

# JCL and Programming On the MVS Turnkey System

This assumes you've already set up, run, connected to, and logged in to an MVS Turnkey system. If not, you can find instructions [here](#).

## About

A job to be run consists of two parts:

1. [JCL](#) to tell the mainframe how to run the program, and
2. The actual program, written in a language with a supporting compiler on the mainframe.

```
// JCL comes first, and is prefixed by two slashes  
The program source follows.
```



The JCL message class must be set to 'H', or you won't be able to see the output from your jobs.

The 'H' indicates to the system that the output should be 'Held', making it available for viewing in the OUTLIST utility, in Data Set Utilities.

## COBOL

COBOL first appeared in 1959. The latest stable release was in 2014. It's meant to be "English-like" in syntax.

COBOL is still widely deployed. For example, as of 2017, about 95 percent of ATM swipes use COBOL code, and it powers 80 percent of in-person transactions.

## Create and Submit the Job

Your starting point should be the main screen:

```
Terminal CUU0C0                               Date 03.12.21
System  TK4-                                  Time 00:58:19
TSO User HERC01

Option ==> 

      The MVS 3.8j Tur(n)key System
TK4- Version 1.00 Update 08 -- MVS PUT 8505

      TSO Applications

1 RFE      "SPF like" productivity tool
2 RPF      "SPF like" productivity tool
3 IM       IMON/370 system monitor
4 QUEUE    spool browser
5 HELP     general TSO help
6 UTILS    information on utilities and commands available
7 TERMTEST verify 3270 terminal capabilities

      Enter X to Terminate

PF3=Terminate
```

Enter '1' to access the RFE tool:

```
Terminal CUU0C0                               Date 03.12.21
System  TK4-                                  Time 00:58:19
TSO User HERC01

Option ==> 1 

      The MVS 3.8j Tur(n)key System
TK4- Version 1.00 Update 08 -- MVS PUT 8505

      TSO Applications

1 RFE      "SPF like" productivity tool
2 RPF      "SPF like" productivity tool
3 IM       IMON/370 system monitor
4 QUEUE    spool browser
5 HELP     general TSO help
6 UTILS    information on utilities and commands available
7 TERMTEST verify 3270 terminal capabilities

      Enter X to Terminate

PF3=Terminate
```

Enter '3' to access utility functions:

```
----- REVIEW FRONT END -----  
COMMAND ===> 3  
  
1  BROWSE   - VIEW OR BROWSE DATA SET CONTENTS      USERID   - HERC01  
2  EDIT     - UPDATE OR CREATE DATA SET CONTENTS    SYSTEM    - TK4-  
3  UTILITIES - PERFORM UTILITY FUNCTIONS             TERMINAL  - CUU0C0  
6  COMMAND  - ISSUE TSO OR CLIST COMMAND             NETWORK   -  
X  EXIT     - TERMINATE RFE                          RELEASE   - 46.6  
                                           DAY       - FRI      337  
                                           DATE      - 2021-12-03  
                                           TIME      - 00:58
```

Enter '4' to access the Data Set list:

```
----- DATA SET UTILITIES -----  
COMMAND ===> 4  
  
1  LIBRARY   - PDS COMPRESS AND MEMBER MANAGEMENT  
2  DATASET  - CREATE, DELETE, RENAME, CATALOG OR UNCATALOG DATA SET  
3  MOVE/COPY - MOVE OR COPY PDS MEMBERS OR DATA SET CONTENTS  
4  DSLIST   - PROCESS DATA SETS FROM A CATALOG OR VTOC LIST  
5  SPFSTATS - ADMINISTER STATISTICS OF LIBRARY MEMBERS  
8  OUTLIST  - DISPLAY, DELETE OR PRINT HELD JOB OUTPUT
```

Enter 'SYS2.JCLLIB' to filter the Data Set list:

```
----- RFE DSLIST -----  
COMMAND ==>  
  
blank - display data set list  
ALLOC - allocate a new data set  
  
Data set name prefix ==> SYS2.JCLLIB  
Volume serial number ==>  
  
Data set selection codes  
  
A - Allocate like   B - Browse           C - Catalog  
D - Delete         E - Edit            I - Info  
L - Listcat        R - Rename         S - Short info  
U - Uncatalog      V - View           Z - Compress
```

