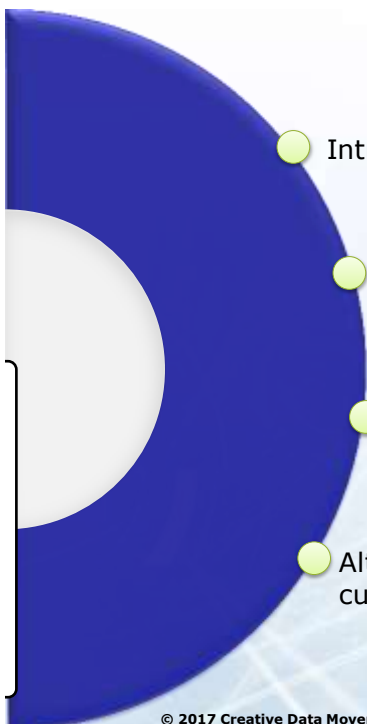


Finding Your Way with CICS Maps

Presented by:

- ✓ *Mary Abdill*
- ✓ *Creative Data Movers*
- ✓ *mary.abdill@CreativeDataMovers.com*
- ✓ *www.CreativeDataMovers.Inc*

© 2017 Creative Data Movers, Inc.

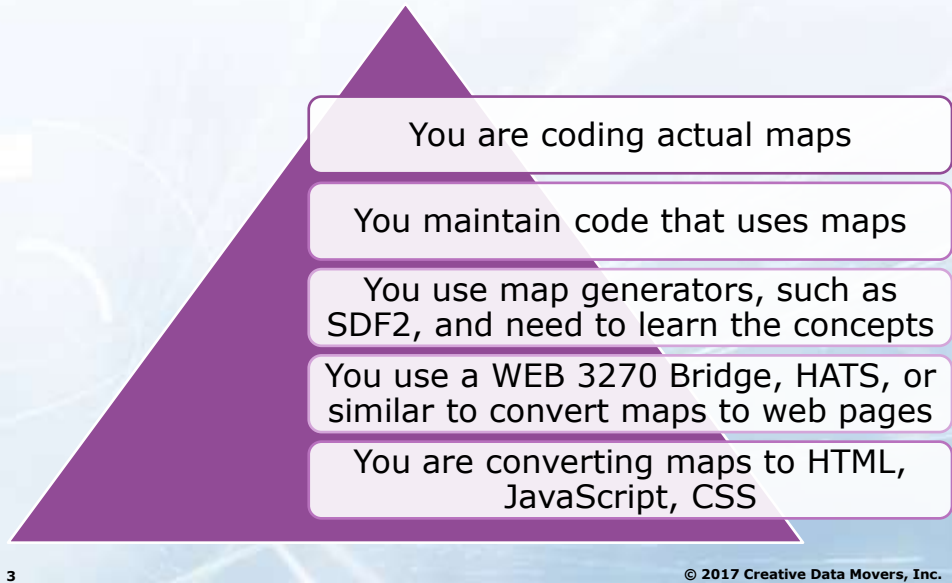


- Introduction
- BMS and the UI
- COBOL: SEND/RECEIVE MAP
- Altering attribute bytes, setting cursor, etc. in COBOL

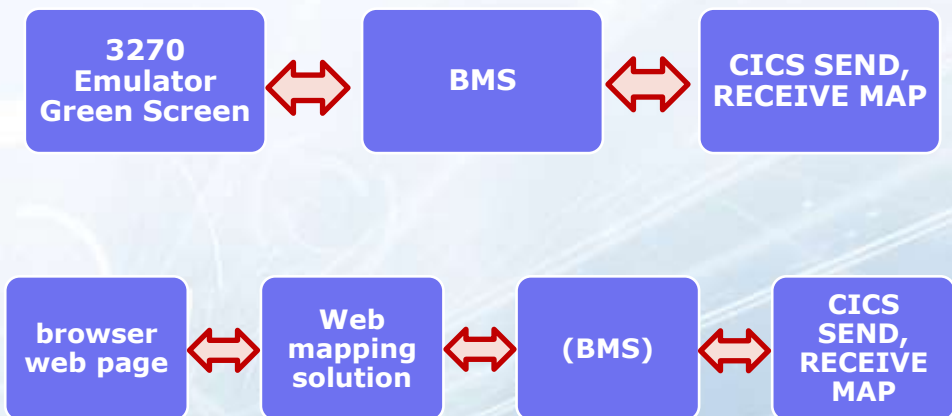
Module Topics

© 2017 Creative Data Movers, Inc.

