					6746	*					

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6748 *	/* *****	*/
				6749 *	/* Subroutine - Determine JJ from MMDD	(R5) */
				6750 *	/* -----	*/
				6751 *	/* Determine JJJ from MM DD	*/
				6752 *	/* - Input : DC#LEAP L packed	*/
				6753 *	/* - DCDDD 0DD	*/
				6754 *	/* - DCMMM 0MM	*/
				6755 *	/* - Work : FW	*/
				6756 *	/* - Output: DC#JJJ JJJ packed	*/
				6757 *	/* - DC#TRM T packed	*/
				6758 *	/* - DC#DD 0DD packed	*/
				6759 *	/* - DCJJJ JJJ	*/
				6760 *	/* - R4=addr of MONTHNAME	*/
				6761 *	/* - R6=Length of MONTHNAME	*/
				6762 *	/* -----	*/
				6763 *	/* - R3, R4, R6 Working Register	*/
				6764 *	/* *****	*/
0021B0	5050 D070	00070	021B0	6765 S#MD2J EQU *		
				6766 ST R5,SAVER5		
				6767 *	/* -----	*/
				6768 *	/* Find MMTABLE entry for MM	*/
				6769 *	/* -----	*/
0021B4	4140 CD1B	02D1B		6770 LA R4,MMTABLE	R4=Address of MonthTable	
		00000		6771 USING MPT,R4	Tell Assembler	
0021B8	4130 000C	0000C		6772 LA R3,MMTBLN	R3=12 Months	
			021BC	6773 CHKMM EQU *		
0021BC	D501 D245 4000	00245 00000		6774 CLC DCMMM+1(2),MPTMM	MM = MPTMM ?	
0021C2	4780 C1D2	021D2		6775 BE GOTMM	Yes, R4=MMTABLE entry	
0021C6	4144 0011	00011		6776 LA R4,MPTLE(R4)	No, move to next MMTABLE entry	
0021CA	4630 C1BC	021BC		6777 BCT R3,CHKMM	Check again...	
0021CE	47F0 B9BC	019BC		6778 B ERR4049	Error, Invalid MM	
				6779 *	/* -----	*/
				6780 *	/* Found MM Entry, check valid DD for MM	*/
				6781 *	/* -----	*/
			021D2	6782 GOTMM EQU *	R4=correct MMTABLE entry	
0021D2	F212 D225 4004	00225 00004		6783 ** DC C'02280313February '		
0021D8	F200 D224 4007	00224 00007		6784 PACK DC#JJJ,MPTDAT@1	Pack JJJ b4 start of month	
				6785 PACK DC#TRM,MPTDWTRM	Pack T (term for weekday)	
				6786 *		
0021DE	D501 D242 C86E	00242 0286E		6787 CLC DCDDD+1(2),=C'01'	DD < 01 ?	
0021E4	4740 B9AE	019AE		6788 BL ERR4047	Yes, Error, invalid DD	
0021E8	D501 D245 C864	00245 02864		6789 CLC DCMMM+1(2),=C'02'	MM = Feb?	
0021EE	4770 C20A	0220A		6790 BNE CHKDDMAX	No, use DD in MM for max days	
0021F2	F900 D239 C88E	00239 0288E		6791 CP DC#LEAP,=P'1'	Yes, Leap Year?	
0021F8	4770 C20A	0220A		6792 BNE CHKDDMAX	No, use DD in MM for max days	
0021FC	D501 D242 C870	00242 02870		6793 CLC DCDDD+1(2),=C'29'	Yes, DD > 29 ?	
002202	4720 B9AE	019AE		6794 BH ERR4047	Yes, Error, invalid DD	
002206	47F0 C214	02214		6795 B GOTMMX	No, done.	
			0220A	6796 CHKDDMAX EQU *		
00220A	D501 D242 4002	00242 00002		6797 CLC DCDDD+1(2),MPTMMAXD	DD > MPT MAX DD for MM	
002210	4720 B9CA	019CA		6798 BH ERR4050	Yes, Error invalid DD for MM	
			02214	6799 GOTMMX EQU *		
				6800 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6802	*	*/-----/	
					6803	*	/* Determine JJJ	*/
					6804	*	*/-----/	
002214	F212	D237	D241	00237	00241	6805	PACK DC#DD,DCDDD	Pack DD
00221A	FA11	D225	D237	00225	00237	6806	AP DC#JJJ,DC#DD	JJJ = JJJ + DD
002220	D501	D245	C864	00245	02864	6807	CLC DCMMM+1(2),=C'02'	MM > 02 (FEB) ?
002226	47D0	C230		02230		6808	BNH GOTJJJX	No. Do not adjust
00222A	FA10	D225	D239	00225	00239	6809	AP DC#JJJ,DC#LEAP	Yes. Adjust JJJ for leap year
				02230		6810	GOTJJJX EQU *	
002230	F321	D247	D225	00247	00225	6811	UNPK DCJJJ,DC#JJJ	DCJJJ and fix F-zone
002236	96F0	D249		00249		6812	OI DCJJJ+2,X'F0'	...
					6813	*	*/-----/	
					6814	*	/* Load R4 and R6 with MONTHNAME and Length	*/
					6815	*	*/-----/	
00223A	4140	4008		00008		6816	LA R4,MPTMNAME	Point to MONTHNAME
00223E	4160	0009		00009		6817	LA R6,MPTMNL	R6=Length of MONTHNAME
					6818		DROP R4	Tell Assembler
					6819	*		
				02242		6820	S#MD2JX EQU *	
002242	5850	D070		00070		6821	L R5,SAVER5	
002246	07F5					6822	BR R5	Return to caller (R5)
					6823	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6825	*	/* ***** */	
					6826	*	/* Subroutine - Updt/Create VAR1nnn Variable (R5) */	
					6827	*	/* ----- */	
					6828	*	/* Input: */	
					6829	*	/* R3 - Length of content in VARWORK */	
					6830	*	/* VARWORK */	
					6831	*	/* NAM1 */	
					6832	*	/* ***** */	
					6833	*	/* - R0, R1, R2 Working Register */	
					6834	*	/* ***** */	
				02248	6835	S#CVARN	EQU *	
002248	5050	D070	00070		6836		ST R5,SAVER5	
00224C	9540	D4D0	004D0		6837		CLI NAM1,C' ' Bypass if no NAM1 exists	
002250	4780	C380	02380		6838		BE S#CVARNX	
					6839	*	/* ***** */	
					6840	*	/* Updt/Create VAR1nnn Variable */	
					6841	*	/* ***** */	
					6842	*	/* ----- */	
					6843	*	/* VAR1nnn Variable Value and Length */	
					6844	*	/* ----- */	
002254	D2FF	D3C8	D6E4	003C8	006E4	6845	MVC VAL4,VARWORK	Move VARWORK content
00225A	5030	D4C8		004C8		6846	ST R3,VALL4	Store Length of VARWORK content
00225E	FA10	D1D4	C88E	001D4	0288E	6847	AP NNN,=P'1'	Add 1 to nnn
					6848	*	/* ----- */	
					6849	*	/* VAR1nnn Variable Name w/ appended NNN */	
					6850	*	/* ----- */	
002264	4120	D3B4		003B4		6851	LA R2,NAM4	Address of NAM4 (copy of NAM1)
002268	5A20	D278		00278		6852	A R2,NAML1	Point to past end of var name
00226C	D207	D3B4	D4D0	003B4	004D0	6853	MVC NAM4(8),NAM1	Copy NAM1 variable name
002272	F910	D1D4	C89C	001D4	0289C	6854	CP NNN,=P'9'	0-9
002278	47D0	C2AA		022AA		6855	BNH S#CVAR1	Yes, 1 digit
00227C	F911	D1D4	C872	001D4	02872	6856	CP NNN,=P'99'	10-99
002282	47D0	C298		02298		6857	BNH S#CVAR2	Yes, 2 digits
					6858	*	No, Must be...	
				02286	6859	S#CVAR3	EQU *	3 digits, 100-999
002286	F321	2000	D1D4	00000	001D4	6860	UNPK 0(3,R2),NNN	Append to VAR name
00228C	96F0	2002		00002		6861	OI 2(R2),X'F0'	F-Zone
002290	4110	0003		00003		6862	LA R1,3	Number of digits - 3
002294	47F0	C2BC		022BC		6863	B S#CVARC	
				02298	6864	S#CVAR2	EQU *	2 Digits, 10-99
002298	F311	2000	D1D4	00000	001D4	6865	UNPK 0(2,R2),NNN	Append to VAR name
00229E	96F0	2001		00001		6866	OI 1(R2),X'F0'	F-Zone
0022A2	4110	0002		00002		6867	LA R1,2	Number of digits - 2
0022A6	47F0	C2BC		022BC		6868	B S#CVARC	
				022AA	6869	S#CVAR1	EQU *	1 Digit, 0-9
0022AA	F300	2000	D1D5	00000	001D5	6870	UNPK 0(1,R2),NNN+1(1)	Append to VAR name
0022B0	96F0	2000		00000		6871	OI 0(R2),X'F0'	F-Zone
0022B4	4110	0001		00001		6872	LA R1,1	Number of digits - 2
0022B8	47F0	C2BC		022BC		6873	B S#CVARC	
				022BC	6874	S#CVARC	EQU *	Common Exit
					6875	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					6877	*	*/-----/	
					6878	*	/* VAR1nnn Variable Name Length */	
					6879	*	*/-----/	
0022BC	5A10	D278	00278		6880		A R1,NAML1 Digits + NAM1 Variable length	
0022C0	5010	D3C0	003C0		6881		ST R1,NAML4 Store Length	
					6882	*	*/-----/	
					6883	*	/* Initialize IKJCT441 Parm List area */	
					6884	*	*/-----/	
0022C4	4100	CAA8	02AA8		6885		LA R0,ECUPDT Entry Code	
0022C8	5000	D260	00260		6886		ST R0,CT441P1	
0022CC	4100	D3C4	003C4		6887		LA R0,NAMP4 Name Pointer	
0022D0	5000	D264	00264		6888		ST R0,CT441P2	
0022D4	4100	D3C0	003C0		6889		LA R0,NAML4 Name Length	
0022D8	5000	D268	00268		6890		ST R0,CT441P3	
0022DC	4100	D4CC	004CC		6891		LA R0,VALP4 Var Pointer	
0022E0	5000	D26C	0026C		6892		ST R0,CT441P4	
0022E4	4100	D4C8	004C8		6893		LA R0,VALL4 Var Length	
0022E8	5000	D270	00270		6894		ST R0,CT441P5	
0022EC	4100	D1EC	001EC		6895		LA R0,TOKEN Token	
0022F0	5000	D274	00274		6896		ST R0,CT441P6	
0022F4	9280	D274	00274		6897		MVI CT441P6,B'10000000' Last Parm designation	
					6898	*	*/-----/	
					6899	*	/* Link to IKJCT441 */	
					6900	*	*/-----/	
0022F8	4110	D260	00260		6901		LA R1,CT441LST R1 = Parm List	
0022FC	58F0	D25C	0025C		6902		L R15,IKJCT441	
002300	05EF				6903		BALR R14,R15	
					6904	*		
002302	50F0	D04C	0004C		6905		ST R15,RC#441 Save R15 from IKJCT441	
					6906	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6908 *	*/*****	
				6909 *	/* Updt/Create VAR10 Variable */	
				6910 *	*/*****	
				6911 *	/*-----	
				6912 *	/* VAR10 Variable Value from NNN */	
				6913 *	/*-----	
002306	F321 D3C8 D1D4	003C8	001D4	6914	UNPK VAL4(3),NNN(2)	Unpack and
00230C	96F0 D3CA	003CA		6915	OI VAL4+2,X'F0'	... force an F zone
				6916 *	/*-----	
				6917 *	/* VAR10 Variable Value Length set to 3 */	
				6918 *	/*-----	
002310	4100 0003	00003		6919	LA R0,3	
002314	5000 D4C8	004C8		6920	ST R0,VALL4	Store Length of VARWORK content
				6921 *	/*-----	
				6922 *	/* VAR10 Variable Name */	
				6923 *	/*-----	
002318	D207 D3B4 D4D0	003B4	004D0	6924	MVC NAM4(8),NAM1	Copy NAM1 variable name
00231E	D202 D3BC DBBA	003BC	00BBA	6925	MVC NAM4+8(3),BLANKS	and blank remainder
002324	4120 D3B4	003B4		6926	LA R2,NAM4	Address of NAM4 (copy of NAM1)
002328	5A20 D278	00278		6927	A R2,NAML1	Point to past end of var name
00232C	D200 2000 C89D	00000	0289D	6928	MVC 0(1,R2),=C'0'	VAR10 Name
				6929 *	/*-----	
				6930 *	/* VAR10 Variable Name Length */	
				6931 *	/*-----	
002332	5800 D278	00278		6932	L R0,NAML1	Store Length for Var Name
002336	5A00 C7F4	027F4		6933	A R0,=F'1'	Adjust by 1
00233A	5000 D3C0	003C0		6934	ST R0,NAML4	Store Length
				6935 *	/*-----	
				6936 *	/* Initialize IKJCT441 Parm List area */	
				6937 *	/*-----	
00233E	4100 CAA8	02AA8		6938	LA R0,ECUPDT	Entry Code
002342	5000 D260	00260		6939	ST R0,CT441P1	
002346	4100 D3C4	003C4		6940	LA R0,NAMP4	Name Pointer
00234A	5000 D264	00264		6941	ST R0,CT441P2	
00234E	4100 D3C0	003C0		6942	LA R0,NAML4	Name Length
002352	5000 D268	00268		6943	ST R0,CT441P3	
002356	4100 D4CC	004CC		6944	LA R0,VALP4	Var Pointer
00235A	5000 D26C	0026C		6945	ST R0,CT441P4	
00235E	4100 D4C8	004C8		6946	LA R0,VALL4	Var Length
002362	5000 D270	00270		6947	ST R0,CT441P5	
002366	4100 D1EC	001EC		6948	LA R0,TOKEN	Token
00236A	5000 D274	00274		6949	ST R0,CT441P6	
00236E	9280 D274	00274		6950	MVI CT441P6,B'10000000'	Last Parm designation
				6951 *	/*-----	
				6952 *	/* Link to IKJCT441 */	
				6953 *	/*-----	
002372	4110 D260	00260		6954	LA R1,CT441LST	R1 = Parm List
002376	58F0 D25C	0025C		6955	L R15,IKJCT441	
00237A	05EF			6956	BALR R14,R15	
				6957 *		
00237C	50F0 D04C	0004C		6958	ST R15,RC#441	Save R15 from IKJCT441
			02380	6959	S#CVARNX EQU *	
002380	5850 D070	00070		6960	L R5,SAVER5	
002384	07F5			6961	BR R5	Return to caller (R5)
				6962 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				6964 *	/* *****	*/
				6965 *	/* Subroutine - Get Search Params	(R1) */
				6966 *	/* -----	*/
				6967 *	/* Input:	*/
				6968 *	/* R3 - Length of content in VARWORK	*/
				6969 *	/* VARWORK	*/
				6970 *	/* NAM1	*/
				6971 *	/* *****	*/
				6972 *	/* Input:	*/
				6973 *	/* -R5 VAR2 addr	*/
				6974 *	/* -R6 VAR2 length	*/
				6975 *	/* - R0, R2, R3, R4	Working Registers */
				6976 *	/* *****	*/
		02386		6977 S#SRCHP	EQU *	
				6978 *	/* -----	*/
				6979 *	/* Load Search String 1	*/
				6980 *	/* -----	*/
002386	4130 D7E4	007E4		6981	LA R3,SRCHSTR	Address of SRCHSTR
				6982 *	/* "ssss ssss","rrrrrrr",bbb,eee	<-- VAR2 template */
				6983 *	/* ^^^^^^^^^^^^^	*/
				6984 *	/* -----	*/
				6985 *	/* Need " in position 1	*/
				6986 *	/* -----	*/
00238A	957F 5000	00000		6987	CLI 0(R5),C'"'	Double Quote (DQ) ?
00238E	4770 B914	01914		6988	BNE ERR4030	NO, error, need DQ
002392	47F0 C3AC	023AC		6989	B LOADSSL	YES, load search substring
			02396	6990 LOADSS	EQU *	Load substring until DQ
002396	4155 0001	00001		6991	LA R5,1(R5)	Bump to next VAR2 byte
00239A	957F 5000	00000		6992	CLI 0(R5),C'"'	Double Quote?
00239E	4780 C3B4	023B4		6993	BE LOADSSX	YES, done with substring
0023A2	D200 3000 5000	00000 00000		6994	MVC 0(1,R3),0(R5)	NO, load substring
0023A8	4133 0001	00001		6995	LA R3,1(R3)	Bump to next substring byte
			023AC	6996 LOADSSL	EQU *	
0023AC	4660 C396	02396		6997	BCT R6,LOADSS	Keep loading substring...
0023B0	47F0 B922	01922		6998	B ERR4031	No ending DQ found
				6999 *	/* -----	*/
				7000 *	/* Search string ended with "	*/
				7001 *	/* -----	*/
			023B4	7002 LOADSSX	EQU *	Done with SS after DQ found
0023B4	4140 D7E4	007E4		7003	LA R4,SRCHSTR	Determine length
0023B8	1B34			7004	SR R3,R4	of search string
0023BA	5030 D1FC	001FC		7005	ST R3,SRCHSTR	Save search string length
0023BE	5930 C7D8	027D8		7006	C R3,=F'0'	Length = 0?
0023C2	4780 B968	01968		7007	BE ERR4036	YES, null string not allowed
				7008 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
0023C6	D507	DB09	C798	00B09	02798	7010	CLC	FUNCT,=C'REPLACE ' Replace needs string 2
0023CC	4770	C440		02440		7011	BNE	LOADBEGA Not replace, go to start position
				023D0		7012	LOADST2	
						7013	*	/* ----- */
						7014	*	/* Load Search String 2, optional */
						7015	*	/* ----- */
						7016	*	/* "ssss ssss", "rrrrrrr", bbb, eee <-- VAR2 template */
						7017	*	/* ^ */
						7018	*	/* ----- */
						7019	*	/* Decrement R6 for " on exit from prior parm */
						7020	*	/* ----- */
0023D0	4660	C3D8		023D8		7021	BCT	R6, LOADST20 xx Decrement for "
0023D4	47F0	C504		02504		7022	B	POSDONE Loop expired, done w parm scan
				023D8		7023	LOADST20	EQU *
0023D8	4130	D8E4		008E4		7024	LA	R3, SRCHST2 Addr of String2
0023DC	4120	0000		00000		7025	LA	R2, 0 Length String2
						7026	*	/* "ssss ssss", "rrrrrrr", bbb, eee <-- VAR2 memplate */
						7027	*	/* ^ */
						7028	*	/* ----- */
						7029	*	/* Decrement R6 for delimit check */
						7030	*	/* ----- */
0023E0	4660	C3E8		023E8		7031	BCT	R6, LOADST21 xx Decrement for delim chk
0023E4	47F0	C504		02504		7032	B	POSDONE Loop expired, done w parm scan
				023E8		7033	LOADST21	EQU *
0023E8	4155	0001		00001		7034	LA	R5, 1(R5) Bump to next VAR2 byte
0023EC	9540	5000		00000		7035	CLI	0(R5), C' ' Blank Delimiter?
0023F0	4780	C504		02504		7036	BE	POSDONE Yes, Done. go process...
0023F4	956B	5000		00000		7037	CLI	0(R5), C', ' Comma?
0023F8	4780	C400		02400		7038	BE	LOADS2B Yes, do string2
0023FC	47F0	B93E		0193E		7039	B	ERR4033 Error, inv char after search string
						7040	*	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7042 *	/* "ssss ssss", "rrrrrrr", bbb, eee	<-- VAR2 template */
				7043 *	/* ^	*/
				7044 *	/* -----	*/
				7045 *	/* Need " in next position	*/
				7046 *	/* -----	*/
		02400		7047 LOADS2B	EQU *	
002400	4155 0001	00001		7048	LA R5, 1(R5)	Bump to next VAR2 byte
002404	957F 5000	00000		7049	CLI 0(R5), C'"'	Double Quote (DQ) ?
002408	4770 B914	01914		7050	BNE ERR4030	NO, error, need DQ
00240C	47F0 C426	02426		7051	B LOADS2L	YES, load search substring2
				7052 *	/* "ssss ssss", "rrrrrrr", bbb, eee	<-- VAR2 template */
				7053 *	/* ^^^^^^^^	*/
		02410		7054 LOADS2	EQU *	Load substring2 until DQ
002410	4155 0001	00001		7055	LA R5, 1(R5)	Bump to next VAR2 byte
002414	957F 5000	00000		7056	CLI 0(R5), C'"'	Double Quote?
002418	4780 C42E	0242E		7057	BE LOADS2X	YES, done with substring
00241C	D200 3000 5000	00000 00000		7058	MVC 0(1, R3), 0(R5)	NO, load substring
002422	4133 0001	00001		7059	LA R3, 1(R3)	Bump to next substring byte
		02426		7060 LOADS2L	EQU *	
002426	4660 C410	02410		7061	BCT R6, LOADS2	Keep loading substring...
00242A	47F0 B922	01922		7062	B ERR4031	No ending DQ found
				7063 *	/* Search string ended with "	*/
		0242E		7064 LOADS2X	EQU *	Done with SS after DQ found
00242E	4140 D8E4	008E4		7065	LA R4, SRCHST2	Determine length
002432	1B34			7066	SR R3, R4	of search string
002434	5030 D200	00200		7067	ST R3, SRCHST2L	Save search string length
002438	5930 C7D8	027D8		7068	C R3, =F'0'	Length = 0?
00243C	4780 B968	01968		7069	BE ERR4036	YES, null string not allowed
				7070 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
					7072 *	/* ----- */	
					7073 *	/* Load Search Start position 3-digits */	
					7074 *	/* ----- */	
			02440		7075 LOADBEGA EQU *		
					7076 *	/* "ssss ssss", "rrrrrrr", bbb, eee <-- VAR2 template */	
					7077 *	/* ^ */	
					7078 *	/* ----- */	
					7079 *	/* Decrement R6 for " on exit from prior parm */	
					7080 *	/* ----- */	
002440	4660	C448	02448		7081	BCT R6, LOADBEG0 xx Decrement for "	
002444	47F0	C504	02504		7082	B POSDONE Loop expired, done w parm scan	
			02448		7083 LOADBEG0 EQU *		
					7084 *	/* ----- */	
					7085 *	/* Setup for search begin position */	
					7086 *	/* ----- */	
002448	4130	D9E4	009E4		7087	LA R3, SRCHBEG Addr of POS BEG	
00244C	4120	0000	00000		7088	LA R2, 0 Length POS BEG	
					7089 *	/* "ssss ssss", "rrrrrrr", bbb, eee <-- VAR2 template */	
					7090 *	/* ^ */	
					7091 *	/* ----- */	
					7092 *	/* Decrement R6 for delimit check */	
					7093 *	/* ----- */	
002450	4660	C458	02458		7094	BCT R6, LOADBEG1 xx Decrement for delim chk	
002454	47F0	C504	02504		7095	B POSDONE Loop expired, done w parm scan	
			02458		7096 LOADBEG1 EQU *		
					7097 *	/* "ssss ssss", "rrrrrrr", bbb, eee <-- VAR2 template */	
					7098 *	/* ^^^^ */	
002458	4155	0001	00001		7099	LA R5, 1(R5) Bump to next VAR2 byte	
00245C	9540	5000	00000		7100	CLI 0(R5), C' ' Blank Delimiter?	
002460	4780	C504	02504		7101	BE POSDONE Yes, Done. go process...	
002464	956B	5000	00000		7102	CLI 0(R5), C', ' Comma?	
002468	4780	C470	02470		7103	BE LOADBEG Yes, up to 3 digits start	
00246C	47F0	B93E	0193E		7104	B ERR4033 Error, inv char after search string	
			02470		7105 LOADBEG EQU *		
002470	4155	0001	00001		7106	LA R5, 1(R5)	
002474	9540	5000	00000		7107	CLI 0(R5), C' ' Blank Delimiter?	
002478	4780	C504	02504		7108	BE POSDONE Yes, Done. go process...	
00247C	956B	5000	00000		7109	CLI 0(R5), C', ' Comma?	
002480	4780	C4B6	024B6		7110	BE LOADENDA Yes, up to 3 digits end	
002484	95F0	5000	00000		7111	CLI 0(R5), C'0' < 0 ?	
002488	4740	B94C	0194C		7112	BL ERR4034 YES. must be numeric	
00248C	95F9	5000	00000		7113	CLI 0(R5), C'9' > 9 ?	
002490	4720	B94C	0194C		7114	BH ERR4034 YES. must be numeric	
002494	D200	3000	5000 00000 00000		7115	MVC 0(1, R3), 0(R5) load start position byte	
00249A	4133	0001	00001		7116	LA R3, 1(R3) Bump to next substring byte	
00249E	4122	0001	00001		7117	LA R2, 1(R2) Count digits	
0024A2	5020	D1F4	001F4		7118	ST R2, SRCHBEG Store Length	
0024A6	5920	C7DC	027DC		7119	C R2, =F'3' More than 3 digits?	
0024AA	4720	B95A	0195A		7120	BH ERR4035 YES, error. too many digits	
0024AE	4660	C470	02470		7121	BCT R6, LOADBEG Keep loading substring...	
0024B2	47F0	C504	02504		7122	B POSDONE Loop expired, done w parm scan	
					7123 *		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					7125	*	*/-----/	
					7126	*	/* Load Search End position 3-digits	*/
					7127	*	*/-----/	
				024B6	7128	LOADENDA	EQU *	
					7129	*	/* "ssss ssss", "rrrrrrr", bbb, eee	<-- VAR2 template */
					7130	*	/*	*/
					7131	*	/* Decrement R6 for , exit above	*/
0024B6	4660	C4BE	024BE		7132	BCT	R6, LOADEND0	xx Decrement for ,