Tab to the detail line for SYS2.JCLLIB, then enter 'e' in the S column, for Edit:

```
----- RFE DSLIST ----- Row 1 of 2  
Command ==> Scroll ==> CS  
S DATA-SET-NAME- VOLUME ALTRK USTRK  ORG  FRMT  %  XT  LRECL BLKSZ REFDT CREDIT EXPDT  
e SYS2.JCLLIB     MVSRES   60   47  PO  FB   78  1   80  19040 21337 13314  
**END**          TOTALS:   60 TRKS ALLOC          47 TRKS USED          1 EXTENTS
```

The Data Set list will display:

```
SYS2.JCLLIB on MVSRES ----- Row 1 of 115
Command ==> [ ] Scroll ==> CS
```

NAME	TTR	VV.MM	CREATED	CHANGED	INIT	SIZE	MOD	ID
.\$\$\$INDEX	002E01	01.02	14-11-12	16-09-17 14:55:58	135	138	0	JUERGEN
.\$HISTORY	000103							
.\$ADDALIAS	000105							
.\$ADDUSER	000B0D	80.02	74-06-28	74-06-28 13:25:00	79	17	0	HERC01
.\$ADDUSERP	000B11	80.02	74-06-28	02-07-14 20:20:00	17	17	0	HERC01
.\$ALGSAMP1	001303	02.01	14-12-18	14-12-18 12:00:00	84	84	0	LEVEL
.\$ALGSAMP2	002C05	02.01	14-12-18	14-12-18 12:00:00	57	57	0	LEVEL
.\$ALGSAMP3	002C03	02.01	14-12-18	14-12-18 12:00:00	100	100	0	LEVEL
.\$ALGSAMP4	001401	02.01	14-12-18	14-12-18 12:00:00	896	896	0	LEVEL
.\$ALLALIAS	000D0B							
.\$AMASPZAP	000107							
.\$AMDPRDMP	000109							
.\$BAT#EDIT	000201							
.\$BSPHRCMD	000203	01.06	13-10-02	13-10-06 11:36:03	8	26	0	HERC01
.\$BSPOSCMD	000205							
.\$BSPVTMT	000A0D	80.00	73-12-23	73-12-23 14:06:00	4	4	0	HERC02
.\$BYPASSNQ	000207							
.\$CHGPWD	000B0F	80.02	74-06-28	74-06-28 14:18:00	17	23	0	HERC01
.\$CLIPDASD	000209							
.\$COMPPROC	000301							
.\$COMPRESS	000303							
.\$CONFIGSE	000A09	80.01	73-06-16	73-06-16 12:48:00	29	14	0	HERC01
.\$COPY001	000B01	00.00	74-06-14	74-06-15 14:01:00	01	01	0	HERC01

If you press [F8] (page down) a few times, you'll see several Data Sets with names that begin with 'TEST'. These are test programs for various languages:

```

. TAPEMAP 000A05
. TESTALG 001301
. TESTCOB 000E03
. TESTFORT 000E05
. TESTGCC 00110B 01.00 13-12-22 13-12-22 20:30:00 37 37 0 JUERGEN
. TESTJCC 001801 01.00 14-11-12 14-11-12 17:45:00 37 37 0 TK4-
. TESTPL1 000E07
. TESTRPG 000E09
. TESTSIMU 001201 01.00 14-01-05 14-01-05 06:00:00 151 151 0 JUERGEN
. TESTSORT 000F01
. TESTWATF 001703 01.00 14-09-30 14-09-30 22:00:00 19 19 0 JUERGEN
. TSOBATCH 000902

```

We'll be using TESTCOB as a template for our COBOL job. We don't want to use it directly, creating a copy instead. The first thing to do is to create a new, empty Data Set, with the name NEWCOB:

```
SYS2.JCLLIB on MVSRES ----- Row 81 of 115
Command ==> s NEWCOB Scroll ==> CS
```