Possible Reasons to Learn About Maps



Possible Data Flow Scenerios



BMS provides the data mapping between the screen and the application

What are maps

Mapset

- Contains one or more maps

Compiled & decompiled code

- Handles I/O between the CICS application and the user's terminal

Physical Map

- Compiled assembler macro code

Symbolic Map

- copybook of the map inputs/outputs

5

© 2017 Creative Data Movers, Inc.

BMS: Formatted Terminal Displays

- ✓ *Basic Mapping Support*
- ✓ *BMS allows you to format the user's screen*
- ✓ *Allows the application to simply deal with the user data that is entered or changed*

© 2017 Creative Data Movers, Inc.

BMS Concepts

Describes
the UI

- "This method of map definition is still widely used"
- IBM manual—CICS TS v5.3.0

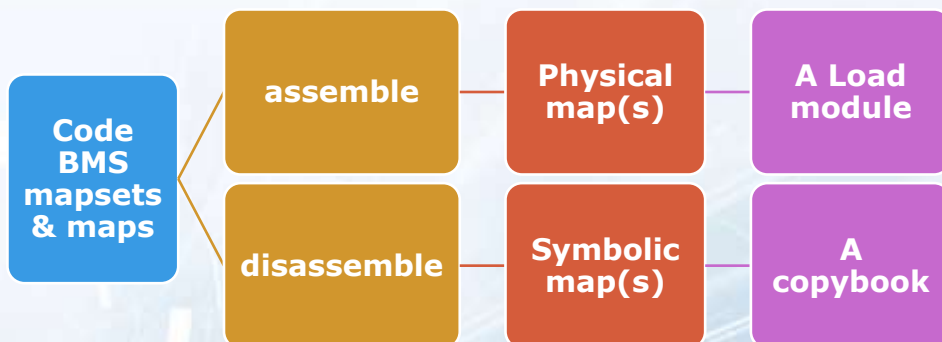
Assembler

- Three assembler language macros for defining maps
- Tools available for drawing

7

© 2017 Creative Data Movers, Inc.

Map Preparation



Then, NEWCOPY the mapset load module:

CEMT SET PROG(*mapset*) NEW

Then check if map looks good:

CECI SEND MAP(*map*) MAPSET(*mapset*) ERASE FREEKB

8

© 2017 Creative Data Movers, Inc.

Fields

- 3270 screens are character-based and mark fields with attribute bytes



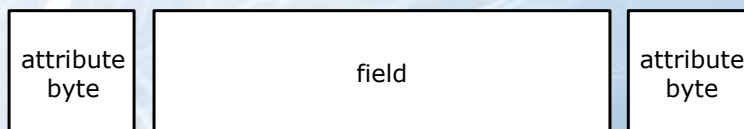
- Field starts immediately after attribute byte
 - Generally goes up to the next attribute byte
 - Output and text fields do not need ending attribute
- Field has a length
 - But length does not stop user from typing
 - Length controls amount of data transferred

9

© 2017 Creative Data Movers, Inc.

Attribute bytes

- Mark beginning and usually end of field
- Are protected
 - User cannot type over an attribute byte
- Provide features for the upcoming field
 - Is input or output (unprotected or protected)
 - Bold
 - Is autoskip on?
 - Is MDT on? etc.



10

© 2017 Creative Data Movers, Inc.

Stopper byte

Without a stopper byte



- User types into the field, but is not stopped if typing goes beyond the end

With a stopper byte

- User cannot type past stopper byte
- Stopper byte is a protected attribute byte

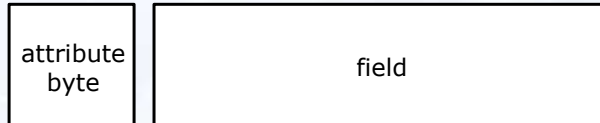


11

© 2017 Creative Data Movers, Inc.