0024BA	47F0	C504	02504		7133	B	POSDONE	Loop expired, done w parm scan
				024BE	7134	LOADEND0	EQU *	
					7135	*	*/-----/	
					7136	*	/* Setup for search end position	*/
					7137	*	*/-----/	
0024BE	4130	D9E7	009E7		7138	LA	R3, SRCHEND	Addr of POS END
0024C2	4120	0000	00000		7139	LA	R2, 0	Addr of POS END
				024C6	7140	LOADEND	EQU *	
					7141	*	/* "ssss ssss", "rrrrrrr", bbb, eee	<-- VAR2 template */
					7142	*	/*	*/
0024C6	4155	0001	00001		7143	LA	R5, 1(R5)	
0024CA	9540	5000	00000		7144	CLI	0(R5), C' '	Blank Delimiter?
0024CE	4780	C504	02504		7145	BE	POSDONE	Yes, Done. go process...
0024D2	95F0	5000	00000		7146	CLI	0(R5), C'0'	< 0 ?
0024D6	4740	B94C	0194C		7147	BL	ERR4034	YES. must be numeric
0024DA	95F9	5000	00000		7148	CLI	0(R5), C'9'	> 9 ?
0024DE	4720	B94C	0194C		7149	BH	ERR4034	YES. must be numeric
0024E2	D200	3000	5000	00000	00000	7150	MVC	0(1, R3), 0(R5)
0024E8	4133	0001	00001		7151	LA	R3, 1(R3)	Bump to next substring byte
0024EC	4122	0001	00001		7152	LA	R2, 1(R2)	Count digits
0024F0	5020	D1F8	001F8		7153	ST	R2, SRCHENDL	Store Length
0024F4	5920	C7DC	027DC		7154	C	R2, =F'3'	More than 3 digits?
0024F8	4720	B95A	0195A		7155	BH	ERR4035	YES, error. too many digits
0024FC	4660	C4C6	024C6		7156	BCT	R6, LOADEND	Keep loading substring...
002500	47F0	C504	02504		7157	B	POSDONE	Loop expired, done w parm scan
					7158	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					7160	*	*/-----/	
					7161	*	/* Set Search Begin position	*/
					7162	*	/* - R0	Working Register
					7163	*	*/-----/	
				02504	7164	POSDONE	EQU *	
002504	5800	D1F4	001F4		7165		L R0,SRCHBEG	
002508	5900	C7D8	027D8		7166		C R0,=F'0'	BEG position specified?
00250C	4780	C53E	0253E		7167		BE DFLTBE	NO, use default 1
002510	5900	C7F4	027F4		7168		C R0,=F'1'	BEG pos len = 1?
002514	4780	C534	02534		7169		BE SRCHBPK1	YES.
002518	5900	C83C	0283C		7170		C R0,=F'2'	BEG pos len = 2?
00251C	4780	C52A	0252A		7171		BE SRCHBPK2	YES.
				02520	7172	SRCHBPK3	EQU *	Must be pos len = 3...
002520	F272	D218	D9E4	00218	009E4	7173	PACK DW,SRCHBEG(3)	
002526	47F0	C548		02548	7174		B SRCHBPKX	
				0252A	7175	SRCHBPK2	EQU *	
00252A	F271	D218	D9E4	00218	009E4	7176	PACK DW,SRCHBEG(2)	
002530	47F0	C548		02548	7177		B SRCHBPKX	
				02534	7178	SRCHBPK1	EQU *	
002534	F270	D218	D9E4	00218	009E4	7179	PACK DW,SRCHBEG(1)	
00253A	47F0	C548		02548	7180		B SRCHBPKX	
				0253E	7181	DFLTBE	EQU *	
00253E	F270	D218	C898	00218	02898	7182	PACK DW,=C'1'	
002544	47F0	C548		02548	7183		B SRCHBPKX	
				02548	7184	SRCHBPKX	EQU *	
002548	4F00	D218		00218	7185		CVB R0,DW	Convert to binary
00254C	5000	D204		00204	7186		ST R0,SRCHS	Store
				02550	7187	SRCHBOK	EQU *	
002550	5900	D280		00280	7188		C R0,VALL1	Start POS > VAR1 Len?
002554	47D0	C560		02560	7189		BNH SRCHBXT	NO, within reason...
002558	4100	0001		00001	7190		LA R0,1	YES, override start POS
00255C	5000	D204		00204	7191		ST R0,SRCHS	Store
				02560	7192	SRCHBXT	EQU *	
					7193	*		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					7195	*	*/-----/	
					7196	*	/* Set Search End position	*/
					7197	*	/* - R0 Working Register	*/
					7198	*	*/-----/	
002560	5800	D1F8	001F8		7199		L R0,SRCHENDL	
002564	5900	C7D8	027D8		7200		C R0,=F'0'	BEG position specified?
002568	4780	C59A	0259A		7201		BE DFLTEND	NO, use default 999
00256C	5900	C7F4	027F4		7202		C R0,=F'1'	BEG pos len = 1?
002570	4780	C590	02590		7203		BE SRCHEPK1	YES.
002574	5900	C83C	0283C		7204		C R0,=F'2'	BEG pos len = 2?
002578	4780	C586	02586		7205		BE SRCHEPK2	YES.
				0257C	7206	SRCHEPK3	EQU *	NO, Must be pos len = 3...
00257C	F272	D218	D9E7	00218	009E7	7207	PACK DW,SRCHEND(3)	
002582	47F0	C5A4	025A4		7208		B SRCHEPKX	
				02586	7209	SRCHEPK2	EQU *	
002586	F271	D218	D9E7	00218	009E7	7210	PACK DW,SRCHEND(2)	
00258C	47F0	C5A4	025A4		7211		B SRCHEPKX	
				02590	7212	SRCHEPK1	EQU *	
002590	F270	D218	D9E7	00218	009E7	7213	PACK DW,SRCHEND(1)	
002596	47F0	C5A4	025A4		7214		B SRCHEPKX	
				0259A	7215	DFLTEND	EQU *	
00259A	F272	D218	C89E	00218	0289E	7216	PACK DW,=C'999'	
0025A0	47F0	C5A4	025A4		7217		B SRCHEPKX	
				025A4	7218	SRCHEPKX	EQU *	
0025A4	4F00	D218	00218		7219		CVB R0,DW	Convert to binary
0025A8	5000	D208	00208		7220		ST R0,SRCHE	Store
				025AC	7221	SRCHEOK	EQU *	
0025AC	5900	D280	00280		7222		C R0,VALL1	End POS > VAR1 Len?
0025B0	47D0	C5BA	025BA		7223		BNH SRCHEXT	NO, within reason...
0025B4	D203	D208	D280	00208	00280	7224	MVC SRCHE,VALL1	YES, override end POS
				025BA	7225	SRCHEXT	EQU *	
				025BA	7226	S#SRCHPX	EQU *	
0025BA	07F1				7227		BR R1	Return to caller (R1)
					7228	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7230 *	* /*****	
				7231 *	* /* Literal Pool	
				7232 *	* /*****	
0025C0				7233	LTORG	
0025C0	E2C5E3E5C1D94040			7234	=C' SETVAR	
0025C8	C1C2D6E4E3404040			7235	=C' ABOUT	
0025D0	D3E3D9C9D4404040			7236	=C' LTRIM	
0025D8	D9E3D9C9D4404040			7237	=C' RTRIM	
0025E0	E3D9C9D440404040			7238	=C' TRIM	
0025E8	E3D9C9D45B404040			7239	=C' TRIM\$	
0025F0	C9D5C4C5E7404040			7240	=C' INDEX	
0025F8	C9D5C4C5E7C24040			7241	=C' INDEXB	
002600	C9E2D5E4D4404040			7242	=C' ISNUM	
002608	C9E2C1D3D7C8C140			7243	=C' ISALPHA	
002610	C9E2D3D6E6C5D940			7244	=C' ISLOWER	
002618	C9E2E4D7D7C5D940			7245	=C' ISUPPER	
002620	C9E2C2D3C1D5D240			7246	=C' ISBLANK	
002628	C9E2C4E2D5404040			7247	=C' ISDSN	
002630	C5C3C8D640404040			7248	=C' ECHO	
002638	C5C3C8D6D8404040			7249	=C' ECHOQ	
002640	D9C5E5D9E2404040			7250	=C' REVR\$	
002648	E4D7D7C5D9C34040			7251	=C' UPPERC	
002650	D3D6E6C5D9C34040			7252	=C' LOWERC	
002658	C3D6E4D5E3404040			7253	=C' COUNT	
002660	C6C9D5C4C1D3D340			7254	=C' FINDALL	
002668	C6C9D5C4D3404040			7255	=C' FINDL	
002670	C6C9D5C440404040			7256	=C' FIND	
002678	C3C5D5E3C5D94040			7257	=C' CENTER	
002680	D3D1E4E2E3404040			7258	=C' LJUST	
002688	D9D1E4E2E3404040			7259	=C' RJUST	
002690	E9C6C9D3D3404040			7260	=C' ZFILL	
002698	E6D6D9C4E2404040			7261	=C' WORDS	
0026A0	C7C5D57B40404040			7262	=C' GEN#	
0026A8	C4C4F2C4E2D54040			7263	=C' DD2DSN	
0026B0	D1D6C2C9D5C6D640			7264	=C' JOBINFO	
0026B8	C4C1E8E2D4D44040			7265	=C' DAYSMM	
0026C0	C4C1E8E2E8E84040			7266	=C' DAYSYY	
0026C8	C9E2D3C5C1D74040			7267	=C' ISLEAP	
0026D0	C3E8D160C4F84040			7268	=C' CYJ-D8	
0026D8	C3E8D160C4C1E840			7269	=C' CYJ-DAY	
0026E0	C3E8D160C4D6E640			7270	=C' CYJ-DOW	
0026E8	D1C3E860C4F84040			7271	=C' JCY-D8	
0026F0	D1C3E860C4C1E840			7272	=C' JCY-DAY	
0026F8	D1C3E860C4D6E640			7273	=C' JCY-DOW	
002700	D4C4C3E860C4F840			7274	=C' MDCY-D8	
002708	D4C4C3E860C4C1E8			7275	=C' MDCY-DAY	
002710	D4C4C3E860C4D6E6			7276	=C' MDCY-DOW	
002718	C4D4C3E860C4F840			7277	=C' DMCY-D8	
002720	C4D4C3E860C4C1E8			7278	=C' DMCY-DAY	
002728	C4D4C3E860C4D6E6			7279	=C' DMCY-DOW	
002730	C3E8D4C460C4F840			7280	=C' CYMD-D8	
002738	C3E8D4C460C4C1E8			7281	=C' CYMD-DAY	
002740	C3E8D4C460C4D6E6			7282	=C' CYMD-DOW	
002748	C3E8C4D460C4F840			7283	=C' CYDM-D8	
002750	C3E8C4D460C4C1E8			7284	=C' CYDM-DAY	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
002758	C3E8C4D460C4D6E6			7285	=C'CYDM-DOW'	
002760	C6C9D3D340404040			7286	=C'FILL'	
002768	D3E2E3D9C9D74040			7287	=C'LSTRIP'	
002770	D9E2E3D9C9D74040			7288	=C'RSTRIP'	
002778	E2E3D9C9D7404040			7289	=C'STRIP'	
002780	E2E3D9C9D75B4040			7290	=C'STRIP\$'	
002788	C3D6D5C3C1E34040			7291	=C'CONCAT'	
002790	E4D5E2E3D9404040			7292	=C'UNSTR'	
002798	D9C5D7D3C1C3C540			7293	=C'REPLACE'	
0027A0	E5C5E7C9E2E34040			7294	=C'VEXIST'	
0027A8	E3C4E2D540404040			7295	=C'TDSN'	
0027B0	D5D6E64040404040			7296	=C'NOW'	
0027B8	D7C1C44040404040			7297	=C'PAD'	
0027C0	5BD5D6E5C1D9F240			7298	=C'\$NOVAR2'	
0027C8	D8E4D6E3C5404040			7299	=C'QUOTE'	
0027D0	C5D9D9D4E2C74040			7300	=C'ERRMSG'	
0027D8	00000000			7301	=F'0'	
0027DC	00000003			7302	=F'3'	
0027E0	606040E2C5E3E5C1			7303	=C'-- SETVAR word 3'	
0027F4	00000001			7304	=F'1'	
0027F8	C3E8D160			7305	=C'CYJ-'	
0027FC	D1C3E860			7306	=C'JCY-'	
002800	00000FA0			7307	=F'4000'	
002804	0000002C			7308	=F'44'	
002808	00000008			7309	=F'8'	
00280C	00000010			7310	=F'16'	
002810	C4C44BD5D6E34BC6			7311	=C'DD.NOT.FOUND'	
00281C	0000000C			7312	=F'12'	
002820	00000018			7313	=F'24'	
002824	00000006			7314	=F'6'	
002828	00000004			7315	=F'4'	
00282C	00000007			7316	=F'7'	
002830	C4F84040			7317	=C'D8'	
002834	C4C1E840			7318	=C'DAY'	
002838	C4D6E640			7319	=C'DOW'	
00283C	00000002			7320	=F'2'	
002840	00000100			7321	=F'256'	
002844	00001002			7322	=F'4098'	
002848	00001003			7323	=F'4099'	
00284C	D4C4C3E8			7324	=C'MDCY'	
002850	C4D4C3E8			7325	=C'DMCY'	
002854	C3E8D4C4			7326	=C'CYMD'	
002858	C3E8C4D4			7327	=C'CYDM'	
00285C	0004			7328	=H'4'	
00285E	0000			7329	=X'0000'	
002860	00FF			7330	=X'00FF'	
002862	019C			7331	=P'19'	
002864	F0F2			7332	=C'02'	
002866	365C			7333	=P'365'	
002868	100C			7334	=P'100'	
00286A	400C			7335	=P'400'	
00286C	6B40			7336	=C','	
00286E	F0F1			7337	=C'01'	
002870	F2F9			7338	=C'29'	
002872	099C			7339	=P'99'	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7361	* /******	
				7362	* /* Upper-Lower Case Table	
				7363	* /******	
0028A1	0001020304050607			7364	TRTAB DC 256AL1(*-TRTAB)	
0029A1		02962		7365	ORG TRTAB+C'A'	
002962	8182838485868788			7366	DC C'abcdefghi'	
00296B		02972		7367	ORG TRTAB+C'J'	
002972	9192939495969798			7368	DC C'jklmnopqr'	
00297B		02983		7369	ORG TRTAB+C'S'	
002983	A2A3A4A5A6A7A8A9			7370	DC C'stuvwxyz'	
00298B		029A1		7371	ORG ,	
				7372	* /******	
				7373	* /* Upper-Lower Case Table	
				7374	* /******	
0029A1				7375	TRTBL DC 0XL256'0'	
0029A1	0001020304050607			7376	DC (C'A')AL1(*-TRTBL),9AL1(*-TRTBL-X'40')	
002A6B	CACBCCCDCECFD091			7377	DC (C'J'-(*-TRTBL))AL1(*-TRTBL),9AL1(*-TRTBL-X'40')	
002A7B	DADBDCDDDEDFE0E1			7378	DC (C'S'-(*-TRTBL))AL1(*-TRTBL),8AL1(*-TRTBL-X'40')	
002A8B	EAEBECEDEEEFF0F1			7379	DC (256-(*-TRTBL))AL1(*-TRTBL)	
				7380	* /******	
				7381	* /* Entry codes for IKJCT441	
				7382	* /******	
				7383	* /******	
002AA1	000000			7384	ECNOIMP DC A(TSVNOIMP)	ENTRY CODE FOR RETRIEVE/NOCREATE
002AA4	00000012			7385	ECUPDT DC A(TSVEUPDT)	ENTRY CODE FOR UPDATE/CREATE
002AA8	00000002			7386	* *	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7388 *	*/*****	
				7389 *	/* Messages - Internal	
				7390 *	*/*****	
002AAC	E2A4838385A2A286			7391 MSGDONE	DC C'Successful Request '	
		00013		7392 MSGDONEL	EQU *-MSGDONE	
002ABF	C6C1D3E2C5			7393 MSG4000	DC C'FALSE'	
		00005		7394 MSG4000L	EQU *-MSG4000	
002AC4	E3D9E4C5			7395 MSG4001	DC C'TRUE'	
		00004		7396 MSG4001L	EQU *-MSG4001	
002AC8	E5C1D9F1409596A3			7397 MSG4002	DC C'VAR1 not found'	
		0000E		7398 MSG4002L	EQU *-MSG4002	
002AD6	D59640D7C1D9D4			7399 MSG4004	DC C'No PARM'	
		00007		7400 MSG4004L	EQU *-MSG4004	
002ADD	E6D6D9C440C7E340			7401 MSG4005	DC C'WORD GT 8 bytes'	
		0000F		7402 MSG4005L	EQU *-MSG4005	
002AEC	C3D7D7D340959640			7403 MSG4008	DC C'CPPL no PARM'	
		0000C		7404 MSG4008L	EQU *-MSG4008	
002AF8	E3969640948195A8			7405 MSG4009	DC C'Too many words'	
		0000E		7406 MSG4009L	EQU *-MSG4009	
002B06	E2C5E3E5C1D94095			7407 MSG4010	DC C'SETVAR no content'	
		00011		7408 MSG4010L	EQU *-MSG4010	
002B17	C995A58193898440			7409 MSG4011	DC C'Invalid Function'	
		00010		7410 MSG4011L	EQU *-MSG4011	
002B27	E5C1D9F2409596A3			7411 MSG4012	DC C'VAR2 not updated'	
		00010		7412 MSG4012L	EQU *-MSG4012	
002B37	E5C1D9F140C7E340			7413 MSG4014	DC C'VAR1 GT 256 bytes'	
		00011		7414 MSG4014L	EQU *-MSG4014	
002B48	E5C1D9F240998598			7415 MSG4015	DC C'VAR2 required'	
		0000D		7416 MSG4015L	EQU *-MSG4015	
002B55	E5C1D9F2409596A3			7417 MSG4016	DC C'VAR2 not found'	
		0000E		7418 MSG4016L	EQU *-MSG4016	
002B63	E5C1D9F240C7E340			7419 MSG4017	DC C'VAR2 GT 256 bytes'	
		00011		7420 MSG4017L	EQU *-MSG4017	
002B74	E5C1D9F24095A493			7421 MSG4018	DC C'VAR2 null'	
		00009		7422 MSG4018L	EQU *-MSG4018	
002B7D	E5C1D9F14095A493			7423 MSG4019	DC C'VAR1 null'	
		00009		7424 MSG4019L	EQU *-MSG4019	
002B86	C5D9D9D4E2C74083			7425 MSG4020	DC C'ERRMSG cannot create updt'	
		00019		7426 MSG4020L	EQU *-MSG4020	
002B9F	E5C1D9F140998598			7427 MSG4021	DC C'VAR1 required'	
		0000D		7428 MSG4021L	EQU *-MSG4021	
002BAC	E5C1D9F2409596A3			7429 MSG4022	DC C'VAR2 not required'	
		00011		7430 MSG4022L	EQU *-MSG4022	
002BBD	D985A2A493A340C7			7431 MSG4023	DC C'Result GT 256'	
		0000D		7432 MSG4023L	EQU *-MSG4023	
002BCA	D596408285878995			7433 MSG4030	DC C'No begin ''	
		0000A		7434 MSG4030L	EQU *-MSG4030	
002BD4	D59640859584407F			7435 MSG4031	DC C'No end ''	
		00008		7436 MSG4031L	EQU *-MSG4031	
002BDC	E2A38199A340C7E3			7437 MSG4032	DC C'Start GT End'	
		0000C		7438 MSG4032L	EQU *-MSG4032	
002BE8	C995A58193898440			7439 MSG4033	DC C'Invalid char'	
		0000C		7440 MSG4033L	EQU *-MSG4033	
002BF4	D596A34095A49485			7441 MSG4034	DC C'Not numeric'	
		0000B		7442 MSG4034L	EQU *-MSG4034	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
002BFF	E3969640948195A8			7443	MSG4035	DC	C'Too many digits'
			0000F	7444	MSG4035L	EQU	*-MSG4035
002C0E	D5A4939340A58193			7445	MSG4036	DC	C'Null value not allowed'
			00016	7446	MSG4036L	EQU	*-MSG4036
002C24	D596408285878995			7447	MSG4037	DC	C'No begin QUOTE'
			0000E	7448	MSG4037L	EQU	*-MSG4037
002C32	D5964085958440D8			7449	MSG4038	DC	C'No end QUOTE'
			0000C	7450	MSG4038L	EQU	*-MSG4038
002C3E	D5964085958440D8			7451	MSG4039	DC	C'No end QUOTE found'
			00012	7452	MSG4039L	EQU	*-MSG4039
002C50	C995A58193898440			7453	MSG4040	DC	C'Invalid length'
			0000E	7454	MSG4040L	EQU	*-MSG4040
002C5E	C995A58193898440			7455	MSG4047	DC	C'Invalid DD'
			0000A	7456	MSG4047L	EQU	*-MSG4047
002C68	C995A58193898440			7457	MSG4049	DC	C'Invalid MM'
			0000A	7458	MSG4049L	EQU	*-MSG4049
002C72	C995A58193898440			7459	MSG4050	DC	C'Invalid DD for MM'
			00011	7460	MSG4050L	EQU	*-MSG4050
002C83	C995A58193898440			7461	MSG4051	DC	C'Invalid JJJ'
			0000B	7462	MSG4051L	EQU	*-MSG4051
002C8E	C381959596A34093			7463	MSG4098	DC	C'Cannot link to IKJCT441'
			00017	7464	MSG4098L	EQU	*-MSG4098
002CA5	D4A4A2A34099A495			7465	MSG4099	DC	C'Must run under TSO'
			00012	7466	MSG4099L	EQU	*-MSG4099
002CB7				7467	CL100	DS	CL100
				7468	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7470 *	/* *****/	
				7471 *	/* Month Paramter Table */	
				7472 *	/* *****/	
		02D1B		7473 MMTABLE EQU *		
				7474 *	+----- 2 C	Month Number
				7475 *	+----- 2 C	Days in month
				7476 *	+----- 3 C	Days at start of month
				7477 *	+----- 1 C	Term 0,3,2,5,0,3,5,1,4,6,2,4
				7478 *	+----- 9 C	Monthname
				7479 *		
002D1B	F0F1F3F1F0F0F0F0			7480	DC C'01310000January '	
		00011		7481 MMTBLEN EQU	*-MMTABLE	Length of table entry
002D2C	F0F2F2F8F0F3F1F3			7482	DC C'02280313February '	
002D3D	F0F3F3F1F0F5F9F2			7483	DC C'03310592March '	
002D4E	F0F4F3F0F0F9F0F5			7484	DC C'04300905April '	
002D5F	F0F5F3F1F1F2F0F0			7485	DC C'05311200May '	
002D70	F0F6F3F0F1F5F1F3			7486	DC C'06301513June '	
002D81	F0F7F3F1F1F8F1F5			7487	DC C'07311815July '	
002D92	F0F8F3F1F2F1F2F1			7488	DC C'08312121August '	
002DA3	F0F9F3F0F2F4F3F4			7489	DC C'09302434September '	
002DB4	F1F0F3F1F2F7F3F6			7490	DC C'10312736October '	
002DC5	F1F1F3F0F3F0F4F2			7491	DC C'11303042November '	
002DD6	F1F2F3F1F3F3F4F4			7492	DC C'12313344December '	
		02DE7		7493 MMTABLEE EQU *		
		0000C		7494 MMTBLN EQU	(*-MMTABLE)/MMTBLEN	Number of table entries
				7495 *MMTBLN EQU	(MMTABLEE)/MMTBLEN	Number of table entries
				7496 *		
				7497 *	/* *****/	
				7498 *	/* Day of Week Table */	
				7499 *	/* *****/	
		02DE7		7500 DOWTABLE EQU *		
				7501 *	+----- 1 C	Weekday Number
				7502 *	+----- 9 C	Weekday name
				7503 *		
002DE7	F0E2A4958481A840			7504	DC C'0Sunday '	
		0000A		7505 DOWTBLL EQU	*-DOWTABLE	Length of table entry
002DF1	F1D496958481A840			7506	DC C'1Monday '	
002DFB	F2E3A485A28481A8			7507	DC C'2Tuesday '	
002E05	F3E685849585A284			7508	DC C'3Wednesday '	
002E0F	F4E388A499A28481			7509	DC C'4Thursday '	
002E19	F5C699898481A840			7510	DC C'5Friday '	
002E23	F6E281A3A4998481			7511	DC C'6Saturday '	
		00007		7512 DOWTBLN EQU	(*-DOWTABLE)/DOWTBLL	Number of table entries
				7513 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				7515	* /* *****/	
				7516	* /* Translate table reverse it! */	
				7517	* /* *****/	
				7518	* /* abcdefghi to zyxwvutsr */	
				7519	* /* jklmnopqr to qponmlkji */	
				7520	* /* stuvwxyz to hgfedcba */	
				7521	* /* ABCDEFGHI to ZYXWVUTSR */	
				7522	* /* jklmnopqr to qponmlkji */	
				7523	* /* stuvwxyz to hgfedcba */	
				7524	* /* 0123456789 to 9876543210 */	
				7525	* /* *****/	
002E2D	4040404040404040			7526	TREVRs DC 256C' ' 256 blanks	
002F2D		02EAE		7527	ORG TREVRs+C'a'	
002EAE	A9A8A7A6A5A4A3A2			7528	DC C'zyxwvutsr'	
002EB7		02EBE		7529	ORG TREVRs+C'j'	
002EBE	9897969594939291			7530	DC C'qponmlkji'	
002EC7		02ECF		7531	ORG TREVRs+C's'	
002ECF	8887868584838281			7532	DC C'hgfedcba'	
002ED7		02EEE		7533	ORG TREVRs+C'A'	
002EEE	E9E8E7E6E5E4E3E2			7534	DC C'ZYXWVUTSR' tr string(5),TREVRs	
002EF7		02EFE		7535	ORG TREVRs+C'J'	
002EFE	D8D7D6D5D4D3D2D1			7536	DC C'QPONMLKJI'	
002F07		02F0F		7537	ORG TREVRs+C'S'	
002F0F	C8C7C6C5C4C3C2C1			7538	DC C'HGFEDCBA'	
002F17		02F1D		7539	ORG TREVRs+C'0'	
002F1D	F9F8F7F6F5F4F3F2			7540	DC C'9876543210'	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					7542	*	*/*****	
					7543	*	/* Instructions used for EXECUTES	
					7544	*	*/*****	
002F27	00							
002F28	D200	D4E1	8000	004E1	00000	7545	MVCVAR1# MVC	VAL1(0),0(R8) Save VAR1 value
				00006	7546	MVCVAR1L EQU	*-MVCVAR1#	
					7547	*		
002F2E	D200	D6E4	8000	006E4	00000	7548	MVCVARW# MVC	VARWORK(0),0(R8) Save VARWORK value
				00006	7549	MVCVARWL EQU	*-MVCVARW#	
					7550	*		
002F34	D600	2000	DBBA	00000	00BBA	7551	UPPERCS# OC	0(0,R2),BLANKS EBCDIC Lower-Upper Case Translate
				00006	7552	UPPERCSL EQU	*-UPPERCS#	
					7553	*		
002F3A	D200	6000	8000	00000	00000	7554	MVCR8\$6# MVC	0(0,R6),0(R8) MVC R8 to R6
				00006	7555	MVCR8\$6L EQU	*-MVCR8\$6#	
					7556	*		
002F40	DC00	2000	C9A1	00000	029A1	7557	EXTRLC# TR	0(0,R2),TRTBL EBCDIC Upper-Lower Case Translate
				00006	7558	EXTRLCL EQU	*-EXTRLC#	
					7559	*		
002F46	D500	2000	D7E4	00000	007E4	7560	SRCHCLC# CLC	0(0,R2),SRCHSTR
				00006	7561	SRCHCLCL EQU	*-SRCHCLC#	
					7562	*		
					7563	*CHKDDN# CLC	0(0,R8),TIOEDDNM DDNAME = TIOT entry?	
					7564	*CHKDDNL EQU	*-CHKDDN#	
					7565	*		
002F4C	DC00	D6E4	CE2D	006E4	02E2D	7566	REVR\$# TR	VARWORK(0),TREVR\$ Reverse letters and numbers
				00006	7567	REVRSL EQU	*-REVR\$#	
					7568	*		