NAME	TTR	YY.MM	CREATED	CHANGED	INIT	SIZE	MOD	ID
. PTPCH001	000B03	80.02	74-06-14	74-06-15 14:21:00	19	19	0	HERC01
. PTPCH002	000B05	80.01	74-06-15	74-06-15 23:40:00	19	19	0	HERC01
. PUNPDSXM	000701							
. PWDPRINT	000C09	80.03	02-07-22	02-07-22 11:02:00	19	24	0	HERC01
. RD#PDSXM	000703							
. READOMAT	000901							
. RECV370P	000801	01.07	13-10-01	13-10-06 12:16:23	53	69	0	HERC01
. RECV370S	000803							
. RENMEMBR	000805							
. REVLMOD	000A0F	80.01	73-12-23	73-12-23 14:08:00	4	4	0	HERC02
. SAVEOLD	000A01							
. SHUTDOWN	001109	01.00	13-07-15	13-07-15 22:07:00	13	13	0	SYSOPER
. STMRPTM	001202	01.00	14-01-01	14-01-01 19:20:00	65	65	0	THURGEN

The new, empty Data Set opens in REVEDIT.

Next, we indicate that we want to populate it with a copy of the contents of TESTCOB:

```
REVEDIT SYS2.JCLLIB(NEWCOB) - 1.00 COLUMNS 00001 00072
COMMAND ==> COPY TESTCOB Scroll ==> CS
***** ZAP ***** AUTOSAVE ***** TOP OF DATA *****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

The editor will display the copied text:

```
REVEDIT 33 LINE(S) COPIED                                COLUMNS 00001 00072
COMMAND ==> [ ]                                          SCROLL ==> CS
***** ZAP*****AUTOSAVE***** TOP OF DATA *****
000001 //TESTCOB JOB (SETUP),
000002 //          "TEST COBOL",
000003 //          CLASS=A,
000004 //          MSGCLASS=A,
000005 //          MSGLEVEL=(1,1)
000006 //*****
000007 //*
000008 //* NAME: SYS2.JCLLIB(TESTCOB)
000009 //*
000010 //* DESC: TEST COBOL INSTALLATION
000011 //*
000012 //*****
000013 //HELOWRLD EXEC COBUCLG
000014 //COB.SYSIN DD *
000015 001 IDENTIFICATION DIVISION.
000016 002 PROGRAM-ID. "HELLO".
000017 003 ENVIRONMENT DIVISION.
000018 004 CONFIGURATION SECTION.
000019 005 SOURCE-COMPUTER. IBM-360.
000020 006 OBJECT-COMPUTER. IBM-360.
000021 0065 SPECIAL-NAMES.
000022 0066     CONSOLE IS CNSL.
000023 007 DATA DIVISION.
000024 008 WORKING-STORAGE SECTION.
000025 009 77 HELLO-CONST PIC X(12) VALUE "HELLO, WORLD".
000026 075 PROCEDURE DIVISION.
000027 090 000-DISPLAY.
000028 100     DISPLAY HELLO-CONST UPON CNSL.
000029 110     STOP RUN.
000030 //LKED.SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR
000031 //          DD DSNAME=SYS1.LINKLIB,DISP=SHR
000032 //GO.SYSPRINT DD SYSOUT=A
000033 //
***** ZAP*****AUTOSAVE***** BOTTOM OF DATA *****
```

Make a few edits to the copied text. First, in line 0001, change TESTCOB to NEWCOB:

```
REVEDIT 33 LINE(S) COPIED                                COLUMNS 00001 00072
COMMAND ==> [ ]                                          SCROLL ==> CS
***** ZAP*****AUTOSAVE***** TOP OF DATA *****
000001 //NEWCOB JOB (SETUP),
000002 //          "TEST COBOL",
000003 //          CLASS=A,
000004 //          MSGCLASS=A,
000005 //          MSGLEVEL=(1,1)
```

In line 0002, update the description:

```
REVEDIT 33 LINE(S) COPIED COLUMNS 00001 00072
COMMAND ==> SCROLL ==> CS
***** ZAP*****AUTOSAVE***** TOP OF DATA *****
000001 //NEWCOB JOB (SETUP),
000002 // 'NEW COBOL',
000003 // CLASS=A,
000004 // MSGCLASS=A,
000005 // MSGLEVEL=(1,1)
```

Finally, in line 0004, change the MSGCLASS to 'H':