Autoskip (skipper) byte

Without an autoskip byte



- User not stopped if typing goes too far

With an autoskip byte

- When user types last byte in field, cursor automatically skips to next input field
- Autoskip byte is a special attribute bit



12

© 2017 Creative Data Movers, Inc.

Stopper and Skipper Bytes



A byte that causes the cursor to lock or skip when the end of a field is reached

Immediately follows an unprotected field

Can be defined as ASKIP to cause the cursor to jump to the next unprotected field

Can be defined as PROT which will cause the keyboard to lock if it encounters this area

Take up one byte position on the screen even if they have zero length

13

© 2017 Creative Data Movers, Inc.

Choice – stopper or autoskip byte

- Autoskip is appropriate if input value always the same length
 - Area code for US phone number
 - 5-digit zip code
 - A one character code such as N/S/E/W
- Stopper best if input is different lengths
 - Name, Address,
 - Comments, etc.
 - User uses tab to go to next field
 - User gets stopped if they try to type more
 - Reset – Ctrl+r, or other

14

© 2017 Creative Data Movers, Inc.

MDT = Modified Data Tag

● Purpose

- If MDT is 0, data is not transferred; and if MDT is 1, then data is transferred.
- Turn MDT on to transmit data from UI
- Turned on when user enters data, deletes data, or uses "end" or "clear to erase data"

● FRSET

- Turns off all MDTs in Map; recommended

● FSET

- Turns on MDT for a specific field
- Allows that field to be returned, even if user makes no changes to it

15

© 2017 Creative Data Movers, Inc.

3 Assembler macros—define BMS maps

DFHMDF

- Defines an individual field on a screen or web page

DFHMDI

- Defines a map as a collection of fields
- Fields can be:
 - input (unprotected)
 - output (protected)
 - text (not seen by application program)

DFHMDS

- Groups maps into a single mapset

16

© 2017 Creative Data Movers, Inc.

Creating BMS Maps

Code BMS macros:

- Assembler macros
- Assemble and link
- Disassemble

Tools to draw screens and create maps include:

- SDF
- SDF II screen
- RDz
- Micro Focus CICS/MTS
- Various CASE tools
- Etc.



18

© 2017 Creative Data Movers, Inc.

Screen Illustration

```

                                COURSE BOOKING SYSTEM

KEY IN EITHER
  COURSE CODE:  <CODEMMDDYYYY> 
OR...
  COURSE TITLE:  COURSE DATE: 

  CLIENT NAME: 

  ACTION: 

  ERROR:

  PF3: RETURN TO MENU      CLEAR KEY: EXIT      ENTER: PROCEED
  
```

24

© 2017 Creative Data Movers, Inc.



Let's look at some actual
BMS code

27

© 2017 Creative Data Movers, Inc.

BMS Macro Format Source Columns

1	9 or 10	16		72
↓	↓	↓		↓
name	opcode	parameter,parameter	comments	

*			comments	
---	--	--	----------	--

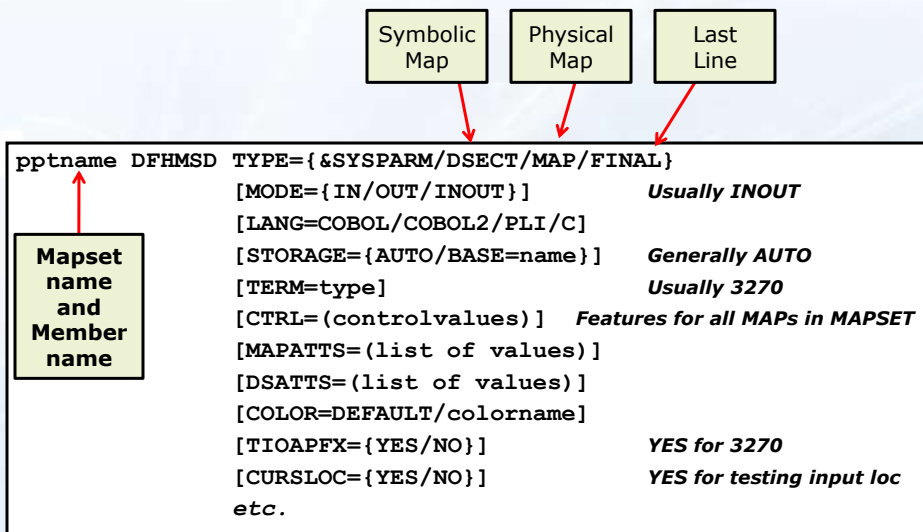
	opcode	parameter,parameter, parameter	comments	X
--	--------	-----------------------------------	----------	---



28

© 2017 Creative Data Movers, Inc.