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

7570 * * /*****

7571 * * /* IKJCT441 Entry Code Values */

7572 * * /*****

00001 7573 TSVERETR EQU 1 Retrieve/Create Request

00002 7574 TSVEUPDT EQU 2 Update/Create Request

00003 7575 TSVELOC EQU 3 Locate ALL Request (uses TOKEN)

00012 7576 TSVNOIMP EQU 18 Retrieve/Nocreate Request

00014 7577 TSVRNOM EQU 20 No more variables to return (locate)

7578 * * /*****

7579 * * /* Register Equates */

7580 * * /*****

7581 PRINT GEN

7582 YREGS

00000 7584+R0 EQU 0 00060000

00001 7585+R1 EQU 1 00070000

00002 7586+R2 EQU 2 00080000

00003 7587+R3 EQU 3 00090000

00004 7588+R4 EQU 4 00100000

00005 7589+R5 EQU 5 00110000

00006 7590+R6 EQU 6 00120000

00007 7591+R7 EQU 7 00130000

00008 7592+R8 EQU 8 00140000

00009 7593+R9 EQU 9 00150000

0000A 7594+R10 EQU 10 00160000

0000B 7595+R11 EQU 11 00170000

0000C 7596+R12 EQU 12 00180000

0000D 7597+R13 EQU 13 00190000

0000E 7598+R14 EQU 14 00200000

0000F 7599+R15 EQU 15 00210000

7601 PRINT NOGEN

7602 * *

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
000000				7604	MPT	DSECT	Month Parameter Table
000000				7605	MPTMM	DS CL2	Month
000002				7606	MPTMMAXD	DS CL2	Month Max Days
000004				7607	MPTDAT@1	DS CL3	Days before 1st of Month
000007				7608	MPTDWTRM	DS CL1	DOW Term (0,3,2,5,0,3,5,1,4,6,2,4)
			00008	7609	MPTMNB	EQU *	Start of Month Name
000008				7610	MPTMNAME	DS CL9	Month Name
			00009	7611	MPTMNL	EQU *-MPTMNAME	Length of Month Name
			00011	7612	MPTLE	EQU *-MPT	Length of Entry

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
-----	--------	------	-------	-------	------	--------	-----------	-------------------------

				7614		IHAPSA	Prefixed Save Area	
--	--	--	--	------	--	--------	--------------------	--

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

8083 CVT DSECT=YES,LIST=YES Communication Vector Table

LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT ASM 0201 16.49 04/12/20

9083 IKJCPPL Command Processor Parameter List

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
-----	--------	------	-------	-------	------	--------	-----------	-------------------------

					9095	IKJT	IKJT DSECT=YES,LIST=YES Task Control Block	
--	--	--	--	--	------	------	--	--

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	16.49	04/12/20
000000					9932	TIOT	DSECT				
					9933		IEFTIOT1	Task IO Table			

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
000000				10079	JFCB	DSECT	
				10080		IEFJFCBN LIST=YES	Job File Control BLock

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
-----	--------	------	-------	-------	------	--------	-----------	-------------------------

				10700		IHAASCB	Address Space Control Block	
--	--	--	--	-------	--	---------	-----------------------------	--

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
				10934	*	*/*****	*/
				10935	*	/* Working Storage	*/
				10936	*	*/*****	*/
000000				10937	WSDSECT	DSECT	
		000000		10938	WSAREA	EQU *	Working Storage START
000000				10939	*	*	
				10940	SAVEAREA	DS 18F	Savearea
				10941	**	DS F	+00 Unused
				10942	**	DS F	+04 Backward Savearea pointer
				10943	**	DS F	+08 Forward Savearea pointer
				10944	**	DS F	+12 R14 contents
				10945	**	DS F	+16 R15 contents
				10946	**	DS F	+20 R0 contents
				10947	**	DS F	+24 R1 contents
				10948	**	DS F	+28 R2 contents
				10949	**	DS F	+32 R3 contents
				10950	**	DS F	+36 R4 contents
				10951	**	DS F	+40 R5 contents
				10952	**	DS F	+44 R6 contents
				10953	**	DS F	+48 R7 contents
				10954	**	DS F	+52 R8 contents
				10955	**	DS F	+56 R9 contents
				10956	**	DS F	+60 R10 contents
				10957	**	DS F	+64 R11 contents
				10958	**	DS F	+68 R12 contents
000048				10959	SAVER15	DS F	Save R15 for exit
00004C				10960	RC#441	DS F	Save R15 for link IKJCT441
000050				10961	MYR1	DS F	My PARM address on entry
000054				10962	MYR4	DS F	My Starting Parm's address
000058				10963	MYR6	DS F	My Starting Parm's length
00005C				10964	R7HOLD	DS F	R7 Hold area
000060				10965	R8HOLD	DS F	R8 Hold area
				10966	*		
000064				10967	NEWSEG	DS CL1	New dataset segment switch
000065				10968	KSEGCHAR	DS PL2	Segment character count
000067				10969	KSEGCNT	DS PL2	Segment count
				10970	*		
00006C				10971	SAVER1	DS F	Save R1
000070				10972	SAVER5	DS F	Save R5
000074				10973	SAVER7R8	DS 2F	Save R7,R8 for PUTLINE Subroutine
00007C				10974	MYUPT	DS F	User Profile Table Address
000080				10975	MYECT	DS F	Env Control Table Address
000084				10976	MYECB	DS F	Env Control Block for PUTLINE
000088				10977	IOPLADS	DS 4F	I/O parameter block
				10978	PUTBLK	PUTLINE MF=L	List Form of PUTLINE
0000A4				10983	OUTTXT	DS H	Len of line data (incl header)
0000A6				10984		DS H	reserved
0000A8				10985	INFOLINE	DS CL300	Output data area
		00130		10986	OUTTXTL	EQU *-OUTTXT	
0001D4				10987	NNN	DS PL2	Variable counter
		00080		10988	TSO	EQU X'80'	
				10989	*		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
0001D8				10991	WORKAREA DS	F	Workarea
0001DC				10992	UNPKAREA DS	CL16	Workarea for unpacking data
0001EC				10993	TOKEN DS	F	TOKEN (UNUSED HERE) Init to ZERO
				10994	* - - - -		AND NEVER CHANGED BY THE CALLER
				10995	* DS	0D	
0001F0				10996	SRCHFND DS	F	Search String Found Position
0001F4				10997	SRCHBEG DS	F	Search String BEG Pos Length
0001F8				10998	SRCHENDL DS	F	Search String END Pos Length
0001FC				10999	SRCHSTRL DS	F	Search String Length
000200				11000	SRCHST2L DS	F	Search String Length 2
000204				11001	SRCHS DS	F	Search Starting Position
000208				11002	SRCHE DS	F	Search Ending Position
00020C				11003	WORDL DS	F	Number of WORDS
000210				11004	MAXWORDS DS	F	Max Words
000218				11005	DW DS	D	Workarea DW
000220				11006	FW DS	F	Workarea FW
000224				11007	DC#TRM DS	PL1	DOW Term
000225				11008	DC#JJJ DS	PL2	Julian Day
000227				11009	DC#YR4 DS	PL4	CCYY / 4
00022B				11010	DC#YR100 DS	PL4	CCYY / 100
00022F				11011	DC#YR400 DS	PL4	CCYY / 400
000233				11012	DC#DOW DS	PL4	Day of Week
000237				11013	DC#DD DS	PL2	DD
000239				11014	DC#LEAP DS	PL1	Leap Year (0-No, 1-Yes)
00023A				11015	DC#YDAYS DS	PL2	Days in Year (365 or 366)
00023C				11016	DCCCCYY DS	CL5	0CCYY character area
000241				11017	DCDDD DS	CL3	0DD character area
000244				11018	DCMMM DS	CL3	0MM character area
000247				11019	DCJJJ DS	CL3	JJJ character area
00024A				11020	DCCDOW DS	CL1	Day of Week character area
00024B				11021	DCCWDAY DS	CL9	Weekday character area
000254				11022	PKJJJ DS	PL3 060	Workarea pack JJJ
000257				11023	PKACCUM DS	PL3	Workarea
				11024	* *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				11026 *	/* *****	
				11027 *	/* IKJCT441 Parm Address List	
				11028 *	/* *****	
00025C				11029 IKJCT441	DS F	Entry point for IKJCT441
000260				11030 CT441LST	DS 0F	
000260				11031 CT441P1	DS A	Entry Code
000264				11032 CT441P2	DS A	VAR NAME POINTER
000268				11033 CT441P3	DS A	VAR NAME LENGTH
00026C				11034 CT441P4	DS A	VAR VALUE POINTER
000270				11035 CT441P5	DS A	VAR VALUE LENGTH
000274				11036 CT441P6	DS A	Token
				11037 *		
				11038 *	/* *****	
				11039 *	/* IKJCT441 Parm List variable - VAR1	
				11040 *	/* *****	
000278 00000008				11041 NAML1	DC A(L'NAM1)	VAR NAME LENGTH
00027C 000004D0				11042 NAMP1	DC A(NAM1)	VAR NAME POINTER
000280 00000100				11043 VALL1	DC A(L'VAL1)	VAR VALUE LENGTH
000284 000004E1				11044 VALP1	DC A(VAL1)	VAR VALUE POINTER
				11045 *		
				11046 *	/* *****	
				11047 *	/* IKJCT441 Parm List variable - VAR2	
				11048 *	/* *****	
000288 00000008				11049 NAML2	DC A(L'NAM2)	VAR NAME LENGTH
00028C 000004D8				11050 NAMP2	DC A(NAM2)	VAR NAME POINTER
000290 00000100				11051 VALL2	DC A(L'VAL2)	VAR VALUE LENGTH
000294 000005E3				11052 VALP2	DC A(VAL2)	VAR VALUE POINTER
				11053 *		
				11054 *	/* *****	
				11055 *	/* IKJCT441 Parm List variable - VAR3	
				11056 *	/* *****	
000298				11057 NAM3	DS CL9	VAR NAME
0002A4				11058 NAML3	DS F	VAR NAME LENGTH
0002A8				11059 NAMP3	DS F	VAR NAME POINTER
0002AC				11060 VAL3	DS CL256	VAR VALUE
0003AC				11061 VALL3	DS F	VAR VALUE LENGTH
0003B0				11062 VALP3	DS F	VAR VALUE POINTER
				11063 *		
				11064 *	/* *****	
				11065 *	/* IKJCT441 Parm List variable - VAR4	
				11066 *	/* *****	
0003B4				11067 NAM4	DS CL11	VAR NAME
0003C0				11068 NAML4	DS F	VAR NAME LENGTH
0003C4				11069 NAMP4	DS F	VAR NAME POINTER
0003C8				11070 VAL4	DS CL256	VAR VALUE
0004C8				11071 VALL4	DS F	VAR VALUE LENGTH
0004CC				11072 VALP4	DS F	VAR VALUE POINTER
				11073 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 16.49 04/12/20
				11075 *	/* *****/	
				11076 *	/* Variable names and values area */	
				11077 *	/* *****/	
		004D0		11078 VARBLK	EQU *	
0004D0				11079 NAM1	DS CL8	VAR NAME 1
0004D8				11080 NAM2	DS CL8	VAR NAME 2
0004E0				11081 VAL1Q	DS CL1	VAR VALUE 1 QUOTE
0004E1				11082 VAL1	DS CL256	VAR VALUE 1
0005E1				11083	DS CL1	VAR VALUE 1 QUOTE
0005E2				11084 VAL2Q	DS CL1	VAR VALUE 2 QUOTE
0005E3				11085 VAL2	DS CL256	VAR VALUE 2
0006E3				11086	DS CL1	VAR VALUE 2 QUOTE
0006E4				11087 VARWORK	DS CL256	VAR WORK
0007E4				11088 SRCHSTR	DS CL256	Search String
0008E4				11089 SRCHST2	DS CL256	Search String 2
0009E4				11090 SRCHBEG	DS CL3	Search String BEG Pos
0009E7				11091 SRCHEND	DS CL3	Search String END Pos
0009EA				11092 DELIM	DS CL1	Delimiter 1-byte
0009EB				11093 ERRMSG	DS CL30	Error Message Text
000A09				11094 VARSET1	DS CL256	
		00639		11095 VARBLKL	EQU *-VARBLK	
				11096 *	/* *****/	
				11097 *	/* Block to store each parsed WORD */	
				11098 *	/* - All words are of equal length */	
				11099 *	/* *****/	
		00B09		11101 WORDBLK	EQU *	
000B09				11102 FUNCT	DS CL8	Function
		00008		11103 WORDLEN	EQU (*-FUNCT)	WORD Length, words are equal len
000B11				11104 VAR1	DS CL8	Symbolic Variable name 1
000B19				11105 VAR2	DS CL8	Symbolic Variable name 2
000B21				11106 SQUOTE	DS CL8	Quotes surrounding variable value
		00020		11107 WORDBLKK	EQU *-WORDBLK	
		00004		11108 MAXWD	EQU WORDBLKK/WORDLEN	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
					11110	*	*	
					11111	*	/* *****	*/
					11112	*	/* Block to store each parsed WORD length	*/
					11113	*	/* - Must be in same order as WORDBLK	*/
					11114	*	/* *****	*/
000B2C					11115	WORDBLKL	DS 0F	
000B2C					11116	FUNCTL	DS F	Length of FUNCT
000B30					11117	VAR1L	DS F	Length of VAR1
000B34					11118	VAR2L	DS F	Length of VAR2
000B38					11119	SQOUTEL	DS F	Length of QOUTE keyword
000B3C					11120	VARSET1L	DS F	Length of VARSET
					11121	*		
000B40					11122	CHKV1NL	DS C	Test VAR1 for NULL flag
000B41					11123	CHKV2NL	DS C	Test VAR2 for NULL flag
000B42					11124	DDD	DS CL120	
000BBA					11125	BLANKS	DS CL256	Blanks used for Case Traslation
					11126	*	*	
					11127	*	/* *****	*/
					11128	*	/* EX instructions	*/
					11129	*	/* *****	*/
000CBA	D200	D6E4	8000	006E4	00000	11130	MVCVARWK MVC VARWORK(0),0(R8)	Save VARWORK value
					11131	*		
000CC0	D200	D4E1	8000	004E1	00000	11132	MVCVAR1 MVC VAL1(0),0(R8)	Save VAR1 value
					11133	*		
000CC6	D600	2000	DBBA	00000	00BBA	11134	UPPERCS OC 0(0,R2),BLANKS	EBCDIC Lower-Upper Case Translate
					11135	*UPPERCS	OC 0(0,R2),=256X'40'	EBCDIC Lower-Upper Case Translate
					11136	*		
000CCC	D200	6000	8000	00000	00000	11137	MVCR8\$6 MVC 0(0,R6),0(R8)	MVC R8 to R6
					11138	*		
000CD2	DC00	2000	C9A1	00000	029A1	11139	EXTRLC TR 0(0,R2),TRTBL	EBCDIC Upper-Lower Case Translate
					11140	*		
000CD8	D500	2000	D7E4	00000	007E4	11141	SRCHCLC CLC 0(0,R2),SRCHSTR	
					11142	*		
					11143	*CHKDDN	CLC 0(0,R8),TIOEDDNM	DDNAME = TIOT entry?
					11144	*		
000CDE	DC00	D6E4	CE2D	006E4	02E2D	11145	REVRS TR VARWORK(0),TREVRS	Reverse letters and numbers
					11146	*	*	
					11147	*	/* *****	*/
					11148	*	/* End of Working Storage	*/
					11149	*	/* *****	*/
000CE8					11150		DS 0D	
			00CE8		11151	WSAREAE	EQU *	Working Storage END
			00CE8		11152	WSAREAL	EQU WSAREAE-WSAREA	Working Storage Area Length

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 16.49 04/12/20
-----	--------	------	-------	-------	------	--------	-----------	-------------------------

11154 *

11155

END

PAGE 232

ASM 0201 16.49 04/12/20

0001	0001	0C	001E44
------	------	----	--------

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
ASCB	00001	00000000	10717	02644	
ASCBTSB	00004	0000003C	10771	02644	
BLANKS	00256	00000BBA	11125	02715	02716 02716 02716 04481 04724 06925 07551 11134
BUMPEND\$	00001	0000068E	03101	03095	
BUMPV\$	00001	00000682	03096	03089	
BUMPVW\$	00001	00000686	03098	03093	
BYPSV1NL	00001	00001ABE	05967	05963	
BYPSV2NL	00001	00001BBE	06098	06094	
CALCDSNL	00001	00000E8A	04185	04190	
CALCEXIT	00001	00000E9E	04191	04187	
CHKDDMAX	00001	0000220A	06796	06790	06792
CHKDDN	00006	00000E5E	04169	04161	
CHKFIR\$T	00001	000006A2	03121	03116	
CHKMM	00001	000021BC	06773	06777	
CHKSEG	00001	000008F4	03420	03379	03381 03383 03385 03397 03401 03405 03419
CHKV1NL	00001	00000B40	11122	05913	05962
CHKV2NL	00001	00000B41	11123	06057	06093
CHK4VAR	00001	00001640	05439	05446	05465
CONT00L	00001	0000036A	02886	02874	
CPPL	00001	00000000	09088	02751	
CPPL\$IN	00001	0000020A	02750	02742	
CPPLECT	00004	0000000C	09092	02753	
CPPLUPT	00004	00000004	09090	02752	
CR8VAR	00001	00001682	05467	05463	
CR8VARS	00001	0000163A	05436	05471	
CR8VCHK	00001	00001670	05460	05441	
CTRBEG	00001	00000B34	03747	03742	
CTREND	00001	00000B5E	03767	03760	
CT441LST	00004	00000260	11030	05945	06083 06147 06207 06259 06313 06901 06954
CT441P1	00004	00000260	11031	05930	06068 06131 06192 06244 06298 06886 06939
CT441P2	00004	00000264	11032	05932	06070 06133 06194 06246 06300 06888 06941
CT441P3	00004	00000268	11033	05934	06072 06135 06196 06248 06302 06890 06943
CT441P4	00004	0000026C	11034	05936	06074 06137 06198 06250 06304 06892 06945
CT441P5	00004	00000270	11035	05938	06076 06139 06200 06252 06306 06894 06947
CT441P6	00004	00000274	11036	05940	05941 06078 06079 06141 06142 06202 06203 06254 06255 06308 06309 06896 06897 06949
				06950	
CUTIL00	00001	00000000	02561	02562	
CVT	00001	00000000	08136	04144	04229
CVTBRET	00002	00000052	08194	08761	
CVTPTR	00001	00000010	08132	04143	04228
CVTTCBP	00004	00000000	08139	04145	04230
CVTXTNT1	00001	00000000	08944	08371	
CVTXTNT2	00001	00000000	08953	08442	
CVT4MS1	00001	00000010	08213	08220	
CVT6DAT	00001	00000002	08216	08220	
DC#DD	00002	00000237	11013	06515	06532 06805 06806
DC#DOW	00004	00000233	11012	06521	06528 06529 06530 06531 06532 06533 06534
DC#JJJ	00002	00000225	11008	06784	06806 06809 06811
DC#LEAP	00001	00000239	11014	04356	04435 06410 06428 06713 06791 06809
DC#TRM	00001	00000224	11007	06531	06730 06785
DC#YDAYS	00002	0000023A	11015	04431	04550 06411 06429
DC#YR100	00004	0000022B	11010	06414	06417 06422 06522 06525 06529
DC#YR4	00004	00000227	11009	06412	06414 06415 06416 06419 06516 06519 06521 06522 06523 06524 06528
DC#YR400	00004	0000022F	11011	06415	06418 06425 06523 06526 06530
DCCCCYY	00005	0000023C	11016	04074	04075 04330 04331 04420 04421 06412 06473 06516 06566 06574 06578 06598 06611 06616