```
REVEDIT 33 LINE(S) COPIED COLUMNS 00001 00072
COMMAND ==> SCROLL ==> CS
***** ZAP*****AUTOSAVE***** TOP OF DATA *****
000001 //NEWCOB JOB (SETUP),
000002 // 'NEW COBOL',
000003 // CLASS=A,
000004 // MSGCLASS=H,
000005 // MSGLEVEL=(1,1)
000006 //*****
000007 //*
```

(This will ensure that the output from the job is retained and viewable after we run it)

Save your changes:

```
REVEDIT 33 LINE(S) COPIED COLUMNS 00001 00072
COMMAND ==> SAVE
***** ZAP*****AUTOSAVE***** TOP OF DATA *****
000001 //NEWCOB JOB (SETUP),
000002 // 'NEW COBOL',
000003 // CLASS=A,
000004 // MSGCLASS=H,
000005 // MSGLEVEL=(1,1)
000006 //*****
000007 //*
000008 //* NAME: SYS2.JCLLIB(TESTCOB)
000009 //*
000010 //* DESC: TEST COBOL INSTALLATION
000011 //*
000012 //*****
000013 //HEL OWPLD EXEC COBUCLG
```

Submit the job:



```

REVFDTT  DATA SAVED                                COLUMNS 00001 00072
COMMAND ==> SUBMIT
***** ZAP ***** AUTOSAVE ***** TOP OF DATA *****
000001 //NEWCOB JOB (SETUP),
000002 //          "NEW COBOL",
000003 //          CLASS=A,
000004 //          MSGCLASS=H,
000005 //          MSGLEVEL=(1,1)
000006 //*****
000007 //*
000008 //* NAME: SYS2.JCLLIB(TESTCOB)
000009 //*
000010 //* DESC: TEST COBOL INSTALLATION
000011 //*
000012 //*****
000013 //*****

```

You'll see a confirmation message, indicating that the job has been submitted:

```

JOB NEWCOB(JOB00006) SUBMITTED
***

```

### Check the Results

If you aren't already on the main screen, press [F3] until it's displayed:

```

Terminal CUU0C0                                     Date 03.12.21
System   TK4-                                       Time 00:58:19
TSO User HERC01

Option ==>

      The MVS 3.8j Tur(n)key System
TK4- Version 1.00 Update 08 -- MVS PUT 8505

      TSO Applications

1 RFE      "SPF like" productivity tool
2 RPF      "SPF like" productivity tool
3 IM       IMON/370 system monitor
4 QUEUE    spool browser
5 HELP     general TSO help
6 UTILS    information on utilities and commands available
7 TERMTEST verify 3270 terminal capabilities

Enter X to Terminate

PF3=Terminate

```

Enter '1' to access the RFE tool:

```
Terminal CUU0C0                               Date 03.12.21
System  TK4-                                  Time 00:58:19
TSO User HERC01

Option ==> 1 

      The MVS 3.8j Tur(n)key System
      TK4- Version 1.00 Update 08 -- MVS PUT 8505

      TSO Applications

1  RFE      "SPF like" productivity tool
2  RPF      "SPF like" productivity tool
3  IM       IMON/370 system monitor
4  QUEUE    spool browser
5  HELP     general TSO help
6  UTILS    information on utilities and commands available
7  TERMTEST verify 3270 terminal capabilities

      Enter X to Terminate

PF3=Terminate
```

Enter '3' to access utility functions:

```
----- REVIEW FRONT END -----
COMMAND ==> 3 

1  BROWSE   - VIEW OR BROWSE DATA SET CONTENTS
2  EDIT     - UPDATE OR CREATE DATA SET CONTENTS
3  UTILITIES - PERFORM UTILITY FUNCTIONS
6  COMMAND  - ISSUE TSO OR CLIST COMMAND
X  EXIT     - TERMINATE RFE

USERID      - HERC01
SYSTEM      - TK4-
TERMINAL    - CUU0C0
NETWORK     -
RELEASE     - 46.6
DAY         - FRI      337
DATE       - 2021-12-03
TIME       - 00:58
```

Enter '8' to access held job output:

```

----- DATA SET UTILITIES -----
COMMAND ===> 8

1  LIBRARY   - PDS COMPRESS AND MEMBER MANAGEMENT
2  DATASET  - CREATE, DELETE, RENAME, CATALOG OR UNCATALOG DATA SET
3  MOVE/COPY - MOVE OR COPY PDS MEMBERS OR DATA SET CONTENTS
4  DSLIST   - PROCESS DATA SETS FROM A CATALOG OR VTOC LIST
5  SPFSTATS - ADMINISTER STATISTICS OF LIBRARY MEMBERS
8  OUTLIST  - DISPLAY, DELETE OR PRINT HELD JOB OUTPUT

```

Enter 'ST \*', indicating that you want to display all held jobs:

```

REYOUT                                     LINE 1 OF 1
COMMAND ===> ST *                         SCROLL ===> CS
S Q JOBNAME  JOBIDENT  QUEUE   EXEC STATUS   STEPNAME  PROCSTEP  CPU-TIME
' ' HERC01    TSU00006  XEQ @   TK4-         TSOLOGON                1.10

```

You'll see 'NEWCOB', the job you recently submitted, at the end of the list:

```

REYOUT *                                     LINE 1 OF 17
COMMAND ===>                               SCROLL ===> CS
S Q JOBNAME  JOBIDENT  QUEUE   EXEC STATUS   STEPNAME  PROCSTEP  CPU-TIME
' ' SYSLOG    STC00088  XEQ $   TK4-                0.99
' ' BSPPILOT  STC00089  XEQ $   TK4-        BSPPILOT  C3P0      0.08
' ' INIT      STC00090  XEQ $   TK4-        INIT      IEFPROC   0.00
' ' INIT      STC00091  XEQ $   TK4-        INIT      IEFPROC   0.00
' ' INIT      STC00092  XEQ $   TK4-        INIT      IEFPROC   0.00
' ' INIT      STC00093  XEQ $   TK4-        INIT      IEFPROC   0.00
' ' INIT      STC00094  XEQ $   TK4-        INIT      IEFPROC   0.00
' ' INIT      STC00095  XEQ $   TK4-        INIT      IEFPROC   0.01
' ' NET       STC00098  XEQ $   TK4-        NET       IEFPROC   0.32
' ' TP        STC00099  XEQ $   TK4-        TP        TCAM      0.69
' ' MF1       STC00100  XEQ $   TK4-        MF1       IEFPROC   0.04
' ' TSO       STC00101  XEQ $   TK4-        TSO       STEP1     0.01
' ' SNASOL    STC00102  XEQ $   TK4-        SNASOL    SOLICIT   0.59
' ' JRP       STC00103  XEQ $   TK4-        JRP       JRP       0.11
' ' HERC01    TSU00006  XEQ @   TK4-        TSOLOGON                1.11
' ' NEWJCC    JOB00003  PRTPUN
' ' NEWCOB    JOB00006  PRTPUN

```

Enter 'S' in the S column for the NEWCOB job:

```
REVOU*          LINE 1 OF 17
COMMAND ==>    SCROLL ==> CS
S Q JOBNAME  JOBIDENT QUEUE EXEC STATUS          STEPNAME PROCSTEP  CPU-TIME
' ' SYSLOG    STC00088 XEQ $  TK4-                BSPPILOT C3P0      1.01
' ' BSPPILOT  STC00089 XEQ $  TK4-                BSPPILOT C3P0      0.08
' ' INIT      STC00090 XEQ $  TK4-                INIT      IEFPROC      0.00
' ' INIT      STC00091 XEQ $  TK4-                INIT      IEFPROC      0.00
' ' INIT      STC00092 XEQ $  TK4-                INIT      IEFPROC      0.00
' ' INIT      STC00093 XEQ $  TK4-                INIT      IEFPROC      0.00
' ' INIT      STC00094 XEQ $  TK4-                INIT      IEFPROC      0.00
' ' INIT      STC00095 XEQ $  TK4-                INIT      IEFPROC      0.01
' ' NET       STC00098 XEQ $  TK4-                NET       IEFPROC      0.32
' ' TP        STC00099 XEQ $  TK4-                TP        TCAM          0.69
' ' MF1       STC00100 XEQ $  TK4-                MF1       IEFPROC      0.04
' ' TSO       STC00101 XEQ $  TK4-                TSO       STEP1         0.01
' ' SNASOL    STC00102 XEQ $  TK4-                SNASOL    SOLICIT       0.62
' ' JRP       STC00103 XEQ $  TK4-                JRP       JRP           0.12
' ' HERC01    TSU00006 XEQ @  TK4-                TSOLOGON  1.13
' ' NEWJCC    JOB00003 PRT PUN
S [ NEWCOB    JOB00006 PRT PUN
```

Job output is displayed:

```

SYS21337.T010558.RA000.HERC01.JOB00006 ----- Line 1 Col 2 81
Command ==> [ ] Scroll ==> CS
      10      20      30      40      50      60      70      80
-----+-----+-----+-----+-----+-----+-----+-----+
                                     J E S 2  J O B  L O G
01.03.49 JOB      6 IEF677I WARNING MESSAGE(S) FOR JOB NEWCOB  ISSUED
01.03.49 JOB      6 $HASP373 NEWCOB  STARTED - INIT 1 - CLASS A - SYS TK4-
01.03.49 JOB      6 IEF403I NEWCOB - STARTED - TIME=01.03.49
01.03.49 JOB      6 IEC130I SYSPUNCH DD STATEMENT MISSING
01.03.49 JOB      6 IEC130I SYSLIB  DD STATEMENT MISSING
01.03.49 JOB      6 IEC130I SYSPUNCH DD STATEMENT MISSING
01.03.49 JOB      6 IEFACRT - Stepname Procstep Program Retcode
01.03.49 JOB      6 NEWCOB  HELOWRLD COB      IKFCBL00 RC= 0000
01.03.49 JOB      6 NEWCOB  HELOWRLD LKED      IEWL      RC= 0000
01.03.49 JOB      6 +HELLO, WORLD
01.03.49 JOB      6 NEWCOB  HELOWRLD GO          PGM=*.DD RC= 0000
01.03.49 JOB      6 IEF404I NEWCOB - ENDED - TIME=01.03.49
01.03.49 JOB      6 $HASP395 NEWCOB  ENDED
  1  //NEWCOB JOB (SETUP),
    //      *NEW COBOL*,
    //      CLASS=A,
    //      MSGCLASS=H,
    //      MSGLEVEL=(1,1),
    //      USER=HERC01,PASSWORD=
                                     GENERATED BY GDL
*****
***
*** NAME: SYS2.JCLLIB(TESTCOB)
***
*** DESC: TEST COBOL INSTALLATION
***
*****
  2  //HELOWRLD EXEC COBUCLG
  3  XXCOBUCLG PROC SOUT="*"
  4  XXCOB EXEC PGM=IKFCBL00,
    XX      PARM="LOAD,SUPMAP,SIZE=2048K,BUF=1024K"
  5  XXSYSPRINT DD SYSOUT=&SOUT
  6  XXSYSUT1 DD UNIT=SYSDA,SPACE=(460,(700,100))
  7  XXSYSUT2 DD UNIT=SYSDA,SPACE=(460,(700,100))
  8  XXSYSUT3 DD UNIT=SYSDA,SPACE=(460,(700,100))
  9  XXSYSUT4 DD UNIT=SYSDA,SPACE=(460,(700,100))
 10  XXSYSLIN DD DSNAME=&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
    XX      SPACE=(80,(500,100))
                                     7828K FREE

```

Press [F8] to page down, and you'll see the 'Hello World' output:

```
SYS21337.T010558.RA000.HERC01.JOB00006 ----- Line 118 Col 2 81
Command ==> [ ] Scroll ==> CS
      10      20      30      40      50      60      70      80
-----+-----+-----+-----+-----+-----+-----+-----+
*****
*      2. Jobstep of job: NEWCOB      Stepname: LKED      Program name: IEWL
*      elapsed time 00:00:00,04      CPU-Identifier: TK4-
*      CPU time 00:00:00,02      Virtual Storage used: 264K
*      corr. CPU: 00:00:00,02      CPU time has been corrected by 1 / 1,0 m
*
*      I/O Operation
*      Number of records read via DD * or DD DATA: 0
*      140.....22 DMY.....0 180.....10 148.....17 148.....0 170.....0 DM
*
*      Charge for step (w/o SYSOUT):
*****
IEF236I ALLOC. FOR NEWCOB GO HELOWRLD
IEF237I 180 ALLOCATED TO PGM=*.DD
IEF237I JES2 ALLOCATED TO SYSPRINT
HELLO, WORLD
IEF142I NEWCOB GO HELOWRLD - STEP WAS EXECUTED - COND CODE 0000
IEF285I SYS21337.T010349.RA000.NEWCOB.GODATA KEPT *-----0
IEF285I VOL SER NOS= WORK02.
IEF285I JES2.JOB00006.S00104 SYSOUT
IEF373I STEP /GO / START 21337.0103
IEF374I STEP /GO / STOP 21337.0103 CPU OMIN 00.00SEC SRB OMIN 00.00S
IEF237I 180 ALLOCATED TO SYS00001
IEF285I SYS21337.T010349.RA000.NEWCOB.R0000001 KEPT *-----0
IEF285I VOL SER NOS= WORK02.
IEF285I SYS21337.T010349.RA000.NEWCOB.GODATA DELETED
IEF285I VOL SER NOS= WORK02.
IEF375I JOB /NEWCOB / START 21337.0103
IEF376I JOB /NEWCOB / STOP 21337.0103 CPU OMIN 00.07SEC SRB OMIN 00.02S
CB545 V2 LVL78 01MAY72 IBM OS AMERICAN NATIONAL STANDARD COBOL

1
00001 001 IDENTIFICATION DIVISION.
00002 002 PROGRAM-ID. 'HELLO'.
00003 003 ENVIRONMENT DIVISION.
00004 004 CONFIGURATION SECTION.
00005 005 SOURCE-COMPUTER. IBM-360.
00006 006 OBJECT-COMPUTER. IBM-360.
00007 0065 SPECIAL-NAMES. 7828K FREE
```

Press [F3] several times to return to the main screen.

## FORTRAN

FORTRAN first appeared in 1957. The latest stable release was in 2018. It's widely used in scientific and engineering applications.

Follow the COBOL instructions, with the following differences:

1. Copy 'TESTFORT' instead of 'TESTCOB', and name it 'NEWFORT' instead of 'NEWCOB'.
2. When you edit NEWFORT, change the job name and description to indicate NEWFORT instead of

NEWCOB.

## PL/1

You may also see the name written as 'PL/I'. It first appeared in 1964, and the latest stable release was in 2019.

Follow the COBOL instructions, with the following differences:

1. Copy 'TESTPL1' instead of 'TESTCOB', and name it 'NEWPL1' instead of 'NEWCOB'.
2. When you edit NEWPL1, change the job name and description to indicate NEWPL1 instead of NEWCOB.

There's an additional change, and it's important:



There's a critical "gotcha" in PL/1 programs. MVS requires a JOBLIB statement in the JCL, and it's not included in the turnkey sample programs. The following additional line is required in the Data Set. This should be added as a new line at the end of the JCL section:

```
//JOBLIB DD DSN=SYS1.PL1LIB,DISP=SHR
```



To add a new line in the editor, enter an 'I' in the first column of a row, and the new row will be inserted after. Example location:

```
000013 //HELWR  
000014 //COB.SY  
000015 001 I  
000016 002 P  
000017 003 F
```

You can also delete a line by entering a 'D' in the same location.

## C

C first appeared in 1972, and the latest stable release was in 2018. It's the most actively used "old language" by far, heavily used in operating system and kernel development, device drivers, and embedded development.

It's a dangerous language: Memory management is tricky. But, it's also extremely powerful, as it's well suited for getting close to the hardware.

Follow the COBOL instructions, with the following differences:

1. Copy 'TESTJCC' instead of 'TESTCOB', and name it 'NEWJCC' instead of 'NEWCOB'. **Use TESTJCC as your copy source, not TESTGCC. The GCC compiler ABENDs (throws an error) in the MVS Turnkey system.**
2. When you edit NEWJCC, change the job name and description to indicate NEWJCC instead of NEWCOB.

[mainframe](#), [retro](#)

From:  
<https://kbase.devtoprd.com/> - **Knowledge Base**

Permanent link:  
[https://kbase.devtoprd.com/doku.php?id=jcl\\_programming\\_mvs\\_turnkey\\_system](https://kbase.devtoprd.com/doku.php?id=jcl_programming_mvs_turnkey_system)

Last update: **2024/08/11 18:09**

