How to write simple CICS MAP And associate it with simple COBOL-CICS program

1. Design the screen using BMS
2. Compile the BMS screen code using Job and it creates:
    a. Physical Map
        i. Load Module
        ii. Attribute values
    b. Symbolic Map
        i. Source code module
        ii. Variable declaration
3. Make entry in PPT
    a. CESN to sign on CESF to sign off. RACF setting and can be off or on
    b. CEDA DEF MAP (Define Mapset)
        i. Fill MAPSET NAME AND GROUP fields
        ii. (Group name is defined in resource definition file and CICS admin needs to set this group)
        iii. F3 to come out of screen and Pause-break to come back to CICS region
    c. CEDA INS MAP (Install map/load the map)
        i. Fill MAPSET, GROUP
        ii. Success message : Installed successfully
    d. CECI SEND MAP('MAPNAME') MAPSET('MAPSETNAME')   (See the map to test)
    e. CECI: Command level interpretor.
    f. CECI READQ TD (QID)
    g. CEDA DEL MAP (To remove the entry from PPT table)
        i. Delete Successful
    h. CEDA ALTER MAP (To use the same name in case you are getting name already exists)
    i. CEDA DEF TRANS <ENTER> (PPT entry for transaction)
        i. Fill TRANSACTION, PROGRAM, GROUP
        ii. Success message: 'Define successful'
    j. CEDA INS TRANS <ENTER> (Install trans/load trans)
        i. Fill TRANSACTION, GROUP
        ii. Success message: 'Install successful'
    k. CEDA DEF PROG <ENTER> (PPT entry for program)
        i. Fill PROGRAM, LANGUAGE, GROUP
        ii. Success message: 'Define successful'
    l. CEDA IN PROG <ENTER> (Install Prog/Load prog)
        i. Fill PROGRAM,GROUP
        ii. Success message: 'Install successful'
    m. CEMT I TA (others)
    n. CEDC DIS GR(GRP NAME) (others)
    o. CEDA have define, delete, alter access , CEDB has define and alter access, CEDC has display access

4. Write COBOL-CICS program

5. Compile COBOL-CICS program
6. Make entries in PCT, PPT (using CEDA DEF…)
7. Load tables in memory (using CEDA INS …)
8. Enter CICS run-time environment
9. Execute transaction by specifying trans-id
    a. TRANID <ENTER>

BMS Code

BMS code when compiled leads to symbolic map and physical map generation.

DFHMSD Options (Mapset definition)

```
<- Col 1    <- Col 10   < - Col 16                                                      <-72
MSEMP1 DFHMSD TYPE=&SYSPARM, LANG=COBOL, MODE=INOUT,                      *
                          CTRL=(FREEKB,ALARM), STORAGE=AUTO, TIOAPFX=YES,     *
                          MAPATTS=(COLOR,HILIGHT), DSATTS=(COLOR,HILIGHT)
UTIME  DFHMSD TYPE=FINAL
```

* in 72 column is continuation character and it can be any alphanumeric character

TIOAPFX : Terminal Input output area prefix
If lang=cobol than TIOAPFX=yes is a must
(It will insert 12 byte field after every field, which will be used by CICS internally)

Storage=Auto for multiple map under one mapset condition, optional for 1-mapset, 1-map condition

TYPE=$SYSPARM options are MAP, DSECT, $SYSPARM , FINAL based on option you have selected either Physical Map (MAP) , Symbolic map (DSECT) , both ($SYSPARM), End of BMS Code (FINAL) is generated

MAPATT=(COLOR,HILIGHT) in physical map attributes get stored based on this value. If you specify color, it will just have color attribute you assigned, highlight option of blink will be ignored specified through program.

LANG=COBOL, PL1, ASM

DSATTS=(COLOR,HILIGHT) variables for color and highlight in symbolic map gets generated based on this.

MODE=INOUT,IN,OUT

EXTATT=YES/NO in newer version of CICS. It replaces MAPATTS and DSATTS option.

DFHMDI options (Map definition)

```
MAPEMP  DFHMDI   SIZE=(24,80), LINE=1, COLUMN=1, JUSTIFY=RIGHT
```

SIZE=(24,80) for single map or any map size you want. For multiple maps you need to be extra careful with this option.

JUSTIFY=RIGHT/LEFT

DFHMDF option (Field definition)

Display field

```
PTITLE  DFHMDF POS=(2,30), LENGTH=9,ATTRB=(NORM,PROT),              *
                        INITIAL='DODO LOVE YOU',HILIGHT=BLINK,COLOR=BLUE
```

POS=(2,30) Position of the field on the map. Its map specific for multiple map condition. Its relative position and not the actual.
LENGTH=9 Field length
ATTRB=(NORM,PROT,IC) Attribute of field Options are
- Set 1: NORM, DRK, BRT
- Set 2: PROT, UNPROT, ASKIP
- Set 3: This is option and can have value "IC" and sets the cursor to this by default when screen is thrown out first time. Or "FSET" meaning field value returned irrespective if it is changed or not (Refer DFHMDT functionality for further details). NUM is data entry for numeric field and user will not be able to type non-numeric data. (See edit field), position of sets is not important, it can be in any order.

INITIAL='ACC' : length has to be less or equal. Only one quote mark allowed, double quote not acceptable.
HILIGHT=BLANK/REVRSE
COLOR=BLUE/WHITE/RED etc etc…

Stopper field

```
AENO DFHMDF POS(5,34), LENGTH=1, ATTRB=(NORM,ASKIP)
```

Askip field or stopper field is placed at the end of all *editable fields* to stop the text from edit fields running into other MAP areas.

Editable field

```
UENO DFHMDF POS=(5,30), LENGTH=3, ATTRB=(NORM,UNPROT,IC,NUM,FSET),     *
             PICIN='X(3)', PICOUT='X(3)',HILIGHT=REVERSE
```

PICIN and PICOUT are field definition to pass it onto how it gets displayed on map.

Complete BMS example

```
MSEMP1 DFHMSD TYPE=&SYSPARM, LANG=COBOL, MODE=INOUT,                    *
                  CTRL=(FREEKB,ALARM), STORAGE=AUTO, TIOAPFX=YES,      *
                  MAPATTS=(COLOR,HILIGHT), DSATTS=(COLOR,HILIGHT)
MAPEMP  DFHMDI   SIZE=(24,80), LINE=1, COLUMN=1
PTITLE  DFHMDF POS=(2,30), LENGTH=39,ATTRB=(NORM,PROT),            *
                  INITIAL='DODO LOVEs YOU',HILIGHT=BLINK,COLOR=BLUE
UENO DFHMDF POS=(5,30), LENGTH=3, ATTRB=(NORM,UNPROT,IC,NUM,FSET),     *
            PICIN='X(3)', PICOUT='X(3)',HILIGHT=REVERSE
AENO DFHMDF POS(5,34), LENGTH=1, ATTRB=(NORM,ASKIP)
UTIME  DFHMSD TYPE=FINAL
       END
```