SYMBOL	LEN	VALUE	DEFN	REFERENCES												ASM 0201 16.49 04/12/20					
				06621	06626	06663	06666														
DCCDOW	00001	0000024A	11020	04105	04635	06534	06535	06543													
DCCWDAY	00009	0000024B	11021	04106	04647	04648	06548														
DCDDD	00003	00000241	11017	04095	06463	06515	06599	06613	06618	06623	06628	06660	06732	06733	06787	06793	06797	06805			
DCJJJ	00003	00000247	11019	04076	04091	04537	06575	06579	06669	06811	06812										
DCMMM	00003	00000244	11018	04093	04328	04329	04339	04346	06517	06600	06612	06617	06622	06627	06657	06728	06729	06774			
				06789	06807																
DELIM	00001	000009EA	11092	04901	04978	05032	05039	05105	05112	05180	05187	05210	05267	05280	05426	05440					
DFLTBEG	00001	0000253E	07181	07167																	
DFLTEND	00001	0000259A	07215	07201																	
DFLTFILL	00001	00001388	04949	04939																	
DO#DATE	00001	000011A4	04616	04571	04812																
DO#DOW	00001	000011AC	04627	04575	04816																
DO#DOWDY	00001	000011C2	04639	04573	04814																
DO#DOWFD	00001	00001FDC	06547	06544																	
DO#DOWNM	00001	000011CE	04649	04656																	
DO#DOWSR	00001	00001FCA	06542	06546																	
DO#MDCY	00001	00001144	04581	04817																	
DO#MDCY3	00001	00001184	04595	04583	04585	04587	04589														
DO#MDCY4	00001	00001190	04599	04591	04593																
DO#MDCY9	00001	0000119C	04611	04598	04602																
DODISPLY	00001	00000962	03473	03467																	
DODSNSEG	00001	0000088A	03374	03360																	
DOECHO	00001	0000095A	03470	03457																	
DOPARMS	00001	00000236	02764	02749																	
DOPLADJ	00004	00000974	03478	03476																	
DOUPPER	00001	000009DA	03542	03537																	
DOWTABLE	00001	00002DE7	07500	06540	07505	07512															
DOWTBLL	00001	0000000A	07505	06545	06548	07512															
DOWTBLN	00001	00000007	07512	06541																	
DSNNOFND	00001	00000EA8	04195	04159																	
DW	00008	00000218	11005	02822	02831	02864	02873	03455	03484	04001	04002	04003	04147	04148	04232	04233	04345	04356			
				04358	04431	04432	04941	04944	04947	04950	04952	06237	06238	06291	06292	07173	07176	07179			
				07182	07185	07207	07210	07213	07216	07219											
ECNOIMP	00004	00002AA4	07384	05929	06067																
ECUPDT	00004	00002AA8	07385	06130	06191	06243	06297	06885	06938												
ENDME	00002	00001A02	05846	05681	05685	05689	05693	05697	05701	05705	05709	05713	05718	05722	05726	05730	05734	05738			
				05742	05746	05750	05755	05759	05763	05767	05772	05776	05780	05784	05788	05792	05796	05801			
				05805	05809	05814	05818	05822	05827	05831	05835	05839									
ENDVAR1	00001	00001AF4	05999	05995																	
ERRDONE	00001	000017E2	05679	03062	03136	03486	03513	03549	03800	03827	03849	03880	03914	04116	04201	04247	04270	04662			
				04991	05062	05137	05228	05297	05377	05476	05630	05673									
ERRDONEF	00001	000017EC	05682	03721	03966	04359	04433														
ERRMSG	00030	000009EB	11093	05680	05683	05684	05688	05692	05696	05700	05704	05708	05712	05717	05721	05725	05729	05733			
				05737	05741	05745	05749	05754	05758	05762	05766	05771	05775	05779	05783	05787	05791	05795			
				05800	05804	05808	05813	05817	05821	05826	05830	05834	05838	06183	06183	06184					
ERR4000	00001	000017FC	05686	03163	03214	03216	03220	03224	03227	03239	03241	03245	03249	03252	03264	03266	03270	03274			
				03277	03308	03352	03369	03371	03395	03399	03403	03410	03417	03428	04437	05668					
ERR4001	00001	0000180A	05690	03164	03231	03256	03281	03312	03430	04436											
ERR4002	00001	00001818	05694	05951																	
ERR4004	00001	00001826	05698	02739	02747	02801															
ERR4005	00001	00001834	05702	02806																	
ERR4008	00001	00001842	05706	02758	02762																
ERR4009	00001	00001850	05710	02815																	
ERR4010	00001	0000185E	05715	02840	04267																

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
ERR4011	00001	0000186C	05719	03039 04482 04494 04594 04725 04735 06571 06609	
ERR4012	00001	0000187A	05723	06153	
ERR4014	00001	00001888	05727	05961	
ERR4015	00001	00001896	05731	04141 04863 05321 05520 05527 05538	
ERR4016	00001	000018A4	05735	03630 04870 05328	
ERR4017	00001	000018B2	05739	06092	
ERR4018	00001	000018C0	05743	06097	
ERR4019	00001	000018CE	05747	05966	
ERR4021	00001	000018EA	05756	05920	
ERR4022	00001	000018F8	05760	03158 03192 03303 03340 03452 03937 03991 04225 04294 04386 05653	
ERR4023	00001	00001906	05764	05595 05610	
ERR4030	00001	00001914	05769	06988 07050	
ERR4031	00001	00001922	05773	06998 07062	
ERR4032	00001	00001930	05777	03655 05549	
ERR4033	00001	0000193E	05781	07039 07104	
ERR4034	00001	0000194C	05785	04321 04413 04527 04771 04919 07112 07114 07147 07149	
ERR4035	00001	0000195A	05789	07120 07155	
ERR4036	00001	00001968	05793	07007 07069	
ERR4037	00001	00001976	05798	05989	
ERR4038	00001	00001984	05802	06001	
ERR4039	00001	00001992	05806	05998	
ERR4040	00001	000019A0	05811	04312 04404 04520 04764 04891 04893 04957 04959	
ERR4047	00001	000019AE	05815	06788 06794	
ERR4049	00001	000019BC	05819	04343 06778	
ERR4050	00001	000019CA	05824	06798	
ERR4051	00001	000019D8	05828	04551 04553	
ERR4098	00001	000019E6	05832	02636	
ERR4099	00001	000019F4	05836	02646	
EXTRLC	00006	00000CD2	11139	02724 03539	
EXTRLC#	00006	00002F40	07557	02724 07558	
EXTRLCL	00001	00000006	07558	02724	
FFF1	00001	00000314	02853	02850	
FFF2	00001	0000032A	02859	02852	
FFF3	00001	00000334	02863	02858	
FILLEM	00001	000013C0	04977	04969 04980	
FILLLEN	00001	000012F8	04883	04888	
FLCEICOD	00002	00000086	07688	07689	
FLCENPSW	00004	00000058	07667	07669	
FLCEOPSW	00008	00000018	07650	07651	
FLCINPSW	00004	00000078	07680	07682	
FLCIOPSW	00008	00000038	07658	07659	
FLCIPPSW	00008	00000000	07641	07644	
FLCMNPSW	00004	00000070	07676	07679	
FLCMOPSW	00008	00000030	07656	07657	
FLCPICOD	00002	0000008E	07708	07709	
FLCPIILC	00001	0000008D	07702	07707	
FLCPNPSW	00004	00000068	07673	07675	
FLCPOPSW	00008	00000028	07654	07655	
FLCSNPSW	00004	00000060	07670	07672	
FLCSOPSW	00008	00000020	07652	07653	
FLCSVCN	00002	0000008A	07698	07699	
FLCSVILC	00001	00000089	07693	07697	
FLCTIMER	00004	00000050	07664	07665	
FNDMM	00001	00002182	06724	06716	
FNXTBLNK	00001	000002DE	02835	02839	

SYMBOL	LEN	VALUE	DEFN	REFERENCES											ASM 0201 16.49 04/12/20				
FUNCT	00008	00000B09	11102	02818	02895	02897	02899	02901	02903	02905	02907	02909	02911	02913	02915	02917	02919	02921	
				02923	02925	02927	02929	02931	02933	02935	02937	02940	02942	02944	02946	02948	02950	02952	
				02954	02958	02960	02962	02965	02967	02969	02971	02973	02975	02977	02979	02981	02983	02985	
				02987	02989	02991	02993	02995	02997	02999	03001	03003	03005	03007	03009	03011	03014	03016	
				03018	03020	03022	03024	03026	03028	03030	03032	03034	03036	03195	03197	03456	03536	03691	
				03693	03705	03707	03900	04005	04007	04429	04479	04481	04570	04572	04574	04582	04584	04586	
				04588	04590	04592	04596	04600	04722	04724	04811	04813	04815	04968	05013	05051	05086	05126	
				05162	05217	05684	06567	06569	06601	06603	06605	06607	07010	11103					
FW	00004	00000220	11006	04066	04067	04068	04172	04172	04173	04174	06710	06713	06715	06731					
GETLFILL	00001	0000134A	04926	04906															
GETMMJJJ	00001	0000215A	06709	06718															
GETWORDS	00001	0000026C	02799	02885															
GOODDDN	00001	00000E64	04171	04163															
GOTFLEN	00001	0000130C	04889	04887															
GOTJJJX	00001	00002230	06810	06808															
GOTMM	00001	000021D2	06782	06775															
GOTMMX	00001	00002214	06799	06795															
IKJ\$0007	00002	00001E56	06356	06348															
IKJ@0007	00002	00001E3E	06349	06345															
IKJCT441	00004	0000025C	11029	02637	05946	06084	06148	06208	06260	06314	06902	06955							
INCYDM	00001	000020AC	06625	06608															
INCYJ	00001	00002008	06573	06568															
INCYMD	00001	00002096	06620	06606															
INDMCY	00001	00002080	06615	06604															
INFOLINE	00300	000000A8	10985	03058	03488														
INJCY	00001	00002018	06577	06570															
INMDCY	00001	0000206A	06610	06602															
IOPLADS	00004	00000088	10977	06332															
JFCB	00001	00000000	10079	04176															
JFCBDSNM	00044	00000000	10116	04177															
JFCRESRV	00004	0000006C	10562	10587															
KSEGCHAR	00002	00000065	10968	03356	03368	03370	03372	03409	03421	03427									
KSEGCNT	00002	00000067	10969	03357	03362														
LEAPTF	00001	0000105A	04434	04430															
LFILLX	00001	0000138E	04951	04942	04945	04948													
LFILL1	00001	0000136A	04940	04934															
LFILL2	00001	00001374	04943	04936															
LFILL3	00001	0000137E	04946	04938															
LOADBEG	00001	00002470	07105	07103	07121														
LOADBEGA	00001	00002440	07075	07011															
LOADBEG0	00001	00002448	07083	07081															
LOADBEG1	00001	00002458	07096	07094															
LOADDFLT	00001	00000A2A	03619	03612	03614														
LOADEND	00001	000024C6	07140	07156															
LOADENDA	00001	000024B6	07128	07110															
LOADEND0	00001	000024BE	07134	07132															
LOADINSS	00001	00000A48	03634	03622															
LOADSS	00001	00002396	06990	06997															
LOADSSL	00001	000023AC	06996	06989															
LOADSSX	00001	000023B4	07002	06993															
LOADST20	00001	000023D8	07023	07021															
LOADST21	00001	000023E8	07033	07031															
LOADS2	00001	00002410	07054	07061															
LOADS2B	00001	00002400	07047	07038															
LOADS2L	00001	00002426	07060	07051															

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
LOADS2X	00001	0000242E	07064	07057	
LOADVAR2	00001	00000A3A	03626	03615	
LOOKBBB	00001	00001AE2	05993	05997	
LOOPDDN	00001	00000E42	04157	04167	
MAXWD	00001	00000004	11108	02790	
MAXWORDS	00004	00000210	11004	02791	02814
MMDAYS	00001	00000FA4	04338	04342	
MMTABLE	00001	00002D1B	07473	04336	06770 07481 07494
MMTABLEE	00001	00002DE7	07493	06704	
MMTBLEN	00001	00000011	07481	06706	07494
MMTBLN	00001	0000000C	07494	04335	06708 06772
MOVCC	00001	0000210C	06662	06650	
MOVDD	00001	00002102	06659	06648	
MOVJJJ	00001	00002120	06668	06654	
MOVMM	00001	000020F8	06656	06646	
MOVVAR1L	00001	00000304	02848	02857	
MOVXX	00001	00002132	06673	06658	06661 06664 06667 06672
MOVYY	00001	00002116	06665	06652	
MPT	00001	00000000	07604	04337	06705 06771 07612
MPTDAT@1	00003	00000004	07607	06710	06784
MPTDWTRM	00001	00000007	07608	06730	06785
MPTLE	00001	00000011	07612	04341	06776
MPTMM	00002	00000000	07605	04339	06711 06729 06774
MPTMMAXD	00002	00000002	07606	04345	06797
MPTMNAME	00009	00000008	07610	06739	06816 07611
MPTMNL	00001	00000009	07611	06740	06817
MSGDONE	00019	00002AAC	07391	05680	05683 07392
MSGDONEL	00001	00000013	07392	05680	05683
MSG4000	00005	00002ABF	07393	05688	07394
MSG4000L	00001	00000005	07394	05688	
MSG4001	00004	00002AC4	07395	05692	07396
MSG4001L	00001	00000004	07396	05692	
MSG4002	00014	00002AC8	07397	05696	07398
MSG4002L	00001	0000000E	07398	05696	
MSG4004	00007	00002AD6	07399	05700	07400
MSG4004L	00001	00000007	07400	05700	
MSG4005	00015	00002ADD	07401	05704	07402
MSG4005L	00001	0000000F	07402	05704	
MSG4008	00012	00002AEC	07403	05708	07404
MSG4008L	00001	0000000C	07404	05708	
MSG4009	00014	00002AF8	07405	05712	07406
MSG4009L	00001	0000000E	07406	05712	
MSG4010	00017	00002B06	07407	05717	07408
MSG4010L	00001	00000011	07408	05717	
MSG4011	00016	00002B17	07409	05721	07410
MSG4011L	00001	00000010	07410	05721	
MSG4012	00016	00002B27	07411	05725	07412
MSG4012L	00001	00000010	07412	05725	
MSG4014	00017	00002B37	07413	05729	07414
MSG4014L	00001	00000011	07414	05729	
MSG4015	00013	00002B48	07415	05733	07416
MSG4015L	00001	0000000D	07416	05733	
MSG4016	00014	00002B55	07417	05737	07418
MSG4016L	00001	0000000E	07418	05737	
MSG4017	00017	00002B63	07419	05741	07420

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
MSG4017L	00001	00000011	07420	05741	
MSG4018	00009	00002B74	07421	05745 07422	
MSG4018L	00001	00000009	07422	05745	
MSG4019	00009	00002B7D	07423	05749 07424	
MSG4019L	00001	00000009	07424	05749	
MSG4020	00025	00002B86	07425	05754 07426	
MSG4020L	00001	00000019	07426	05754	
MSG4021	00013	00002B9F	07427	05758 07428	
MSG4021L	00001	0000000D	07428	05758	
MSG4022	00017	00002BAC	07429	05762 07430	
MSG4022L	00001	00000011	07430	05762	
MSG4023	00013	00002BBD	07431	05766 07432	
MSG4023L	00001	0000000D	07432	05766	
MSG4030	00010	00002BCA	07433	05771 07434	
MSG4030L	00001	0000000A	07434	05771	
MSG4031	00008	00002BD4	07435	05775 07436	
MSG4031L	00001	00000008	07436	05775	
MSG4032	00012	00002BDC	07437	05779 07438	
MSG4032L	00001	0000000C	07438	05779	
MSG4033	00012	00002BE8	07439	05783 07440	
MSG4033L	00001	0000000C	07440	05783	
MSG4034	00011	00002BF4	07441	05787 07442	
MSG4034L	00001	0000000B	07442	05787	
MSG4035	00015	00002BFF	07443	05791 07444	
MSG4035L	00001	0000000F	07444	05791	
MSG4036	00022	00002C0E	07445	05795 07446	
MSG4036L	00001	00000016	07446	05795	
MSG4037	00014	00002C24	07447	05800 07448	
MSG4037L	00001	0000000E	07448	05800	
MSG4038	00012	00002C32	07449	05804 07450	
MSG4038L	00001	0000000C	07450	05804	
MSG4039	00018	00002C3E	07451	05808 07452	
MSG4039L	00001	00000012	07452	05808	
MSG4040	00014	00002C50	07453	05813 07454	
MSG4040L	00001	0000000E	07454	05813	
MSG4047	00010	00002C5E	07455	05817 07456	
MSG4047L	00001	0000000A	07456	05817	
MSG4049	00010	00002C68	07457	05821 07458	
MSG4049L	00001	0000000A	07458	05821	
MSG4050	00017	00002C72	07459	05826 07460	
MSG4050L	00001	00000011	07460	05826	
MSG4051	00011	00002C83	07461	05830 07462	
MSG4051L	00001	0000000B	07462	05830	
MSG4098	00023	00002C8E	07463	05834 07464	
MSG4098L	00001	00000017	07464	05834	
MSG4099	00018	00002CA5	07465	05838 07466	
MSG4099L	00001	00000012	07466	05838	
MVCPUTL	00006	00000992	03488	03481	
MVCR8\$6	00006	00000CCC	11137	02723 03796 03844 03894 05365 06007 06011	
MVCR8\$6#	00006	00002F3A	07554	02723 07555	
MVCR8\$6L	00001	00000006	07555	02723	
MVCVARW#	00006	00002F2E	07548	02720 05569 07549	
MVCVARWK	00006	00000CBA	11130	02720 03129 03206 03344 03506 03533 03604 04021 04856 05048 05123 05197 05342	
MVCVARWL	00001	00000006	07549	02720	
MVCVAR1	00006	00000CC0	11132	02721 05971	

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20									
MVCVAR1#	00006	00002F28	07545	02721	07546									
MVCVAR1L	00001	00000006	07546	02721										
MYECB	00004	00000084	10976	06335										
MYECT	00004	00000080	10975	02753										
MYR1	00004	00000050	10961	02765										
MYR4	00004	00000054	10962	02766										
MYR6	00004	00000058	10963	02767	02888									
MYSTAMP	00001	0000001C	02581	02592	03058									
MYSTAMPL	00001	00000034	02592	03054	03058									
MYUPT	00004	0000007C	10974	02752	06330									
NAML1	00004	00000278	11041	02658	05925	05933	06229	06231	06852	06880	06927	06932		
NAML2	00004	00000288	11049	02667	06061	06071	06120	06124	06134	06283	06285			
NAML3	00004	000002A4	11058	02676	06187	06195	06233	06247	06287	06301				
NAML4	00004	000003C0	11068	02685	06881	06889	06934	06942						
NAMP1	00004	0000027C	11042	02660	05931									
NAMP2	00004	0000028C	11050	02669	06069	06132								
NAMP3	00004	000002A8	11059	02678	06193	06245	06299							
NAMP4	00004	000003C4	11069	02687	06887	06940								
NAM1	00008	000004D0	11079	02657	02659	05924	06224	06226	06837	06853	06924	11041	11042	
NAM2	00008	000004D8	11080	02666	02668	06060	06119	06123	06276	06280	11049	11050		
NAM3	00009	00000298	11057	02675	02677	06186	06226	06227	06228	06280	06281	06282		
NAM4	00011	000003B4	11067	02684	02686	06851	06853	06924	06925	06926				
NEXTCHAR	00001	00000362	02883	02809	02812									
NEXTWORD	00001	00000294	02810	02803										
NNN	00002	000001D4	10987	05430	06847	06854	06856	06860	06865	06870	06914			
NOADJLY	00001	00002170	06714	06712										
NOMAINS	00001	00001A30	05876	02612										
NXTC	00001	0000213E	06677	06655										
NXTCONT	00001	00000350	02876	02817	02819									
OUTTXT	00002	000000A4	10983	03056	03057	03474	03477	03479	03480	06341	10986			
OVRSTAMP	00002	000000A6	02596	02580										
PABOUT	00001	00000628	03050	02896										
PCENTER	00001	00000B18	03734	02941										
PCENTERX	00001	00000B88	03798	03746	03764									
PCNTC2	00001	00000A80	03665	03658	03661									
PCONCAT	00001	00001598	05313	03025										
PCONCATC	00001	000015EE	05363	05359										
PCOUNT	00001	000009EA	03587	02906	02908	02932	02934	02936	02938					
PCOUNTF	00001	00000AAE	03689	03685										
PCOUNTL	00001	00000A9A	03682	03667	03670	03687	03702							
PCOUNTX	00001	00000AE0	03704	03688	03692	03694	03696	03703						
PCOUNTXY	00001	00000B0E	03717	03712										
PCOUNTXZ	00001	00000B12	03719	03706	03708	03716								
PCTRE	00001	00000B4C	03758	03762										
PCTRL	00001	00000B20	03740	03744										
PDAYSMM	00001	00000F30	04287	02959										
PDAYSMM#	00001	00000F42	04295	04292										
PDAYSMMX	00001	00000FD4	04357	04347										
PDAYSYY	00001	00000FDC	04379	02961	02963									
PDAYSYY#	00001	00000FEE	04387	04384										
PDDCHKV2	00001	00000E1A	04139	04136										
PDDF8	00001	00000E14	04137	04135										
PDD2DSN	00001	00000E02	04131	02953										
PDD2DSNX	00001	00000EBA	04200	04194										
PDMML	00001	00000F52	04300	04305										

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
PDMMLX	00001	00000F66	04306	04302	
PDYYL	00001	00000FFE	04392	04397	
PDYYLX	00001	00001012	04398	04394	
PECHO	00001	00000914	03445	02922	02924
PECHO#	00001	00000926	03453	03450	
PFILL	00001	000012BC	04850	03015	
PGEN#	00001	00000C90	03982	02951	
PGEN##	00001	00000CA2	03992	03989	
PGEN#X	00001	00000DF6	04111	04013	
PIDSNCX	00001	000008FE	03423	03373	03422
PISALPH#	00001	000006FA	03193	03190	
PISALPHA	00001	000006E8	03183	02912	
PISALPHL	00001	00000724	03212	03230	
PISBLANK	00001	000007FC	03296	02918	
PISBLNK#	00001	0000080E	03304	03301	
PISDSN	00001	00000826	03333	02920	
PISDSN#	00001	00000838	03341	03338	
PISDSNC	00001	0000085E	03358	03425	
PISLAPNX	00001	00000760	03228	03218	03222 03226
PISLOWER	00001	000006E8	03184	02914	
PISLOWRL	00001	0000076C	03237	03196	03255
PISLOWRX	00001	000007A8	03253	03243	03247 03251
PISNUM	00001	000006C2	03151	02910	
PISNUM#	00001	000006D4	03159	03156	
PISUPPER	00001	000006E8	03185	02916	
PISUPPRL	00001	000007B4	03262	03198	03280
PISUPPRX	00001	000007F0	03278	03268	03272 03276
PISWHTL	00001	00000812	03306	03310	
PJCHKF	00001	0000107A	04483	04497	
PJCHKFNX	00001	000010A6	04495	04485	04487 04489 04491
PJDATE	00001	000010AE	04500	02966	02974 04493
PJDAY	00001	000010AE	04501	02968	02976
PJDLL	00001	000010C2	04508	04513	
PJDLLX	00001	000010D6	04514	04510	
PJDOW	00001	000010AE	04502	02970	02978
PJMDCY	00001	00001068	04473	02972	02980
PJMDCYX	00001	000011F0	04658	04613	04626 04638 04651 04657
PJOBINF#	00001	00000ED0	04226	04223	
PJOBINFO	00001	00000EBE	04218	02955	
PKJJJ	00003	00000254	11022	04091	04537 04550 04552 06715 06731 06732
PLJBEG	00001	00000BB4	03828	03821	
PLJSTL	00001	00000B98	03819	03823	
PLJUST	00001	00000B90	03813	02943	
PLOWERC	00001	000009B6	03529	02930	
PLOWERCX	00001	000009DE	03546	03541	
PLSTRIP	00001	000013E4	05006	02898	03017
PLSTRIPX	00001	0000143E	05060	05052	
PMCHKF	00001	00001210	04726	04738	
PMCHKFNX	00001	00001234	04736	04728	04730 04732
PMDATE	00001	0000123C	04741	02982	02990 02998 03006 04734
PMDAY	00001	0000123C	04743	02984	02992 03000 03008
PMDCYJ	00001	000011FE	04714	02988	02996 03004 03012
PMDDL	00001	00001250	04752	04757	
PMDDLX	00001	00001264	04758	04754	
PMDOW	00001	0000123C	04745	02986	02994 03002 03010

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
PNOW	00001	00000C90	03984	03035	
PNOWC	00001	00000D4A	04054	04008	
PNOWX	00001	00000DF6	04113	04109	
POSDONE	00001	00002504	07164	07022 07032 07036 07082 07095 07101 07108 07122 07133 07145 07157	
PPAD	00001	000012BC	04851	03037	
PPADX	00001	000013D0	04986	04973	
PREPL	00001	00001694	05512	03029	
PREPLX	00001	00001790	05624	05598	
PREVRS	00001	00000998	03502	02926	
PRJEND	00001	00000BFA	03881	03874	
PRJSTL	00001	00000BE0	03872	03876	
PRJUST	00001	00000BD4	03864	02945 02947	
PRJX	00001	00000C34	03910	03901 03906	
PRJZF	00001	00000C20	03904	03909	
PRSTRIP	00001	00001446	05077	02900 03019	
PRSTRIPX	00001	000014A6	05135	05127	
PSA	00001	00000000	07639	02643 07851 07856	
PSAAOLD	00004	00000224	07788	02643	
PSAIPCDM	00001	0000026C	07857	07856	
PSAIPCRM	00001	00000264	07852	07851	
PSATNEW	00004	00000218	07784	07785	
PSTRIP	00001	000014AE	05153	02902 03021	
PSTRIP\$	00001	00001532	05242	03023	
PSTRIPX	00001	0000152A	05226	05218	
PTDSN	00001	00000C90	03983	03033	
PTDSNC	00001	00000CE8	04019	04006	
PTDSNX	00001	00000DF6	04112	04047	
PTDSN0	00001	00000D14	04034	04030	
PTDSN1	00001	00000D18	04036	04027	
PTRIM\$	00001	0000064A	03076	02904	
PTRIM\$X	00001	000006BA	03134	03124	
PUNSTR	00001	0000160E	05401	03027	
PUNSTRX	00001	00001690	05475	05452 05455 05466	
PUPPERC	00001	000009B6	03528	02928	
PUTBLK	00004	00000098	10980	06337	
PVEXIST	00001	000017A6	05646	03031	
PVEXIST#	00001	000017B8	05657	05651	
PVEXISTX	00001	000017DE	05672	05666	
PWORDBLK	00001	00000C5A	03944	03949	
PWORDFND	00001	00000C6E	03951	03946	
PWORDNXT	00001	00000C76	03954	03959	
PWORDS	00001	00000C40	03930	02949	
PWORDS#	00001	00000C52	03938	03935	
PWORDSX	00001	00000C8A	03964	03950 03960	
PWORD000	00001	00000C62	03947	03956	
PWORD002	00001	00000C7E	03957	03953	
RC#441	00004	0000004C	10960	03628 04868 05326 05525 05661 05949 06087 06151 06210 06262 06316 06905 06958	
REPL	00001	0000172A	05587	05597 05619	
REPLF	00001	00001754	05602	05590	
REPLFL	00001	0000175C	05605	05612	
REPL000	00001	000016EE	05553	05551	
REPL001	00001	000016F4	05555	05552	
REPL002	00001	00001702	05559	05557	
REPL003	00001	00001708	05561	05558	
REVRS	00006	00000CDE	11145	02726 03508	

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20													
				04324 04329 04331 04390 04393 04395 04395 04405 04416 04421 04506 04509 04511 04511 04521														
				04530 04750 04753 04755 04755 04765 04774 04854 04915 04916 04916 05039 05041 05041 05080														
				05081 05112 05114 05121 05187 05189 05189 05204 05205 05210 05212 05216 05245 05280 05282														
				05286 05286 05316 05352 05404 05440 05442 05445 05445 05464 05464 05470 05470 05515 05572														
				05577 05954 05981 05988 05990 05991 05994 05996 06000 06005 06006 06006 06010 06013 06014														
				06014 06018 06329 06330 06334 06357 06373 06375 06377 06377 06574 06575 06578 06579 06611														
				06612 06613 06616 06617 06618 06621 06622 06623 06626 06627 06628 07545 07548 07554 11130														
				11132 11137														
R8HOLD	00004	00000060	10965	04298 04313 04324 04390 04405 04416 04506 04521 04530 04750 04765 04774 04854 05245 05316														
				05404 05515 05981 06005 06013														
S#BDD	00001	00001F14	06459	06450														
S#BD8	00001	00001EF2	06443	04625														
S#BMDCY	00001	000020C8	06643	04612 06678														
S#BMN	00001	00001EF6	06448	06455														
S#CDOW	00001	00001F50	06494	04104 04634 04646														
S#CDWNA	00001	00001F70	06520	06518														
S#CHKV2	00001	00001C4A	06164	03078 03504 03531 03736 03815 03866 04504 04748 05008 05079 05157 05246 05405														
S#CHKV2X	00001	00001C5C	06169	06167														
S#CVARC	00001	000022BC	06874	06863 06868 06873														
S#CVARN	00001	00002248	06835	05454 05469														
S#CVARNX	00001	00002380	06959	06838														
S#CVAR1	00001	000022AA	06869	06855														
S#CVAR2	00001	00002298	06864	06857														
S#GVAR1	00001	00001A3E	05914	03077 03160 03194 03305 03342 03454 03503 03530 03588 03735 03814 03865 03939 04132 04259														
				04296 04388 04503 04747 05007 05078 05154 05243 05314 05402 05513														
S#GVAR1N	00001	00001A3A	05912	03993 04227 04852														
S#GVAR1X	00001	00001B46	06025	05980 06022 06024														
S#GVAR2	00001	00001B50	06058	03627 04867 05018 05091 05167 05253 05273 05325 05412 05524														
S#GVAR2N	00001	00001B4C	06056	05660														
S#ICYJ	00001	00001FE8	06564	04536														
S#ICYJX	00001	00002028	06581	06576 06580														
S#IMDCY	00001	0000202E	06596	04780														
S#IMDCYX	00001	000020C2	06630	06614 06619 06624 06629														
S#ISLPY	00001	00001E8E	06408	04083 04355 04428 04544 04787														
S#ISLPYT	00001	00001EE0	06427	06423														
S#ISLPYX	00001	00001EEC	06430	06420 06426														
S#ISNUM	00001	00001E60	06371	03161 04319 04411 04525 04769 04917 06378														
S#ISNUMF	00001	00001E84	06383	06374 06376														
S#ISNUMX	00001	00001E88	06385	06382														
S#J2MD	00001	00002148	06699	04092 04561														
S#MD2J	00001	000021B0	06765	04802														
S#PUTL	00001	00001DF0	06327	03059 03482														
S#SRCHP	00001	00002386	06977	03640 05542														
S#UERRM	00001	00001C62	06181	05848														
S#UVAR2	00001	00001BC4	06112	03135 03511 03548 03799 03826 03848 03879 03913 04115 04193 04198 04246 04269 04660 04990														
				05061 05136 05227 05296 05376 05629														
S#UV1L	00001	00001CC8	06222	05849														
S#UV1LX	00001	00001D52	06263	06225														
S#UV2L	00001	00001D58	06274	04874 05022 05095 05171 05257 05332 05416 05529 05850														
S#UV2LX	00001	00001DEA	06317	06277 06279														
SAVER15	00004	00000048	10959	05847 05857														
SAVER5	00004	00000070	10972	05915 06026 06059 06100 06113 06155 06165 06170 06182 06212 06223 06264 06275 06318 06328														
				06359 06372 06386 06409 06431 06444 06477 06495 06551 06565 06582 06597 06631 06644 06680														
				06700 06744 06766 06821 06836 06960														
SAVER7R8	00004	00000074	10973	06329 06357														

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20									
SKIPBLK\$	00001	0000066E	03090	03088 03094										
SKIPDLM	00001	0000157A	05285	05281										
SPCVARB	00001	0000018A	02699	02702										
SPCWORDB	00001	0000019E	02708	02711										
SQUOTE	00008	00000B21	11106	05979										
SRCHBEG	00003	000009E4	11090	07087 07173 07176 07179										
SRCHBEG1	00004	000001F4	10997	03596 07118 07165										
SRCHBPX	00001	00002548	07184	07174 07177 07180 07183										
SRCHBPX1	00001	00002534	07178	07169										
SRCHBPX2	00001	0000252A	07175	07171										
SRCHBXT	00001	00002560	07192	07189										
SRCHCLC	00006	00000CD8	11141	02725 03683 05588										
SRCHCLC#	00006	00002F46	07560	02725 07561										
SRCHCLCL	00001	00000006	07561	02725										
SRCHE	00004	00000208	11002	03648 05548 05556 05560 07220 07224										
SRCHEND	00003	000009E7	11091	07138 07207 07210 07213										
SRCHENDL	00004	000001F8	10998	03597 07153 07199										
SRCHEPKX	00001	000025A4	07218	07208 07211 07214 07217										
SRCHEPK1	00001	00002590	07212	07203										
SRCHEPK2	00001	00002586	07209	07205										
SRCHEXT	00001	000025BA	07225	07223										
SRCHFND	00004	000001F0	10996	03593 03690 03711 03714 03718										
SRCHS	00004	00000204	11001	03647 05548 05550 05554 05567 07186 07191										
SRCHSTR	00256	000007E4	11088	06981 07003 07560 11141										
SRCHSTR1	00004	000001FC	10999	03594 03649 03681 05578 05614 05616 05617 07005										
SRCHST2	00256	000008E4	11089	05603 07024 07065										
SRCHST2L	00004	00000200	11000	03595 05604 07067										
STRIPALL	00001	00001562	05279	05287										
STRIP1	00001	0000140C	05038	05014 05042										
STRIP1L	00001	000014D6	05186	05163 05190										
STRIP1M	00001	0000141E	05046	05040										
STRIP1MM	00001	000014EC	05195	05188										
STRIPR	00001	00001472	05111	05087 05115										
STRIPR1	00001	00001482	05119	05113										
STRIPR1M	00001	00001510	05214	05191 05211										
STRIPR1R	00001	00001500	05209	05213										
SV1CHKQB	00001	000002F2	02841	02837										
SV1NOQ	00001	00000300	02846	02843										
TCB	00001	00000020	09168	04149 04234 09854										
TCBDARPN	00001	00000040	09494	09496										
TCBDARTN	00001	00000080	09491	09493										
TCBFX	00001	00000000	09156	09163										
TCBMNLEN	00001	00000128	09854	09925										
TCBPXLEN	00001	00000020	09163	09925										
TCBTIO	00004	0000002C	09181	04150 04235										
TCBXTNT2	00001	00000000	09859	09923										
TCBX2LEN	00001	00000020	09923	09925										
TIOCNJOB	00008	00000000	09950	04241										
TIOEJFCB	00003	00000024	10017	04173										
TIOELNGH	00001	00000018	09973	04158 04165										
TIOENTRY	00001	00000018	09972	04153 04154 04238 04239										
TIOT	00001	00000000	09932	04152 04237										
TOKEN	00004	000001EC	10993	02653 05939 06077 06140 06201 06253 06307 06895 06948										
TREVRS	00001	00002E2D	07526	07527 07529 07531 07533 07535 07537 07539 07566 11145										
TRIML\$	00001	00000658	03084	03100										

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20														
TRTAB	00001	000028A1	07364	07364 07365 07367 07369															
TRTBL	00256	000029A1	07375	07376 07376 07377 07377 07377 07378 07378 07378 07379 07379 07557 11139															
TSVEUPDT	00001	00000002	07574	07385															
TSVNOIMP	00001	00000012	07576	07384															
UNPKAREA	00016	000001DC	10992	04003 04010 04038 04040 04041 04043 04055 04057 04059 04061 04063 04064 04065 04066 04068															
				04069 04071 04073 04075 04076															
UPPERCS	00006	00000CC6	11134	02722 02778 03209 03543															
UPPERCS#	00006	00002F34	07551	02722 07552															
UPPERCSL	00001	00000006	07552	02722															
USECONT	00001	00001BFC	06126	06121															
USEVAR1	00001	00001BEA	06122	06117															
VALL1	00004	00000280	11043	02662 03461 03465 03472 03713 03748 03754 03777 03784 03824 03832 03846 03877 03887 03902															
				03911 04011 04046 04108 04134 04138 04155 04242 04263 05550 05556 05560 05937 05953 06017															
				06125 06236 07188 07222 07224															
VALL2	00004	00000290	11051	02671 03621 03636 04881 04896 05027 05100 05176 05262 05337 05421 05536 05544 05665 06075															
				06091 06096 06115 06125 06138 06290															
VALL3	00004	000003AC	11061	02680 06185 06199 06235 06251 06289 06305															
VALL4	00004	000004C8	11071	02689 06846 06893 06920 06946															
VALP1	00004	00000284	11044	02664 03755 03778 03892 05121 05216 05935 05954 06018															
VALP2	00004	00000294	11052	02673 03635 04880 04900 04915 04932 04988 05026 05058 05099 05133 05175 05224 05261 05294															
				05336 05374 05420 05535 05543 05627 06073 06136															
VALP3	00004	000003B0	11062	02682 06197 06249 06303															
VALP4	00004	000004CC	11072	02691 06891 06944															
VAL1	00256	000004E1	11082	02661 02663 03471 05203 06009 07545 11043 11044 11132															
VAL1Q	00001	000004E0	11081	03459 03460 03464															
VAL2	00256	000005E3	11085	02670 02672 03620 06114 11051 11052															
VAL3	00256	000002AC	11060	02679 02681 06183 06238 06239 06292 06293															
VAL4	00256	000003C8	11070	02688 02690 06845 06914 06915															
VARBLK	00001	000004D0	11078	02697 11095															
VARBLKL	00001	00000639	11095	02698															
VARSET1	00256	00000A09	11094	02832 02865 04261															
VARSET1L	00004	00000B3C	11120	02728 02728 02833 02834 02834 02847 02860 02862 02866 04263 04265															
VARWORK	00256	000006E4	11087	03079 03112 03122 03208 03347 03535 03607 03652 03790 03838 03886 03903 04010 04024 04055															
				04056 04057 04058 04059 04060 04061 04065 04070 04071 04072 04073 04093 04094 04095 04105															
				04106 04177 04184 04196 04241 04261 04565 04806 04966 04987 05057 05132 05203 05223 05277															
				05293 05350 05373 05437 05573 05626 06004 06010 06019 06020 06020 06020 06114 06845 07548															
				07566 11130 11145															
VAR1	00008	00000B11	11104	05658 05919 05924 06123															
VAR1L	00004	00000B30	11117	05659 05925 06124															
VAR2	00008	00000B19	11105	03155 03157 03189 03191 03300 03302 03337 03339 03449 03451 03611 03613 03934 03936 03988															
				03990 04114 04140 04222 04224 04244 04268 04291 04293 04383 04385 04862 04989 05059 05134															
				05225 05295 05320 05375 05519 05628 05650 05652 05658 06021 06023 06060 06116 06119 06166															
				06168 06278															
VAR2L	00004	00000B34	11118	05659 06061 06120															
VFINDNO	00001	000017DA	05667	05663															
WORDBLK	00001	00000B09	11101	02706 02794 11107															
WORDBLKK	00001	00000020	11107	02707 11108															
WORDBLKL	00004	00000B2C	11115	02798															
WORDL	00004	0000020C	11003	02793 02805															
WORDLEN	00001	00000008	11103	02792 11108															
WORKAREA	00004	000001D8	10991	04000 04001 04039 04040 04062 04063															
WSAREA	00001	00000000	10938	11152															
WSAREAE	00001	00000CE8	11151	11152															
WSAREAL	00001	00000CE8	11152	02608 05867															
WSDSECT	00001	00000000	10937	02621															

CROSS-REFERENCE

PAGE 247

SYMBOL	LEN	VALUE	DEFN	REFERENCES
--------	-----	-------	------	------------

ASM 0201 16.49 04/12/20

YMM	DAYS	00001	00000FBA	04344	04340
-----	------	-------	----------	-------	-------

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
=C 'SETVAR	'				
	00008	000025C0	07234	02818	
=C 'ABOUT	'				
	00008	000025C8	07235	02895	
=C 'LTRIM	'				
	00008	000025D0	07236	02897 05013 05051	
=C 'RTRIM	'				
	00008	000025D8	07237	02899 05086 05126	
=C 'TRIM	'				
	00008	000025E0	07238	02901 05162 05217	
=C 'TRIM\$	'				
	00008	000025E8	07239	02903	
=C 'INDEX	'				
	00008	000025F0	07240	02905 03693	
=C 'INDEXB	'				
	00008	000025F8	07241	02907	
=C 'ISNUM	'				
	00008	00002600	07242	02909	
=C 'ISALPHA	'				
	00008	00002608	07243	02911	
=C 'ISLOWER	'				
	00008	00002610	07244	02913 03195	
=C 'ISUPPER	'				
	00008	00002618	07245	02915 03197	
=C 'ISBLANK	'				
	00008	00002620	07246	02917	
=C 'ISDSN	'				
	00008	00002628	07247	02919	
=C 'ECHO	'				
	00008	00002630	07248	02921	
=C 'ECHOQ	'				
	00008	00002638	07249	02923	
=C 'REVRS	'				
	00008	00002640	07250	02925	
=C 'UPPERC	'				
	00008	00002648	07251	02927 03536	
=C 'LOWERC	'				
	00008	00002650	07252	02929	
=C 'COUNT	'				
	00008	00002658	07253	02931 03705	
=C 'FINDALL	'				
	00008	00002660	07254	02933 03707	
=C 'FINDL	'				
	00008	00002668	07255	02935	
=C 'FIND	'				
	00008	00002670	07256	02937 03691	
=C 'CENTER	'				
	00008	00002678	07257	02940	
=C 'LJUST	'				
	00008	00002680	07258	02942	
=C 'RJUST	'				
	00008	00002688	07259	02944	
=C 'ZFILL	'				
	00008	00002690	07260	02946 03900	
=C 'WORDS	'				

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
	00008	00002698	07261	02948	
=C'GEN#					
	00008	000026A0	07262	02950	
=C'DD2DSN					
	00008	000026A8	07263	02952	
=C'JOBINFO					
	00008	000026B0	07264	02954	
=C'DAYSMM					
	00008	000026B8	07265	02958	
=C'DAYSYY					
	00008	000026C0	07266	02960	
=C'ISLEAP					
	00008	000026C8	07267	02962	04429
=C'CYJ-D8					
	00008	000026D0	07268	02965	
=C'CYJ-DAY					
	00008	000026D8	07269	02967	
=C'CYJ-DOW					
	00008	000026E0	07270	02969	
=C'JCY-D8					
	00008	000026E8	07271	02973	
=C'JCY-DAY					
	00008	000026F0	07272	02975	
=C'JCY-DOW					
	00008	000026F8	07273	02977	
=C'MDCY-D8					
	00008	00002700	07274	02981	
=C'MDCY-DAY'					
	00008	00002708	07275	02983	
=C'MDCY-DOW'					
	00008	00002710	07276	02985	
=C'DMCY-D8					
	00008	00002718	07277	02989	
=C'DMCY-DAY'					
	00008	00002720	07278	02991	
=C'DMCY-DOW'					
	00008	00002728	07279	02993	
=C'CYMD-D8					
	00008	00002730	07280	02997	
=C'CYMD-DAY'					
	00008	00002738	07281	02999	
=C'CYMD-DOW'					
	00008	00002740	07282	03001	
=C'CYDM-D8					
	00008	00002748	07283	03005	
=C'CYDM-DAY'					
	00008	00002750	07284	03007	
=C'CYDM-DOW'					
	00008	00002758	07285	03009	
=C'FILL					
	00008	00002760	07286	03014	04968
=C'LSTRIP					
	00008	00002768	07287	03016	
=C'RSTRIP					
	00008	00002770	07288	03018	

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
=C'STRIP	'				
	00008	00002778	07289	03020	
=C'STRIP\$	'				
	00008	00002780	07290	03022	
=C'CONCAT	'				
	00008	00002788	07291	03024	
=C'UNSTR	'				
	00008	00002790	07292	03026	
=C'REPLACE	'				
	00008	00002798	07293	03028 07010	
=C'VEXIST	'				
	00008	000027A0	07294	03030	
=C'TDSN	'				
	00008	000027A8	07295	03032 04005	
=C'NOW	'				
	00008	000027B0	07296	03034 04007	
=C'PAD	'				
	00008	000027B8	07297	03036	
=C'\$NOVAR2	'				
	00008	000027C0	07298	03155 03189 03300 03337 03449 03613 03934 03988 04222 04291 04383 05650 06021 06166	
=C'QUOTE	'				
	00008	000027C8	07299	05979	
=C'ERRMSG	'				
	00008	000027D0	07300	06186	
=F'0'		000027D8	07301	02653 03711 04958 05537 05965 06096 07006 07068 07166 07200	
=F'3'		000027DC	07302	02816 03621 04011 04935 07119 07154	
=C'-- SETVAR word 3	'				
	00020	000027E0	07303	02823	
=F'1'		000027F4	07304	02849 03674 03715 03750 03757 03773 03834 04892 04905 05554 06933 07168 07202	
=C'CYJ-	'	000027F8	07305	02971 04590 06567	
=C'JCY-	'	000027FC	07306	02979 04592 06569	
=F'4000'		00002800	07307	03162 04320 04412 04526 04770 04918	
=F'44'		00002804	07308	03351	
=F'8'		00002808	07309	04134 04138 04763	
=F'16'		0000280C	07310	04175	
=C'DD.NOT.FOUND'					
	00012	00002810	07311	04196	
=F'12'		0000281C	07312	04197	
=F'24'		00002820	07313	04242 04243	
=F'6'		00002824	07314	04311 06187	
=F'4'		00002828	07315	04403 04890 04937	
=F'7'		0000282C	07316	04519	
=C'D8	'	00002830	07317	04570	
=C'DAY	'	00002834	07318	04572	
=C'DOW	'	00002838	07319	04574	
=F'2'		0000283C	07320	04933 07170 07204	
=F'256'		00002840	07321	04956 05358 05360 05594 05609 05960 06091	
=F'4098'		00002844	07322	05833	
=F'4099'		00002848	07323	05837	
=C'MDCY'		0000284C	07324	06601	
=C'DMCY'		00002850	07325	06603	
=C'CYMD'		00002854	07326	06605	
=C'CYDM'		00002858	07327	06607	
=H'4'		0000285C	07328	02761	
=X'0000'		0000285E	07329	03057 03480	

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 16.49 04/12/20
=X'00FF'	00002	00002860	07330	03475 03477	
=P'19'	00002	00002862	07331	04067	
=C'02'	00002	00002864	07332	04346 06517 06711 06789 06807	
=P'365'	00002	00002866	07333	06411	
=P'100'	00002	00002868	07334	06417 06525	
=P'400'	00002	0000286A	07335	06418 06526	
=C', '	00002	0000286C	07336	06466	
=C'01'	00002	0000286E	07337	06787	
=C'29'	00002	00002870	07338	06793	
=P'99'	00002	00002872	07339	06856	
=C'MDCY-'					
	00005	00002874	07340	02987 04582	
=C'DMCY-'					
	00005	00002879	07341	02995 04584	
=C'CYMD-'					
	00005	0000287E	07342	03003 04586	
=C'CYDM-'					
	00005	00002883	07343	03011 04588	
=P'+0'	00001	00002888	07344	03356 03357 03372 03409 03427	
=P'+1'	00001	00002889	07345	03362 03368 03421	
=P'+8'	00001	0000288A	07346	03370	
=C'" "'	00003	0000288B	07347	03620	
=P'1'	00001	0000288E	07348	04435 04552 06428 06429 06519 06791 06847	
=C'D8 '	00003	0000288F	07349	04811	
=C'DAY'	00003	00002892	07350	04813	
=C'DOW'	00003	00002895	07351	04815	
=C'1'	00001	00002898	07352	04950 07182	
=P'0'	00001	00002899	07353	05430 06410 06419 06422 06425	
=P'4'	00001	0000289A	07354	06416 06524	
=P'7'	00001	0000289B	07355	06533	
=P'9'	00001	0000289C	07356	06854	
=C'0'	00001	0000289D	07357	06928	
=C'999'	00003	0000289E	07358	07216	

ASM 0201 16.49 04/12/20

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, NUMBER, OBJECT, RENT, RLD, STMT, NOLIBMAC, TERMINAL, NOTEST, XREF(SHORT)

SYSPARM()

WORK FILE BUFFER SIZE/NUMBER =19066/ 1

TOTAL RECORDS READ FROM SYSTEM INPUT	7643
--------------------------------------	------

TOTAL RECORDS READ FROM SYSTEM LIBRARY	7005
--	------

TOTAL RECORDS PUNCHED	229
-----------------------	-----

TOTAL RECORDS PRINTED	9061
-----------------------	------

F64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP,LIST,LET,RENT,XREF
DEFAULT OPTION(S) USED - SIZE=(231424,55296)
IEW0000 NAME CUTIL00(R)

CROSS REFERENCE TABLE

CONTROL SECTION			ENTRY							
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
CUTIL00	00	2F52								

LOCATION REFERS TO SYMBOL IN CONTROL SECTION LOCATION REFERS TO SYMBOL IN CONTROL SECTION
ENTRY ADDRESS 00

TOTAL LENGTH 2F58
***CUTIL00 NOW REPLACED IN DATA SET
AUTHORIZATION CODE IS 0.
**MODULE HAS BEEN MARKED REENTERABLE, AND REUSABLE.

./ ADD NAME=CUTIL00

)F FUNCTION -

The CUTIL00 command is a TPM that performs function on CLIST variables such as CENTER and TRIM string functions. All processing occurs internally on CLIST variables. Return code is returned via &LASTCC. See program documentation for more functional details.

)X Statement SYNTAX -

CUTIL00 FUNCT VAR1 VAR2 QUOTE

REQUIRED - FUNCT, VAR1

DEFAULTS - NONE

ALIAS - NONE

)O OPERANDS -

Parameters must appear in order as listed

))FUNCT - required, function to be performed

))VAR1 - required, name of CLIST variable

- o is not prefixed with an '&'
- o variable must exist before invoking CUTIL00
- o 1-8 bytes in length, must follow CLIST variable naming conventions

))VAR2 - optional, name of CLIST variable to store results or contain function parameters

- o is not prefixed with an '&'
- o 1-8 bytes in length, must follow CLIST variable naming conventions
- o Storing results
 - If variable does not exist, it will be created
 - If variable does exist, it will be updated
 - If no variable is specified, use VAR1 variable
 - If '\$NOVAR2' is specified, use VAR1 variable
 - '\$NOVAR2' used as a placeholder if VAR2 is not specified with QUOTE option
- o Function Parameters
 - variable must exist before invoking CUTIL00

))QUOTE - optional, quote delimited string values

- o Used by CUTIL00 3270 ISPF test tool
- o If 'QUOTE' is specified, string content between first and last quotes comprise VAR1 value.
- o Use this option to include leading or trailing spaces in a string.

The following variable names provide additional information after successful execution of CUTIL00:

- o &LASTCC contains return code from CUTIL00 after execution
- o &ERRMSG contains error message text for function executed
- o &var1L contains length of var1 after execution

o &var2L contains length of var2 after execution

NOTE: The above variables may not be declared by CUTIL00 if CUTIL00 abends or does not execute. Variables include &ERRMSG, &var1L and &var2L. Suggested to initialize to NULLS before invoking CUTIL00. See LTRIM CLIST example below.

NOTE: - CLIST variable name length is 8 bytes max
- CLIST variable data length is 256 bytes max
- Result data length is 256 bytes max

Function Summary: (see program for more details on each function)

ABOUT - Display mySTAMP information

LTRIM - Remove leading spaces

RTRIM - Remove trailing spaces

TRIM - Remove leading and trailing spaces

TRIM\$ - Remove leading, trailing, and duplicate spaces

INDEX - Alias for FIND

INDEXB - Alias for FINDL

ISNUM - Test for all numeric (0-9)

ISALPHA - Test for all alphabetic (A-Z,a-z)

ISLOWER - Test for all alphabetic (a-z)

ISUPPER - Test for all alphabetic (A-Z)

ISBLANK - Test for all blank (whitespace)

ISDSN - Test for valid MVS Dataset Name

ECHO - Display string value

ECHOQ - Display string value surrounded by single quotes
on terminal screen

REVRS - Translate to reverse A-Z > Z-A, 0-9 > 9-0

UPPERC - Translate to uppercase

LOWERC - Translate to lowercase

COUNT - Count occurrences of a search string

FINDALL - Alias for COUNT

FIND - Find starting position of FIRST search string

FINDL - Find starting position of LAST search string

CENTER - Center content

LJUST - Left justify content

RJUST - Right justify content

ZFILL - Zero fill after content is right justified

WORDS - Number of words (contiguous non-blank groups)

GEN# - Generate a 3 digit number (000-999)

DD2DSN - Dataset name associated with DD name
JOBINFO - Job Information

DAYSMM - Number of days in MM
DAYSY - Number of days in CCYY (365 or 366)
ISLEAP - Test CCYY for leap year

CYJ-D8 - Date Conversion CCYYJJJ to monthname DD, CCYY
CYJ-DAY - Date Conversion CCYYJJJ to day name
CYJ-DOW - Date Conversion CCYYJJJ to day of week number
CYJ-mdcy - Date Conversion CCYYJJJ to MMDDCCYY
-mdcy can be in any order

JCY-D8 - Date Conversion JJJCCYY to monthname DD, CCYY
JCY-DAY - Date Conversion JJJCCYY to day name
JCY-DOW - Date Conversion JJJCCYY to day of week number
JCY-mdcy - Date Conversion JJJCCYY to MMDDCCYY
-mdcy can be in any order

MDCY-D8 - Date Conversion MMDDCCYY to monthname DD, CCYY
MDCY-DAY - Date Conversion MMDDCCYY to day name
MDCY-DOW - Date Conversion MMDDCCYY to day of week number
MDCY-cyj - Date Conversion MMDDCCYY to CCYYJJJ
-cyj can be in any order

DMCY-D8 - Date Conversion DDMMCCYY to monthname DD, CCYY
DMCY-DAY - Date Conversion DDMMCCYY to day name
DMCY-DOW - Date Conversion DDMMCCYY to day of week number
DMCY-cyj - Date Conversion DDMMCCYY to CCYYJJJ
-cyj can be in any order

CYMD-D8 - Date Conversion CCYYMMDD to monthname DD, CCYY
CYMD-DAY - Date Conversion CCYYMMDD to day name
CYMD-DOW - Date Conversion CCYYMMDD to day of week number
CYMD-cyj - Date Conversion CCYYMMDD to CCYYJJJ
-cyj can be in any order

CYDM-D8 - Date Conversion CCYYDDMM to monthname DD, CCYY
CYDM-DAY - Date Conversion CCYYDDMM to day name
CYDM-DOW - Date Conversion CCYYDDMM to day of week number
CYDM-cyj - Date Conversion CCYYDDMM to CCYYJJJ
-cyj can be in any order

FILL - Fill VAR1 nnn times with delimiter

LSTRIP - Remove leading delimiter
RSTRIP - Remove trailing delimiter
STRIP - Remove leading and trailing delimiter
STRIP\$ - Remove all delimiter

- CONCAT - Concatenate two strings
- UNSTR - Unstring into separate variables
- REPLACE - Replace substring in string
- VEXIST - Test if CLIST Variable exists
- TDSN - Generate temporary DSN nodes
- NOW - Current system date and time
- PAD - PAD string with delimiter to given length

./ ENDUP

IEB816I MEMBER NAME (CUTIL00) FOUND IN NM DIRECTORY. TTR IS NOW ALTERED.
IEB818I HIGHEST CONDITION CODE WAS 00000000
IEB819I END OF JOB IEBUPDTE.

```
./ ADD NAME=CCUTIL00
PROC 1 FUNCT VAR01() VAR02($N$) UQ1() CLS(Y) VERBOSE(Y)

/*****
/*
/*      CLIST: CCUTIL00
/*      AUTHOR: Larry Belmontes Jr.
/*              https://ShareABitOfIT.net/CUTIL00-for-mvs-3-8j/
/*              Copyright (C) 2020 Larry Belmontes, Jr.
/*
/*
/*      Description:
/*      =====
/*      Test harness to invoke CUTIL00 from TSO.
/*
/*
/*      Command Syntax:
/*      =====
/*      CCUTIL00 FUNCT VAR01('xx.') VAR02('yyy yy')
/*
/*      FUNCT (Required) CUTIL00 Function
/*      VAR01 (Required) Content for VAR01 in quotes
/*      VAR02 (Optional) Content for VAR02 in quotes
/*                        dependent on function
/*      CLS   (Y/N)      Clear screen displaying results
/*                        default=Y
/*      VERBOSE (Y/N)    Display header and command line
/*                        default=Y
/*
/*      Example:
/*      CCUTIL00 TRIM VAR01(' TRIM  MY SPACE  ') VAR02(X)
/*
/*
/*      Sample Output:
/*      --- CCUTIL00: CUTIL00 TEST TOOL
/*
/*      CUTIL00 TRIM  VAR01
/*
/*      --- CUTIL00 RESPONSE .....
/*      RC      =0 SUCCESSFUL REQUEST
/*      VAR01 =' TRIM  MY SPACE  '
/*      VAR01L=00018
/*      VAR02 ='TRIM  MY SPACE'
/*      VAR02L=00014
/*
/*      --- CCUTIL00: CUTIL00 TEST TOOL  ...DONE!  TRIM
/*
/*
/*      Disclaimer: <DSCLAIMR>
/*      =====
/*
```

```
/*
/*      No guarantee; No warranty; Install / Use at your own risk.
/*
/*      This software is provided "AS IS" and without any expressed
/* or implied warranties, including, without limitation, the
/* implied warranties of merchantability and fitness for a
/* particular purpose.
/*
/*      The author requests keeping authors name intact in any
/* modified versions.
/*
/*      In addition, the author requests submissions regarding any
/* code modifications / enhancements and associated comments
/* for consideration into a subsequent release (giving credit
/* to contributor(s)) thus, improving overall functionality
/* and further benefiting the MVS 3.8J hobbyist public domain
/* community.
/*
/*
/* Change History:
/* =====
/* MM/DD/CCYY Version  Name / Description
/* -----
/* 03/20/2020 1.0.00    Larry Belmontes Jr.
/*                  - Initial version released to MVS 3.8J
/*                  hobbyist public domain
/*
/*
/* *****
/*
/* *****
/* Clear Screen
/* *****
IF &CLS = Y THEN -
  CLS

SET TPM = &STR(CUTIL00)
SET MYPROC = &STR(CCUTIL00)
IF &VERBOSE = Y THEN DO
  WRITE --- &MYPROC: &TPM Test Tool
  WRITE
END

/* *****
/* Default CUTIL00 return values
/* *****
SET ERRMSG =
SET VAR01L =
SET VAR02L =

/* *****
/* Default CUTIL00 command string
/* -- ABOUT does not need var parms
```

```
/* -- VEXIST uses VAR01 content for VAR1      */
/*****/
IF &STR(&FUNCT) = VEXIST THEN -
  SET CMDLINE= &STR(&TPM &FUNCT &VAR01)
ELSE -
  IF &STR(&FUNCT) = ABOUT THEN -
    SET CMDLINE= &STR(&TPM &FUNCT)
  ELSE -
    SET CMDLINE= &STR(&TPM &FUNCT VAR01)

/*****/
/* Build CUTIL00 command string                */
/* based on VAR02 options                      */
/*****/
IF &STR(&VAR02) = $N$ THEN DO
  SET VAR02 =
END
ELSE -
  IF &STR(&VAR02) = $NOVAR2 THEN DO
    SET VAR02 =
    SET CMDLINE= &STR(&CMDLINE $NOVAR2)
  END
  ELSE DO
    SET CMDLINE= &STR(&CMDLINE VAR02)
  END

/*****/
/* Execute CUTIL00 and display results          */
/*****/
&CMDLINE
SET RC = &LASTCC
IF &VERBOSE = Y THEN DO
  WRITE &CMDLINE
  WRITE
END
WRITE --- &TPM Response .....
WRITE RC      =&RC &ERRMSG
WRITE VAR01   ='&VAR01'
WRITE VAR01L  =&VAR01L
WRITE VAR02   ='&VAR02'
WRITE VAR02L  =&VAR02L
IF &STR(&FUNCT) = UNSTR THEN DO
  WRITE - &FUNCT VARIABLE(S) CREATED: &VAR010
  SET CTR = 0
  DO WHILE &CTR <= &VAR010
    SET EVALLN = &STR(&&VAR01&CTR)
    WRITE VAR01&CTR = '&EVALLN'
    SET CTR = &CTR + 1
  END
END

WRITE --- &MYPROC: &TPM Test Tool ... Done! &FUNCT
```

WRITE

```
/* *****  
/* Exit point  
/* *****
```

ENDIT: -
END

EXIT CODE(0)
./ ENDUP

IEB816I MEMBER NAME (CCUTIL00) FOUND IN NM DIRECTORY. TTR IS NOW ALTERED.
IEB818I HIGHEST CONDITION CODE WAS 00000000
IEB819I END OF JOB IEBUPDTE.

```
./ ADD NAME=CCUTIL0V  
PROC 0
```

CONTROL MAIN

```
/* **** */  
/* */  
/* CLIST: CCUTIL0V */  
/* AUTHOR: Larry Belmontes Jr. */  
/* https://ShareABitOfIT.net/cutil00-for-mvs-3-8j/ */  
/* Copyright (C) 2020 Larry Belmontes, Jr. */  
/* */  
/* Description: */  
/* ===== */  
/* Validation CLIST to execute each CUTIL00 function and print */  
/* results. This CLIST should be submitted under TSO Batch. */  
/* */  
/* Contains samples from program documentation. */  
/* */  
/* Command Syntax: */  
/* ===== */  
/* */  
/* CCUTIL00 */  
/* */  
/* This is a simple CLIST that invokes CUTIL00 multiple times */  
/* using the following approach for each test: */  
/* 1 - SET variables to be used by CUTIL00 function */  
/* 2 - Invoke CUTIL00 */  
/* 3 - Print results from CUTIL00 */  
/* */  
/* Example: */  
/* CCUTIL00 */  
/* */  
/* Disclaimer: <DSCLAIMR> */  
/* ===== */  
/* No guarantee; No warranty; Install / Use at your own risk. */  
/* */  
/* This software is provided "AS IS" and without any expressed */  
/* or implied warranties, including, without limitation, the */  
/* implied warranties of merchantability and fitness for a */  
/* particular purpose. */  
/* */  
/* The author requests keeping authors name intact in any */  
/* modified versions. */  
/* */  
/* In addition, the author requests submissions regarding any */  
/* code modifications / enhancements and associated comments */  
/* for consideration into a subsequent release (giving credit */
```

```
/* to contributor(s)) thus, improving overall functionality */
/* and further benefiting the MVS 3.8J hobbyist public domain */
/* community. */
/*
/*
/* Change History:
/* =====
/* MM/DD/CCYY Version Name / Description
/* -----
/* 03/20/2020 1.0.00 Larry Belmontes Jr.
/* - Initial version released to MVS 3.8J
/* hobbyist public domain
/*
/*
/* *****/
```

```
SET TSTHDR = &STR(====> TEST )
SET I = 0
```

```
WRITE
WRITE ***** Testing CUTIL00 after installation *** &SYSDATE &SYSTIME
WRITE ***** program documention samples ***
WRITE
```

```
WRITE /\ /\ /\ *00 A B O U T /\ /\ /\ -----
/* *****
/* *00 ABOUT from program documentation
/* *****
```

```
SET MYCMD = &STR(CUTIL00 ABOUT MYDATA MYRESULT)
```

```
WRITE &STR(&TSTHDR)&I: &MYCMD
```

```
SET MYDATA =
```

```
SET MYDATAL =
```

```
SET MYRESULT =
```

```
SET MYRESULTL =
```

```
SET ERRMSG =
```

```
&MYCMD
```

```
SET RC = &LASTCC
```

```
WRITE RC = &RC
```

```
WRITE ERRMSG = &ERRMSG
```

```
WRITE MYDATA = '&MYDATA'
```

```
WRITE MYDATAL = &MYDATAL
```

```
WRITE MYRESULT = '&MYRESULT'
```

```
WRITE MYRESULTL = &MYRESULTL
```

```
WRITE
```

```
WRITE /\ /\ /\ *01 L T R I M /\ /\ /\ -----
/* *****
```

```
/* *01 LTRIM from program documentation
/* *****
```

```
SET I = &I + 1
```

```
SET MYCMD = &STR(CUTIL00 LTRIM MYDATA MYRESULT)
```

```
WRITE &STR(&TSTHDR)&I: &MYCMD
```

```
SET MYDATA = &STR( leading spaces)
```

```
SET MYDATAL =
```

```
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *02 R T R I M /\ /\ /\ -----
/*****/
/* RTRIM from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 RTRIM MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(trailing spaces )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *03 T R I M /\ /\ /\ -----
/*****/
/* *03 TRIM from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 TRIM MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(trim my data )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
```

```
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *04  T R I M #                /\ /\ /\ -----
/*****
/* *04 TRIM$ from program documentation      */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 TRIM$ MYDATA  MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR( trim  my  data )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *05  I N D E X                /\ /\ /\ -----
/*****
/* *05 INDEX from program documentation      */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 INDEX MYDATA  DLMTR)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(my, string, ok?  )
SET MYDATAL =
SET DLMTR = &STR(",")
SET DLMTRL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE DLMTR   ='&DLMTR'
WRITE DLMTRL  =&DLMTRL
WRITE
WRITE /\ /\ /\ *06  I N D E X B                /\ /\ /\ -----
/*****
/* *06 INDEXB from program documentation    */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 INDEXB MYDATA  DLMTR)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(my, string, ok?  )
```

```
SET MYDATAL =
SET DLMTR = &STR(",")
SET DLMTRL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE DLMTR = '&DLMTR'
WRITE DLMTRL =&DLMTRL
WRITE
WRITE /\ /\ /\*07 I S N U M /\ /\ /\-----
/*****
/* *07 ISNUM from program documentation */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISNUM MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(0123456789)
SET MYDATAL =
SET MYRESULT = /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*08 I S A L P H A /\ /\ /\-----
/*****
/* *08 ISALPHA from program documentation */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISALPHA MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(abCDEfghiJ)
SET MYDATAL =
SET MYRESULT = /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
```

```
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*09  I S L O W E R                /\ /\ /\-----
/*****
/* *09 ISLOWER from program documentation */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISLOWER MYDATA          )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(abcdefghij)
SET MYDATAL =
SET MYRESULT =                /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*10  I S U P P E R                /\ /\ /\-----
/*****
/* *10 ISUPPER from program documentation */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISUPPER MYDATA          )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(ABCDEFGHIIJ)
SET MYDATAL =
SET MYRESULT =                /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*11  I S B L A N K                /\ /\ /\-----
/*****
/* *11 ISBLANK from program documentation */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISBLANK MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
```

```
SET MYDATA = &STR(      )
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*12 I S D S N          /\ /\ /\-----
/*****
/* *12 ISDSN from program documentation */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISDSN MYDATA )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(A.A1234567)
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*13 E C H O          /\ /\ /\-----
/*****
/* *13 ECHO from program documentation */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ECHO MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(my data to display.)
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
WRITE CUTIL00 ECHO of MYDATA follows...
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
```

```
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*14  E C H O Q                      /\ /\ /\-----
/*****/
/* *14 ECHOQ from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ECHOQ  MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(my data to display.)
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
WRITE CUTIL00 ECHOQ of MYDATA follows...
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*15  R E V R S                      /\ /\ /\-----
/*****/
/* *15 REVRS from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 REVRS  MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(ABCDEFGHIJKLM96%QRS)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*16  U P P E R C                      /\ /\ /\-----
/*****/
/* *16 UPPERC from program documentation */
/*****/
```

```
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 UPPERC MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(abCdeFgHIJklmnopqrS)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE CUTIL00 ECHOQ of MYDATA follows...
CUTIL00 ECHOQ MYDATA
WRITE
WRITE /\ /\ /\*17 L O W E R C /\ /\ /\-----
/*****/
/* *17 LOWERC from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 LOWERC MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(abCdeFgHIJklmnopqrS)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE CUTIL00 ECHOQ of MYDATA follows...
CUTIL00 ECHOQ MYDATA
WRITE
WRITE /\ /\ /\*18 C O U N T /\ /\ /\-----
/*****/
/* *18 COUNT from program documentati */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 COUNT MYDATA SRCHPRMS)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(My name is Dino, big Dino.)
SET MYDATAL =
SET SRCHPRMS = &STR("Dino") /* Search Parm
```

```
SET SRCHPRMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE SRCHPRMS = '&SRCHPRMS'
WRITE SRCHPRMSL=&SRCHPRMSL
WRITE
WRITE /\ /\ /\ *19 F I N D /\ /\ /\ -----
/*****/
/* *19 FIND from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 FIND MYDATA SRCHPRMS)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(My name is Dino, big Dino.)
SET MYDATAL =
SET SRCHPRMS = &STR("Dino") /* Search Params */
SET SRCHPRMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE SRCHPRMS = '&SRCHPRMS'
WRITE SRCHPRMSL=&SRCHPRMSL
WRITE
WRITE /\ /\ /\ *20 F I N D L /\ /\ /\ -----
/*****/
/* *20 FINDL from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 FINDL MYDATA SRCHPRMS)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(My name is Dino, big Dino.)
SET MYDATAL =
SET SRCHPRMS = &STR("Dino") /* Search Params */
SET SRCHPRMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE SRCHPRMS = '&SRCHPRMS'
WRITE SRCHPRMSL=&SRCHPRMSL
```

```
WRITE
WRITE /\ /\ /\*21  C E N T E R                /\ /\ /\-----
/*****/
/* *21 CENTER from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CENTER  MYDATA  MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(Center this T E X T ... )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*22  L J U S T                /\ /\ /\-----
/*****/
/* *22 LJUST from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 LJUST  MYDATA  MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR( JUSTIFY this text )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*23  R J U S T                /\ /\ /\-----
/*****/
/* *23 RJUST from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 RJUST  MYDATA  MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR( JUSTIFY this text )
SET MYDATAL =
```

```
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *24 Z F I L L /\ /\ /\ -----
/* ***** */
/* *24 ZFILL from program documentation */
/* ***** */
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ZFILL MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(12 )
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *25 W O R D S /\ /\ /\ -----
/* ***** */
/* *25 WORDS from program documentation */
/* ***** */
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 WORDS MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR( one two three33 four., five )
SET MYDATAL =
SET MYRESULT = /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
```

```
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*26  G E N #                /\ /\ /\-----
/*****
/* *26 GEN# from program documentation      */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 GEN#      MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = 0
SET MYDATAL =
SET MYRESULT =                /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*27  D D 2 D S N            /\ /\ /\-----
/*****
/* *27 DD2DSN from program documentation      */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DD2DSN  MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(SYSPROC)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*28  J O B I N F O          /\ /\ /\-----
/*****
/* *28 JOBINFO from program documentation      */
/*****

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 JOBINFO MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(123456789012345678901234)
```

```
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE  RC          =&RC
WRITE  ERRMSG      =&ERRMSG
WRITE  MYDATA      ='&MYDATA'
WRITE  MYDATAL     =&MYDATAL
WRITE  MYRESULT    ='&MYRESULT'
WRITE  MYRESULTL   =&MYRESULTL
WRITE
WRITE /\ /\ /\ *29  D A Y S M M          /\ /\ /\ -----
/*****
/* *29 DAYSMM from program documentation          */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DAYSMM  MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(022016)
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE  RC          =&RC
WRITE  ERRMSG      =&ERRMSG
WRITE  MYDATA      ='&MYDATA'
WRITE  MYDATAL     =&MYDATAL
WRITE  MYRESULT    ='&MYRESULT'
WRITE  MYRESULTL   =&MYRESULTL
WRITE
WRITE /\ /\ /\ *30  D A Y S Y Y          /\ /\ /\ -----
/*****
/* *30 DAYSYY from program documentation          */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DAYSYY  MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(2020)
SET MYDATAL =
SET MYRESULT =          /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE  RC          =&RC
WRITE  ERRMSG      =&ERRMSG
WRITE  MYDATA      ='&MYDATA'
WRITE  MYDATAL     =&MYDATAL
```

```
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*31 I S L E A P /\ /\ /\-----
/*****/
/* *31 ISLEAP from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 ISLEAP MYDATA)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(2020)
SET MYDATAL =
SET MYRESULT = /* not used for this function */
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*32 C Y J - D 8 /\ /\ /\-----
/*****/
/* *32 CYJ-D8 from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYJ-D8 MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(2020061)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*33 C Y J - D A Y /\ /\ /\-----
/*****/
/* *33 CYJ-DAY from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYJ-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
```

```
SET MYDATA  = &STR(1981061)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\*34  C Y J - D O W                /\ /\ /\-----
/*****/
/* *34 CYJ-DOW from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYJ-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(1980061)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\*35  C Y J - M D C Y                /\ /\ /\-----
/*****/
/* *35 CYJ-MDCY from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYJ-MDCY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(2020061)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
```

```
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*36 J C Y - D 8                /\ /\ /\-----
/*****/
/* *36 JCY-D8 from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 JCY-D8 MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(1612017)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*37 J C Y - D A Y                /\ /\ /\-----
/*****/
/* *37 JCY-DAY from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 JCY-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(0611981)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
```

```
WRITE /\ /\ /\ *38 J C Y - D O W                /\ /\ /\ -----
/*****/
/* *38 JCY-DOW from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 JCY-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(0611981)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC = &RC
WRITE ERRMSG = &ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL = &MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\ *39 J C Y - M D C Y                /\ /\ /\ -----
/*****/
/* *39 JCY-MDCY from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 JCY-MDCY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(1612017)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC = &RC
WRITE ERRMSG = &ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL = &MYRESULTL
WRITE
WRITE /\ /\ /\ *40 M D C Y - D 8                /\ /\ /\ -----
/*****/
/* *40 MDCY-D8 from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 MDCY-D8 MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(12101990)
```

```
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *41  M D C Y - D A Y          /\ /\ /\-----
/*****/
/* *41 MDCY-DAY from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 MDCY-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(12101990)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ  MYDATA
CUTIL00 ECHOQ  MYRESULT
WRITE
WRITE /\ /\ /\ *42      M D C Y - D O W          /\ /\ /\-----
/*****/
/* *42 MDCY-DOW from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 MDCY-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(12101990)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
```

```
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ  MYDATA
CUTIL00 ECHOQ  MYRESULT
WRITE
WRITE /\ /\ /\*43  M D C Y - C Y J                /\ /\ /\-----
/*****/
/* *43 MDCY-CYJ from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 MDCY-CYJ  MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(12101990)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*44  D M C Y - D 8                /\ /\ /\-----
/*****/
/* *44 DMCY-D8 from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DMCY-D8  MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(15071987)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL  =&MYDATAL
WRITE MYRESULT ='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*45  D M C Y - D A Y                /\ /\ /\-----
/*****/
/* *45 DMCY-DAY from program documentation */
```

```

/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DMCY-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(15071987)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ \*46 D M C Y - D O W /\ /\ \-----
/*****/
/* *46 DMCY-DOW from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DMCY-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(15071987)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ \*47 D M C Y - C Y J /\ /\ \-----
/*****/
/* *47 DMCY-CYJ from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 DMCY-CYJ MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(15071987)
SET MYDATAL =
```

```
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *48 C Y M D - D 8 /\ /\ /\ -----
/*****/
/* *48 CYMD-D8 from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYMD-D8 MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951231)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *49 C Y M D - D A Y /\ /\ /\ -----
/*****/
/* *49 CYMD-DAY from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYMD-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951231)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
```

```
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\*50 C Y M D - D O W /\ /\ /\-----
/*****/
/* *50 CYMD-DOW from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYMD-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951231)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\*51 C Y M D - C Y J /\ /\ /\-----
/*****/
/* *51 CYMD-CYJ from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYMD-CYJ MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951231)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*52 C Y D M - D 8 /\ /\ /\-----
/*****/
/* *52 CYDM-D8 from program documentation */
/*****/
```

```
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYDM-D8 MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951605)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *53 C Y D M - D A Y /\ /\ /\-----
/*****/
/* *53 CYDM-DAY from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYDM-DAY MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951605)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYRESULT = '&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\ *54 C Y D M - D O W /\ /\ /\-----
/*****/
/* *54 CYDM-DOW from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYDM-DOW MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(19951605)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
```

```
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
CUTIL00 ECHOQ MYDATA
CUTIL00 ECHOQ MYRESULT
WRITE
WRITE /\ /\ /\ *55  C Y D M - C Y J          /\ /\ /\ -----
/*****/
/* *55 CYDM-CYJ from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CYDM-CYJ  MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(19951605)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT='&MYRESULT'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *56  F I L L          /\ /\ /\ -----
/*****/
/* *56 FILL from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 FILL      MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(1)
SET MYDATAL =
SET MYPARMS = &STR(*20)  /* 20 *'s in VAR1 */
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYPARMS ='&MYPARMS'
WRITE MYPARMSL=&MYPARMSL
```

```
WRITE
WRITE /\ /\ /\ *57  L S T R I P                /\ /\ /\ -----
/*****
/* *57 LSTRIP from program documentation      */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 LSTRIP  MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(, , 28, BC, )
SET MYDATAL =
SET MYPARMS = &STR(, )
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      = &RC
WRITE ERRMSG  = &ERRMSG
WRITE MYDATA  = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYPARMS = '&MYPARMS'
WRITE MYPARMSL = &MYPARMSL
WRITE
WRITE /\ /\ /\ *58  R S T R I P                /\ /\ /\ -----
/*****
/* *58 RSTRIP from program documentation      */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 RSTRIP  MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(, , 28, BC, )
SET MYDATAL =
SET MYPARMS = &STR(, )
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      = &RC
WRITE ERRMSG  = &ERRMSG
WRITE MYDATA  = '&MYDATA'
WRITE MYDATAL = &MYDATAL
WRITE MYPARMS = '&MYPARMS'
WRITE MYPARMSL = &MYPARMSL
WRITE
WRITE /\ /\ /\ *59  S T R I P                /\ /\ /\ -----
/*****
/* *59 STRIP from program documentation      */
/*****
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 STRIP  MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA  = &STR(, , 28, BC, )
SET MYDATAL =
```

```
SET MYPARMS = &STR(,)
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYPARMS = '&MYPARMS'
WRITE MYPARMSL =&MYPARMSL
WRITE
WRITE /\ /\ /\*60 S T R I P $ /\ /\ /\-----
/*****/
/* *60 STRIP$ from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 STRIP$ MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(, , 28, BC, )
SET MYDATAL =
SET MYPARMS = &STR(,)
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYPARMS = '&MYPARMS'
WRITE MYPARMSL =&MYPARMSL
WRITE
WRITE /\ /\ /\*61 C O N C A T /\ /\ /\-----
/*****/
/* *61 CONCAT from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 CONCAT MYDATA MYRESULT)
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(String data 1.)
SET MYDATAL =
SET MYRESULT = &STR(A result string 2.)
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULT'
```

```
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\*62  U N S T R                /\ /\ /\-----
/*****/
/* *62 UNSTR from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 UNSTR      MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(1234567 and so on . I am 78 long. )
SET MYDATAL =
SET MYPARMS = &STR(7)
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYPARMS ='&MYPARMS'
WRITE MYPARMSL=&MYPARMSL
SET CTR = 0
DO WHILE &CTR <= &MYDATA0
    SET EVALLN = &STR(&&MYDATA&CTR)
    WRITE MYDATA&CTR = '&EVALLN'
    SET CTR = &CTR + 1
END
WRITE
WRITE /\ /\ /\*63  R E P L A C E                /\ /\ /\-----
/*****/
/* *63 REPLACE from program documentation */
/*****/

SET I = &I + 1
SET MYCMD = &STR(CUTIL00 REPLACE  MYDATA MYPARMS )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(1234567 AND SO ON . I AM 7879.)
SET MYDATAL =
SET MYPARMS = &STR("SO ON ", "890123456--")
SET MYPARMSL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC      =&RC
WRITE ERRMSG  =&ERRMSG
WRITE MYDATA  ='&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYPARMS ='&MYPARMS'
WRITE MYPARMSL=&MYPARMSL
WRITE
WRITE /\ /\ /\*64  V E X I S T                /\ /\ /\-----
/*****/
```

```
/* *64 VEXIST from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 VEXIST MYDATA )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA = &STR(TEST)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULTL'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *65 T D S N /\ /\ /\-----
/*****/
/* *65 TDSN from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 TDSN MYDATA )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA =&STR(HERC01.CARDS)
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
&MYCMD
SET RC = &LASTCC
WRITE RC =&RC
WRITE ERRMSG =&ERRMSG
WRITE MYDATA = '&MYDATA'
WRITE MYDATAL =&MYDATAL
WRITE MYRESULT = '&MYRESULTL'
WRITE MYRESULTL=&MYRESULTL
WRITE
WRITE /\ /\ /\ *66 N O W /\ /\ /\-----
/*****/
/* *66 NOW from program documentation */
/*****/
SET I = &I + 1
SET MYCMD = &STR(CUTIL00 NOW MYDATA )
WRITE &STR(&TSTHDR)&I: &MYCMD
SET MYDATA =
SET MYDATAL =
SET MYRESULT =
SET MYRESULTL =
SET ERRMSG =
```

SYSIN

NEW MASTER

IEBUPDTE LOG PAGE 0030

&MYCMD

SET RC = &LASTCC

WRITE RC =&RC

WRITE ERRMSG =&ERRMSG

WRITE MYDATA = '&MYDATA'

WRITE MYDATAL =&MYDATAL

WRITE MYRESULT = '&MYRESULTL'

WRITE MYRESULTL=&MYRESULTL

WRITE

WRITE /\ /\ /\ *67 P A D /\ /\ /\ -----

/*****

/* *67 PAD from program documentation */

/*****

SET I = &I + 1

SET MYCMD = &STR(CUTIL00 PAD MYDATA MYRESULT)

WRITE &STR(&TSTHDR)&I: &MYCMD

SET MYDATA =&STR(PAD ME TO 30 LONG)

SET MYDATAL =

SET MYRESULT =&STR(30)

SET MYRESULTL =

SET ERRMSG =

&MYCMD

SET RC = &LASTCC

WRITE RC =&RC

WRITE ERRMSG =&ERRMSG

WRITE MYDATA = '&MYDATA'

WRITE MYDATAL =&MYDATAL

WRITE MYRESULT = '&MYRESULTL'

WRITE MYRESULTL=&MYRESULTL

WRITE

WRITE ***** Testing CUTIL00 concluded. *** &SYSDATE &SYSTIME

ENDIT: -

END

EXIT CODE(0)

./ ENDUP

IEB816I MEMBER NAME (CCUTIL0V) FOUND IN NM DIRECTORY. TTR IS NOW ALTERED.

IEB818I HIGHEST CONDITION CODE WAS 00000000

IEB819I END OF JOB IEBUPDTE.

CCUTIL0V

***** TESTING CUTIL00 AFTER INSTALLATION *** 04/12/20 16:49:04
***** PROGRAM DOCUMENTION SAMPLES ***

/\ /\ /*00 A B O U T /\ /\ /\-----/*****/
====> TEST 0: CUTIL00 ABOUT MYDATA MYRESULT
CUTIL00 MVS3.8J V1.0.00 03202020 04/12/20 @ 16.49
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='
MYDATAL =
MYRESULT ='
MYRESULTL=

/\ /\ /*01 L T R I M /\ /\ /\-----/*****/
====> TEST 1: CUTIL00 LTRIM MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA =' LEADING SPACES'
MYDATAL =00017
MYRESULT ='LEADING SPACES'
MYRESULTL=00014

/\ /\ /*02 R T R I M /\ /\ /\-----/*****/
====> TEST 2: CUTIL00 RTRIM MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='TRAILING SPACES '
MYDATAL =00017
MYRESULT ='TRAILING SPACES'
MYRESULTL=00015

/\ /\ /*03 T R I M /\ /\ /\-----/*****/
====> TEST 3: CUTIL00 TRIM MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA =' TRIM MY DATA '
MYDATAL =00017
MYRESULT ='TRIM MY DATA'
MYRESULTL=00015

/\ /\ /*04 T R I M # /\ /\ /\-----/*****/
====> TEST 4: CUTIL00 TRIM\$ MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA =' TRIM MY DATA '
MYDATAL =00017
MYRESULT ='TRIM MY DATA'
MYRESULTL=00012

/\ /\ /*05 I N D E X /\ /\ /\-----/*****/
====> TEST 5: CUTIL00 INDEX MYDATA DLMTR
RC =3
ERRMSG =SUCCESSFUL INDEX
MYDATA ='MY, STRING, OK? '

MYDATAL =00018
DLMTR =', ''
DLMTRL =00003

/\ /\ /*06 I N D E X B /\ /\ /\ -----/*****
====> TEST 6: CUTIL00 INDEXB MYDATA DLMTR

RC =11
ERRMSG =SUCCESSFUL INDEXB
MYDATA ='MY, STRING, OK? '
MYDATAL =00018
DLMTR =', ''
DLMTRL =00003

/\ /\ /*07 I S N U M /\ /\ /\ -----/*****
====> TEST 7: CUTIL00 ISNUM MYDATA

RC =4001
ERRMSG =TRUE
MYDATA ='0123456789'
MYDATAL =00010
MYRESULT =''
MYRESULTL=

/\ /\ /*08 I S A L P H A /\ /\ /\ -----/*****
====> TEST 8: CUTIL00 ISALPHA MYDATA

RC =4001
ERRMSG =TRUE
MYDATA ='ABCDEFGHIJ'
MYDATAL =00010
MYRESULT =''
MYRESULTL=

/\ /\ /*09 I S L O W E R /\ /\ /\ -----/*****
====> TEST 9: CUTIL00 ISLOWER MYDATA

RC =4000
ERRMSG =FALSE
MYDATA ='ABCDEFGHIJ'
MYDATAL =00010
MYRESULT =''
MYRESULTL=

/\ /\ /*10 I S U P P E R /\ /\ /\ -----/*****
====> TEST 10: CUTIL00 ISUPPER MYDATA

RC =4001
ERRMSG =TRUE
MYDATA ='ABCDEFGHIJ'
MYDATAL =00010
MYRESULT =''
MYRESULTL=

/\ /\ /*11 I S B L A N K /\ /\ /\ -----/*****
====> TEST 11: CUTIL00 ISBLANK MYDATA

RC =4001
ERRMSG =TRUE
MYDATA ='
MYDATAL =00009
MYRESULT =''

```
MYRESULTL=
/\ /\ /\*12  I S D S N                /\ /\ /\-----/*****/
====> TEST 12: CUTIL00 ISDSN  MYDATA
RC      =4001
ERRMSG   =TRUE
MYDATA   ='A.A1234567'
MYDATA1  =00010
MYRESULT =' '
MYRESULTL=

/\ /\ /\*13  E C H O                /\ /\ /\-----/*****/
====> TEST 13: CUTIL00 ECHO  MYDATA
CUTIL00 ECHO OF MYDATA FOLLOWS...
MY DATA TO DISPLAY.
RC      =0
ERRMSG   =SUCCESSFUL REQUEST
MYDATA   ='MY DATA TO DISPLAY.'
MYDATA1  =00019
MYRESULT =' '
MYRESULTL=

/\ /\ /\*14  E C H O Q                /\ /\ /\-----/*****/
====> TEST 14: CUTIL00 ECHOQ  MYDATA
CUTIL00 ECHOQ OF MYDATA FOLLOWS...
'MY DATA TO DISPLAY.'
RC      =0
ERRMSG   =SUCCESSFUL REQUEST
MYDATA   ='MY DATA TO DISPLAY.'
MYDATA1  =00019
MYRESULT =' '
MYRESULTL=

/\ /\ /\*15  R E V R S                /\ /\ /\-----/*****/
====> TEST 15: CUTIL00 REVRS  MYDATA MYRESULT
RC      =0
ERRMSG   =SUCCESSFUL REQUEST
MYDATA   ='ABCDEFGHJKLM96%QRS'
MYDATA1  =00019
MYRESULT ='ZYXWVUTSRQPON03 JIH'
MYRESULTL=00019

/\ /\ /\*16  U P P E R C                /\ /\ /\-----/*****/
====> TEST 16: CUTIL00 UPPERC  MYDATA
RC      =0
ERRMSG   =SUCCESSFUL REQUEST
MYDATA   ='ABCDEFGHJKLMNOPQRS'
MYDATA1  =00019
MYRESULT =' '
MYRESULTL=
CUTIL00 ECHOQ OF MYDATA FOLLOWS...
'ABCDEFGHJKLMNOPQRS'

/\ /\ /\*17  L O W E R C                /\ /\ /\-----/*****/
====> TEST 17: CUTIL00 LOWERC  MYDATA  MYRESULT
RC      =0
```

```
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='ABCDEFGHJKLMNOPQRS'
MYDATA  =00019
MYRESULT ='ABCDEFGHJKLMNOPQRS'
MYRESULTL=00019
CUTIL00 ECHOQ OF MYDATA FOLLOWS...
'ABCDEFGHJKLMNOPQRS'

/\ /\ /\*18  C O U N T                /\ /\ /\-----/*****/
====> TEST 18: CUTIL00 COUNT  MYDATA  SRCHPRMS
RC      =2
ERRMSG  =SUCCESSFUL COUNT
MYDATA  ='MY NAME IS DINO, BIG DINO.'
MYDATA  =00026
SRCHPRMS ='"DINO"'
SRCHPRMSL=00006

/\ /\ /\*19  F I N D                /\ /\ /\-----/*****/
====> TEST 19: CUTIL00 FIND    MYDATA  SRCHPRMS
RC      =12
ERRMSG  =SUCCESSFUL FIND
MYDATA  ='MY NAME IS DINO, BIG DINO.'
MYDATA  =00026
SRCHPRMS ='"DINO"'
SRCHPRMSL=00006

/\ /\ /\*20  F I N D L              /\ /\ /\-----/*****/
====> TEST 20: CUTIL00 FINDL  MYDATA  SRCHPRMS
RC      =22
ERRMSG  =SUCCESSFUL FINDL
MYDATA  ='MY NAME IS DINO, BIG DINO.'
MYDATA  =00026
SRCHPRMS ='"DINO"'
SRCHPRMSL=00006

/\ /\ /\*21  C E N T E R            /\ /\ /\-----/*****/
====> TEST 21: CUTIL00 CENTER MYDATA  MYRESULT
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='CENTER THIS T E X T ... '
MYDATA  =00030
MYRESULT ='  CENTER THIS T E X T ... '
MYRESULTL=00030

/\ /\ /\*22  L J U S T              /\ /\ /\-----/*****/
====> TEST 22: CUTIL00 LJUST  MYDATA  MYRESULT
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  =' JUSTIFY THIS TEXT '
MYDATA  =00022
MYRESULT ='JUSTIFY THIS TEXT '
MYRESULTL=00022

/\ /\ /\*23  R J U S T              /\ /\ /\-----/*****/
====> TEST 23: CUTIL00 RJUST  MYDATA  MYRESULT
RC      =0
```

```
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  =' JUSTIFY THIS TEXT  '
MYDATA1 =00022
MYRESULT =' JUSTIFY THIS TEXT'
MYRESULTL=00022
```

```
/\ /\ \*24  Z F I L L          /\ /\ \-----/*****/
====> TEST 24: CUTIL00 ZFILL  MYDATA  MYRESULT
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='12  '
MYDATA1 =00005
MYRESULT ='00012'
MYRESULTL=00005
```

```
/\ /\ \*25  W O R D S          /\ /\ \-----/*****/
====> TEST 25: CUTIL00 WORDS  MYDATA
RC      =5
ERRMSG  =SUCCESSFUL WORDS
MYDATA  =' ONE TWO THREE33 FOUR., FIVE '
MYDATA1 =00030
MYRESULT =' '
MYRESULTL=
```

```
/\ /\ \*26  G E N #           /\ /\ \-----/*****/
====> TEST 26: CUTIL00 GEN#   MYDATA
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='431'
MYDATA1 =00003
MYRESULT =' '
MYRESULTL=
```

```
/\ /\ \*27  D D 2 D S N       /\ /\ \-----/*****/
====> TEST 27: CUTIL00 DD2DSN MYDATA MYRESULT
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='SYSPROC'
MYDATA1 =00007
MYRESULT ='SYS2.CMDPROC'
MYRESULTL=00012
```

```
/\ /\ \*28  J O B I N F O     /\ /\ \-----/*****/
====> TEST 28: CUTIL00 JOBINFO MYDATA
RC      =0
ERRMSG  =SUCCESSFUL REQUEST
MYDATA  ='CUTIL001STEP01 TESTIT0 '
MYDATA1 =00024
MYRESULT =' '
MYRESULTL=
```

```
/\ /\ \*29  D A Y S M M       /\ /\ \-----/*****/
====> TEST 29: CUTIL00 DAYSMM  MYDATA
RC      =29
ERRMSG  =SUCCESSFUL DAYSMM
MYDATA  ='022016'
```

MYDATAL =00006
MYRESULT =''
MYRESULTL=

/\ /\ /*30 D A Y S Y Y /\ /\ /\-----/*****
====> TEST 30: CUTIL00 DAYSYY MYDATA

RC =366
ERRMSG =SUCCESSFUL DAYSYY
MYDATA ='2020'
MYDATAL =00004
MYRESULT =''
MYRESULTL=

/\ /\ /*31 I S L E A P /\ /\ /\-----/*****
====> TEST 31: CUTIL00 ISLEAP MYDATA

RC =4001
ERRMSG =TRUE
MYDATA ='2020'
MYDATAL =00004
MYRESULT =''
MYRESULTL=

/\ /\ /*32 C Y J - D 8 /\ /\ /\-----/*****
====> TEST 32: CUTIL00 CYJ-D8 MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='2020061'
MYDATAL =00007
MYRESULT ='MARCH 01, 2020'
MYRESULTL=00014

/\ /\ /*33 C Y J - D A Y /\ /\ /\-----/*****
====> TEST 33: CUTIL00 CYJ-DAY MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='1981061'
MYDATAL =00007
MYRESULT ='MONDAY'
MYRESULTL=00006
'1981061'
'Monday'

/\ /\ /*34 C Y J - D O W /\ /\ /\-----/*****
====> TEST 34: CUTIL00 CYJ-DOW MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='1980061'
MYDATAL =00007
MYRESULT ='6'
MYRESULTL=00001
'1980061'
'6'

/\ /\ /*35 C Y J - M D C Y /\ /\ /\-----/*****
====> TEST 35: CUTIL00 CYJ-MDCY MYDATA MYRESULT

RC =0

ERRMSG =SUCCESSFUL REQUEST
MYDATA ='2020061'
MYDATAL =00007
MYRESULT ='03012020'
MYRESULTL=00008

/\ /\ /*36 J C Y - D 8 /\ /\ /\-----/*****/
====> TEST 36: CUTIL00 JCY-D8 MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='1612017'
MYDATAL =00007
MYRESULT ='JUNE 10, 2017'
MYRESULTL=00013

/\ /\ /*37 J C Y - D A Y /\ /\ /\-----/*****/
====> TEST 37: CUTIL00 JCY-DAY MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='0611981'
MYDATAL =00007
MYRESULT ='MONDAY'
MYRESULTL=00006
'0611981'
'Monday'

/\ /\ /*38 J C Y - D O W /\ /\ /\-----/*****/
====> TEST 38: CUTIL00 JCY-DOW MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='0611981'
MYDATAL =00007
MYRESULT ='1'
MYRESULTL=00001
'0611981'
'1'

/\ /\ /*39 J C Y - M D C Y /\ /\ /\-----/*****/
====> TEST 39: CUTIL00 JCY-MDCY MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='1612017'
MYDATAL =00007
MYRESULT ='06102017'
MYRESULTL=00008

/\ /\ /*40 M D C Y - D 8 /\ /\ /\-----/*****/
====> TEST 40: CUTIL00 MDCY-D8 MYDATA MYRESULT

RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='12101990'
MYDATAL =00008
MYRESULT ='DECEMBER 10, 1990'
MYRESULTL=00017

/\ /\ /*41 M D C Y - D A Y /\ /\ /\-----/*****/

```
====> TEST 41: CUTIL00 MDCY-DAY MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='12101990'
MYDATAL     =00008
MYRESULT    ='MONDAY'
MYRESULTTL  =00006
'12101990'
'Monday'

/\ /\ /\*42      M D C Y - D O W                /\ /\ /\-----/*****
*/

====> TEST 42: CUTIL00 MDCY-DOW MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='12101990'
MYDATAL     =00008
MYRESULT    ='1'
MYRESULTTL  =00001
'12101990'
'1'

/\ /\ /\*43      M D C Y - C Y J                /\ /\ /\-----/*****/

====> TEST 43: CUTIL00 MDCY-CYJ  MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='12101990'
MYDATAL     =00008
MYRESULT    ='1990344'
MYRESULTTL  =00007

/\ /\ /\*44      D M C Y - D 8                  /\ /\ /\-----/*****/

====> TEST 44: CUTIL00 DMCY-D8  MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='15071987'
MYDATAL     =00008
MYRESULT    ='JULY 15, 1987'
MYRESULTTL  =00013

/\ /\ /\*45      D M C Y - D A Y                /\ /\ /\-----/*****/

====> TEST 45: CUTIL00 DMCY-DAY MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='15071987'
MYDATAL     =00008
MYRESULT    ='WEDNESDAY'
MYRESULTTL  =00009
'15071987'
'Wednesday'

/\ /\ /\*46      D M C Y - D O W                /\ /\ /\-----/*****/

====> TEST 46: CUTIL00 DMCY-DOW MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='15071987'
```

MYDATAL =00008
MYRESULT ='3'
MYRESULTL=00001
'15071987'
'3'

/\ /\ /*47 D M C Y - C Y J /\ /\ /\-----/*****/
====> TEST 47: CUTIL00 DMCY-CYJ MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='15071987'
MYDATAL =00008
MYRESULT ='1987196'
MYRESULTL=00007

/\ /\ /*48 C Y M D - D 8 /\ /\ /\-----/*****/
====> TEST 48: CUTIL00 CYMD-D8 MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951231'
MYDATAL =00008
MYRESULT ='DECEMBER 31, 1995'
MYRESULTL=00017

/\ /\ /*49 C Y M D - D A Y /\ /\ /\-----/*****/
====> TEST 49: CUTIL00 CYMD-DAY MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951231'
MYDATAL =00008
MYRESULT ='SUNDAY'
MYRESULTL=00006
'19951231'
'Sunday'

/\ /\ /*50 C Y M D - D O W /\ /\ /\-----/*****/
====> TEST 50: CUTIL00 CYMD-DOW MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951231'
MYDATAL =00008
MYRESULT ='0'
MYRESULTL=00001
'19951231'
'0'

/\ /\ /*51 C Y M D - C Y J /\ /\ /\-----/*****/
====> TEST 51: CUTIL00 CYMD-CYJ MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951231'
MYDATAL =00008
MYRESULT ='1995365'
MYRESULTL=00007

/\ /\ /*52 C Y D M - D 8 /\ /\ /\-----/*****/

```
====> TEST 52: CUTIL00 CYDM-D8 MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951605'
MYDATAL =00008
MYRESULT ='MAY 16, 1995'
MYRESULTL=00012
```

```
/\ /\ /\*53 C Y D M - D A Y /\ /\ /\-----/*****/
```

```
====> TEST 53: CUTIL00 CYDM-DAY MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951605'
MYDATAL =00008
MYRESULT ='TUESDAY'
MYRESULTL=00007
'19951605'
'Tuesday'
```

```
/\ /\ /\*54 C Y D M - D O W /\ /\ /\-----/*****/
```

```
====> TEST 54: CUTIL00 CYDM-DOW MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951605'
MYDATAL =00008
MYRESULT ='2'
MYRESULTL=00001
'19951605'
'2'
```

```
/\ /\ /\*55 C Y D M - C Y J /\ /\ /\-----/*****/
```

```
====> TEST 55: CUTIL00 CYDM-CYJ MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='19951605'
MYDATAL =00008
MYRESULT ='1995136'
MYRESULTL=00007
```

```
/\ /\ /\*56 F I L L /\ /\ /\-----/*****/
```

```
====> TEST 56: CUTIL00 FILL MYDATA MYPARMS
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='*****'
MYDATAL =00020
MYPARMS ='*20'
MYPARMSL =00003
```

```
/\ /\ /\*57 L S T R I P /\ /\ /\-----/*****/
```

```
====> TEST 57: CUTIL00 LSTRIP MYDATA MYPARMS
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='28,BC,'
MYDATAL =00006
MYPARMS =','
MYPARMSL =00001
```

```
/\ /\ /\*58  R S T R I P                /\ /\ /\-----/*****/
====> TEST 58: CUTIL00 RSTRIP    MYDATA MYPARMS
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      =' , , 28, BC'
MYDATAL     =00007
MYPARMS     =' , '
MYPARMSL    =00001

/\ /\ /\*59  S T R I P                /\ /\ /\-----/*****/
====> TEST 59: CUTIL00 STRIP    MYDATA MYPARMS
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      =' 28, BC'
MYDATAL     =00005
MYPARMS     =' , '
MYPARMSL    =00001

/\ /\ /\*60  S T R I P $                /\ /\ /\-----/*****/
====> TEST 60: CUTIL00 STRIP$   MYDATA MYPARMS
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      =' 28BC'
MYDATAL     =00004
MYPARMS     =' , '
MYPARMSL    =00001

/\ /\ /\*61  C O N C A T                /\ /\ /\-----/*****/
====> TEST 61: CUTIL00 CONCAT   MYDATA MYRESULT
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='STRING DATA 1.A RESULT STRING 2.'
MYDATAL     =00032
MYRESULT    ='A RESULT STRING 2.'
MYRESULTL   =00018

/\ /\ /\*62  U N S T R                /\ /\ /\-----/*****/
====> TEST 62: CUTIL00 UNSTR    MYDATA MYPARMS
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='1234567 AND SO ON . I AM 78 LONG. '
MYDATAL     =00034
MYPARMS     ='7'
MYPARMSL    =00001
MYDATA0     ='003'
MYDATA1     ='123456'
MYDATA2     =' AND SO ON . I AM '
MYDATA3     ='8 LONG. '

/\ /\ /\*63  R E P L A C E                /\ /\ /\-----/*****/
====> TEST 63: CUTIL00 REPLACE  MYDATA MYPARMS
RC          =0
ERRMSG      =SUCCESSFUL REQUEST
MYDATA      ='1234567 AND 890123456--. I AM 7879.'
MYDATAL     =00035
```

MYPARMS ='SO ON ", "890123456--"
MYPARMSL =00022

/\ /\ /\ *64 V E X I S T /\ /\ /\ -----/*****/
====> TEST 64: CUTIL00 VEXIST MYDATA
RC =4
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='TEST'
MYDATA =00004
MYRESULT =''
MYRESULTL=

/\ /\ /\ *65 T D S N /\ /\ /\ -----/*****/
====> TEST 65: CUTIL00 TDSN MYDATA
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='HERC01.CARDS.D20103.T164904'
MYDATA =00027
MYRESULT =''
MYRESULTL=

/\ /\ /\ *66 N O W /\ /\ /\ -----/*****/
====> TEST 66: CUTIL00 NOW MYDATA
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='04/12/2020.103 16:49:04.64 0 SUNDAY'
MYDATA =00038
MYRESULT =''
MYRESULTL=

/\ /\ /\ *67 P A D /\ /\ /\ -----/*****/
====> TEST 67: CUTIL00 PAD MYDATA MYRESULT
RC =0
ERRMSG =SUCCESSFUL REQUEST
MYDATA ='PAD ME TO 30 LONG'
MYDATA =00030
MYRESULT ='00003'
MYRESULTL=00003

***** TESTING CUTIL00 CONCLUDED. *** 04/12/20 16:49:04
END

```
CCUTIL00 ABOUT VERBOSE(N)
CUTIL00 MVS3.8J V1.0.00 03202020 04/12/20 @ 16.49
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   =' '
VAR01L  =
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ABOUT

CCUTIL00 LTRIM      VAR01('  LEADING SPACES') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='  LEADING SPACES'
VAR01L  =00017
VAR02   ='LEADING SPACES'
VAR02L  =00014
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! LTRIM

CCUTIL00 RTRIM      VAR01('TRAILING SPACES ') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='TRAILING SPACES '
VAR01L  =00017
VAR02   ='TRAILING SPACES'
VAR02L  =00015
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! RTRIM

CCUTIL00 TRIM        VAR01(' TRIM  MY DATA ') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   =' TRIM  MY DATA '
VAR01L  =00017
VAR02   ='TRIM  MY DATA'
VAR02L  =00015
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! TRIM

CCUTIL00 TRIM$        VAR01(' TRIM  MY DATA ') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   =' TRIM  MY DATA '
VAR01L  =00017
VAR02   ='TRIM MY DATA'
VAR02L  =00012
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! TRIM$

CCUTIL00 INDEX        VAR01('MY, STRING, OK? ') VAR02('","') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =3 SUCCESSFUL INDEX
VAR01   ='MY, STRING, OK? '
VAR01L  =00018
VAR02   ='","'
VAR02L  =00003
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! INDEX

CCUTIL00 INDEXB        VAR01('MY, STRING, OK? ') VAR02('","') VERBOSE(N)
```

```
--- CUTIL00 RESPONSE .....
RC      =11 SUCCESSFUL INDEXB
VAR01   ='MY, STRING, OK?'
VAR01L  =00018
VAR02   ='',''
VAR02L  =00003
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  INDEXB

CCUTIL00 ISNUM      VAR01('0123456789') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
VAR01   ='0123456789'
VAR01L  =00010
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  ISNUM

CCUTIL00 ISALPHA    VAR01('ABCDEFGHIJ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
VAR01   ='ABCDEFGHIJ'
VAR01L  =00010
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  ISALPHA

CCUTIL00 ISLOWER    VAR01('abcdefghij') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4000 FALSE
VAR01   ='ABCDEFGHIJ'
VAR01L  =00010
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  ISLOWER

CCUTIL00 ISUPPER    VAR01('ABCDEFGHIJ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
VAR01   ='ABCDEFGHIJ'
VAR01L  =00010
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  ISUPPER

CCUTIL00 ISBLANK    VAR01(' ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
VAR01   =' '
VAR01L  =00009
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  ISBLANK

CCUTIL00 ISDSN      VAR01('A.A1234567') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
```

```
VAR01 ='A.A1234567'
VAR01L=00010
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ISDSN

CCUTIL00 ECHO      VAR01('MY DATA TO DISPLAY') VERBOSE(N)
MY DATA TO DISPLAY
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='MY DATA TO DISPLAY'
VAR01L=00018
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ECHO

CCUTIL00 ECHOQ     VAR01('MY DATA TO DISPLAY') VERBOSE(N)
'MY DATA TO DISPLAY'
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='MY DATA TO DISPLAY'
VAR01L=00018
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ECHOQ

CCUTIL00 REVRS     VAR01('ABCDEFGHIJKLM96%QRS') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='ABCDEFGHIJKLM96%QRS'
VAR01L=00019
VAR02 ='ZYXWVUTSRQPON03 JIH'
VAR02L=00019
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! REVRS

CCUTIL00 UPPERC    VAR01('ABCDEFGHIJKLMNOPQRS') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='ABCDEFGHIJKLMNOPQRS'
VAR01L=00019
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! UPPERC

CCUTIL00 LOWERC    VAR01('ABCDEFGHIJKLMNOPQRS') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='ABCDEFGHIJKLMNOPQRS'
VAR01L=00019
VAR02 ='ABCDEFGHIJKLMNOPQRS'
VAR02L=00019
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! LOWERC

CCUTIL00 COUNT     VAR01('MY NAME IS DINO, BIG DINO.') VAR02('"DINO"') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =2 SUCCESSFUL COUNT
```

```
VAR01 ='MY NAME IS DINO, BIG DINO.'  
VAR01L=00026  
VAR02 =' "DINO"  
VAR02L=00006  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! COUNT  
  
CCUTIL00 FIND      VAR01('MY NAME IS DINO, BIG DINO.') VAR02(' "DINO"') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =12 SUCCESSFUL FIND  
VAR01 ='MY NAME IS DINO, BIG DINO.'  
VAR01L=00026  
VAR02 =' "DINO"  
VAR02L=00006  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! FIND  
  
CCUTIL00 FINDL     VAR01('MY NAME IS DINO, BIG DINO.') VAR02(' "DINO"') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =22 SUCCESSFUL FINDL  
VAR01 ='MY NAME IS DINO, BIG DINO.'  
VAR01L=00026  
VAR02 =' "DINO"  
VAR02L=00006  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! FINDL  
  
CCUTIL00 CENTER    VAR01('CENTER THIS T E X T ... ') VAR02('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =0 SUCCESSFUL REQUEST  
VAR01 ='CENTER THIS T E X T ... '  
VAR01L=00030  
VAR02 ='  CENTER THIS T E X T ... '  
VAR02L=00030  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! CENTER  
  
CCUTIL00 LJUST     VAR01(' JUSTIFY THIS TEXT ') VAR02('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =0 SUCCESSFUL REQUEST  
VAR01 =' JUSTIFY THIS TEXT '  
VAR01L=00022  
VAR02 ='JUSTIFY THIS TEXT '  
VAR02L=00022  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! LJUST  
  
CCUTIL00 RJUST     VAR01(' JUSTIFY THIS TEXT ') VAR02('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =0 SUCCESSFUL REQUEST  
VAR01 =' JUSTIFY THIS TEXT '  
VAR01L=00022  
VAR02 =' JUSTIFY THIS TEXT '  
VAR02L=00022  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! RJUST  
  
CCUTIL00 ZFILL     VAR01('12 ') VAR02('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC      =0 SUCCESSFUL REQUEST  
VAR01 ='12 '  
VAR01L=00005
```

```
VAR02 ='00012'  
VAR02L=00005  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ZFILL  
  
CCUTIL00 WORDS VAR01(' ONE TEWO THREE33 FOUR., FIVE ') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =5 SUCCESSFUL WORDS  
VAR01 =' ONE TEWO THREE33 FOUR., FIVE '  
VAR01L=00031  
VAR02 =' '  
VAR02L=  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! WORDS  
  
CCUTIL00 GEN# VAR01('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =0 SUCCESSFUL REQUEST  
VAR01 ='559'  
VAR01L=00003  
VAR02 =' '  
VAR02L=  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! GEN#  
  
CCUTIL00 DD2DSN VAR01('SYSPROC') VAR02('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =0 SUCCESSFUL REQUEST  
VAR01 ='SYSPROC'  
VAR01L=00007  
VAR02 ='SYS2.CMDPROC'  
VAR02L=00012  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! DD2DSN  
  
CCUTIL00 JOBINFO VAR01('') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =0 SUCCESSFUL REQUEST  
VAR01 ='CUTIL001STEP01 TESTIT1 '  
VAR01L=00024  
VAR02 =' '  
VAR02L=  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! JOBINFO  
  
CCUTIL00 DAYSMM VAR01('022016') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =29 SUCCESSFUL DAYSMM  
VAR01 ='022016'  
VAR01L=00006  
VAR02 =' '  
VAR02L=  
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! DAYSMM  
  
CCUTIL00 DAYSYY VAR01('2020') VERBOSE(N)  
--- CUTIL00 RESPONSE .....  
RC =366 SUCCESSFUL DAYSYY  
VAR01 ='2020'  
VAR01L=00004  
VAR02 =' '  
VAR02L=
```

```
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! DAYSYY

CCUTIL00 ISLEAP  VAR01('2020') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4001 TRUE
VAR01   ='2020'
VAR01L  =00004
VAR02   =' '
VAR02L  =
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! ISLEAP

CCUTIL00 CYJ-D8  VAR01('2020061') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='2020061'
VAR01L  =00007
VAR02   ='MARCH 01, 2020'
VAR02L  =00014
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! CYJ-D8

CCUTIL00 CYJ-DAY VAR01('1981061') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='1981061'
VAR01L  =00007
VAR02   ='MONDAY'
VAR02L  =00006
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! CYJ-DAY

CCUTIL00 CYJ-DOW VAR01('1981061') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='1981061'
VAR01L  =00007
VAR02   ='1'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! CYJ-DOW

CCUTIL00 CYJ-MDCY VAR01('2020061') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='2020061'
VAR01L  =00007
VAR02   ='03012020'
VAR02L  =00008
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! CYJ-MDCY

CCUTIL00 JCY-D8  VAR01('1612017') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='1612017'
VAR01L  =00007
VAR02   ='JUNE 10, 2017'
VAR02L  =00013
```

```
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! JCY-D8

CCUTIL00 JCY-DAY VAR01('0611981') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='0611981'
VAR01L  =00007
VAR02   ='MONDAY'
VAR02L  =00006
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! JCY-DAY

CCUTIL00 JCY-DOW VAR01('0611981') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='0611981'
VAR01L  =00007
VAR02   ='1'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! JCY-DOW

CCUTIL00 JCY-MDCY VAR01('1612017') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='1612017'
VAR01L  =00007
VAR02   ='06102017'
VAR02L  =00008
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! JCY-MDCY

CCUTIL00 MDCY-D8 VAR01('12101990') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='12101990'
VAR01L  =00008
VAR02   ='DECEMBER 10, 1990'
VAR02L  =00017
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! MDCY-D8

CCUTIL00 MDCY-DAY VAR01('12101990') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='12101990'
VAR01L  =00008
VAR02   ='MONDAY'
VAR02L  =00006
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! MDCY-DAY

CCUTIL00 MDCY-DOW VAR01('12101990') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='12101990'
VAR01L  =00008
VAR02   ='1'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE! MDCY-DOW
```

```
CCUTIL00 MDCY-CYJ VAR01('12101990') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='12101990'
VAR01L  =00008
VAR02   ='1990344'
VAR02L  =00007
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  MDCY-CYJ

CCUTIL00 DMCY-D8  VAR01('15071987') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='15071987'
VAR01L  =00008
VAR02   ='JULY 15, 1987'
VAR02L  =00013
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  DMCY-D8

CCUTIL00 DMCY-DAY VAR01('15071987') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='15071987'
VAR01L  =00008
VAR02   ='WEDNESDAY'
VAR02L  =00009
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  DMCY-DAY

CCUTIL00 DMCY-DOW VAR01('15071987') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='15071987'
VAR01L  =00008
VAR02   ='3'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  DMCY-DOW

CCUTIL00 DMCY-CYJ VAR01('15071987') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='15071987'
VAR01L  =00008
VAR02   ='1987196'
VAR02L  =00007
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  DMCY-CYJ

CCUTIL00 CYMD-D8  VAR01('19951231') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951231'
VAR01L  =00008
VAR02   ='DECEMBER 31, 1995'
VAR02L  =00017
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYMD-D8
```

```
CCUTIL00 CYMD-DAY VAR01('19951231') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951231'
VAR01L  =00008
VAR02   ='SUNDAY'
VAR02L  =00006
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYMD-DAY
```

```
CCUTIL00 CYMD-DOW VAR01('19951231') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951231'
VAR01L  =00008
VAR02   ='0'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYMD-DOW
```

```
CCUTIL00 CYMD-CYJ VAR01('19951231') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951231'
VAR01L  =00008
VAR02   ='1995365'
VAR02L  =00007
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYMD-CYJ
```

```
CCUTIL00 CYDM-D8  VAR01('19951605') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951605'
VAR01L  =00008
VAR02   ='MAY 16, 1995'
VAR02L  =00012
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYDM-D8
```

```
CCUTIL00 CYDM-DAY VAR01('19951605') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951605'
VAR01L  =00008
VAR02   ='TUESDAY'
VAR02L  =00007
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYDM-DAY
```

```
CCUTIL00 CYDM-DOW VAR01('19951605') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951605'
VAR01L  =00008
VAR02   ='2'
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYDM-DOW
```

```
CCUTIL00 CYDM-CYJ VAR01('19951605') VAR02('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='19951605'
VAR01L  =00008
VAR02   ='1995136'
VAR02L  =00007
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CYDM-CYJ

CCUTIL00 FILL      VAR01('') VAR02('*20') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='*****'
VAR01L  =00020
VAR02   ='*20'
VAR02L  =00003
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  FILL

CCUTIL00 LSTRIP    VAR01(', ,28,BC,') VAR02(', ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='28,BC,'
VAR01L  =00006
VAR02   =', '
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  LSTRIP

CCUTIL00 RSTRIP    VAR01(', ,28,BC,') VAR02(', ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   =', ,28,BC'
VAR01L  =00007
VAR02   =', '
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  RSTRIP

CCUTIL00 STRIP     VAR01(', ,28,BC,') VAR02(', ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='28,BC'
VAR01L  =00005
VAR02   =', '
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  STRIP

CCUTIL00 STRIP$    VAR01(', ,28,BC,') VAR02(', ') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='28BC'
VAR01L  =00004
VAR02   =', '
VAR02L  =00001
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  STRIP$

CCUTIL00 CONCAT    VAR01('STRING DATA 1.') VAR02('A RESULT STRING 2.') VERBOSE(N)
--- CUTIL00 RESPONSE .....
```

```
RC      =0 SUCCESSFUL REQUEST
VAR01 ='STRING DATA 1.A RESULT STRING 2.'
VAR01L=00032
VAR02 ='A RESULT STRING 2.'
VAR02L=00018
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  CONCAT

CCUTIL00 UNSTR      VAR01('1234567 AND SO ON . I AM 78 LONG. ')  VAR02('7') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='1234567 AND SO ON . I AM 78 LONG. '
VAR01L=00034
VAR02 ='7'
VAR02L=00001
- UNSTR VARIABLE(S) CREATED: 003
VAR010 = '003'
VAR011 = '123456'
VAR012 = ' AND SO ON . I AM '
VAR013 = '8 LONG. '
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  UNSTR

CCUTIL00 REPLACE  VAR01('1234567 AND SO ON . I AM 7879.')  VAR02('"SO ON ", "890123456--"') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='1234567 AND 890123456--. I AM 7879.'
VAR01L=00035
VAR02 =' "SO ON ", "890123456--" '
VAR02L=00022
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  REPLACE

CCUTIL00 VEXIST   VAR01('VAR01') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =5 SUCCESSFUL REQUEST
VAR01 ='VAR01'
VAR01L=00005
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  VEXIST

CCUTIL00 VEXIST   VAR01('DDD01') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =4000 FALSE
VAR01 ='DDD01'
VAR01L=
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  VEXIST

CCUTIL00 TDSN     VAR01('HERC01.CARDS.') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01 ='HERC01.CARDS.D20103.T164906'
VAR01L=00027
VAR02 =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  TDSN
```

```
CCUTIL00 NOW      VAR01('') VERBOSE(N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='04/12/2020.103 16:49:06.97 0 SUNDAY   '
VAR01L=00038
VAR02   =' '
VAR02L=
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  NOW

CCUTIL00 PAD      VAR01('MAKE ME 30 LONG') VAR02(' 30') VERBOSE (N)
--- CUTIL00 RESPONSE .....
RC      =0 SUCCESSFUL REQUEST
VAR01   ='MAKE ME 30 LONG                      '
VAR01L=00030
VAR02   =' 30 '
VAR02L=00003
--- CCUTIL00: CUTIL00 TEST TOOL ... DONE!  PAD

END
```

				CCCCCCCCC	UU		UU	TTTTTTTTTTT	IIIIIIIII	LL			00000000	00000000	11	
				CCCCCCCCC	UU		UU	TTTTTTTTTTT	IIIIIIIII	LL			0000000000	0000000000	111	
			CC		CC	UU	UU	TT	II	LL		00	0000	00	0000	1111
			CC			UU	UU	TT	II	LL		00	00	00	00	11
			CC			UU	UU	TT	II	LL		00	00	00	00	11
			CC			UU	UU	TT	II	LL		00	00	00	00	11
			CC			UU	UU	TT	II	LL		00	00	00	00	11
			CC			UU	UU	TT	II	LL		00	00	00	00	11
			CC			UU	UU	TT	II	LL		0000	00	0000	00	11
			CC	CC	UU	UU	UU	TT	II	LL		000	00	000	00	11
			CCCCCCCCC	CCCCCCCCC	UUUUUUUUUUUU		TT	IIIIIIIII	LLLLLLLLLLLLL			0000000000	0000000000	111111111		
			CCCCCCCCC		UUUUUUUUUU		TT	IIIIIIIII	LLLLLLLLLLLLL			00000000	00000000	111111111		

				JJJJJJJJJ	333333333	11		00000000	11			AAAAAAAAA	
				JJJJJJJJJ	33333333333	111		0000000000	111			AAAAAAAAAAAA	
				JJ	33	33	1111	00	0000	1111		AA	AA
				JJ		33	11	00	00 00	11		AA	AA
				JJ		33	11	00	00 00	11		AA	AA
				JJ	3333	11	00	00 00	11			AAAAAAAAAAAA	
				JJ	3333	11	00	00 00	11			AAAAAAAAAAAA	
				JJ		33	11	00 00	00	11		AA	AA
			JJ	JJ		33	11	0000	00	11		AA	AA
			JJ	JJ	33	33	11	000	00	11		AA	AA
			JJJJJJJ	33333333333		111111111		0000000000		111111111		AA	AA
			JJJJJJ	333333333		111111111		00000000		111111111		AA	AA

****A	END	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	END	A****
****A	END	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	END	A****
****A	END	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	END	A****
****A	END	JOB 3101	CUTIL001	Install	CUTIL00	TS0	ROOM	4.49.07	PM 12	APR 20	PRINTER1	SYS BSP1	JOB 3101	END	A****