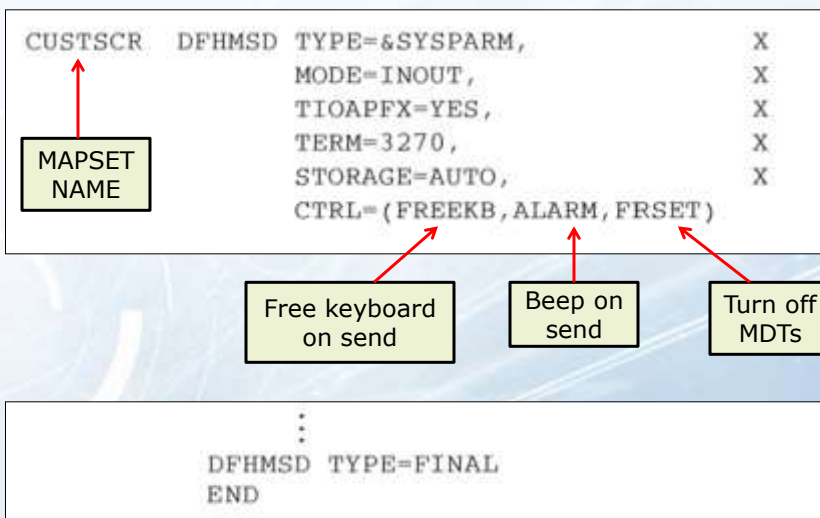
Mapset Definition: DFHMSD Syntax



29

© 2017 Creative Data Movers, Inc.

DFHMSD Examples



30

© 2017 Creative Data Movers, Inc.

Map Definition: DFHMDI Syntax

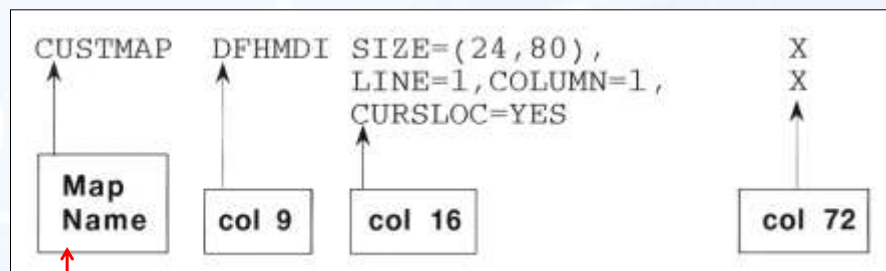
```
mapname DFHMDI SIZE=(lines,columns)
          [LINE={NEXT/line#/SAME}]
          [COLUMN={SAME/NEXT/column#}]
          [JUSTIFY=({LEFT/RIGHT},
                   {FIRST/LAST})]
          [CURSLOC={YES/NO}]
          [CTRL=(controlvalues)]
          [MAPATTS=(list of values)]
          [DSATTS=(list of values)]
          [HILIGHT={OFF/BLINK/REVERSE/UNDERLINE}]
```



31

© 2017 Creative Data Movers, Inc.

DFHMDI Example



Up to 7
characters



32

© 2017 Creative Data Movers, Inc.

Screen Field Definition: DFHMDF Macro

```
fldname DFHMDF POS=(line,column),
          LENGTH=length
          [ATTRB=({ASKIP/PROT/UNPROT[,NUM],
                  {NORM/BRT/DRK},IC,FSET)]
          [{INITIAL='LITERAL'/XINIT='hexlit'}]
          [PICIN='COBOL picture']
          [PICOUT='COBOL picture']
          [JUSTIFY=({LEFT/RIGHT,BLANK/ZERO})]
          [GRPNAME=prev defined fldname]
          [OCCURS=number]
          [HILIGHT={OFF/BLINK/REVERSE/UNDERLINE}]
          [COLOR={DEFAULT/BLUE/RED/PINK/GREEN/
                  YELLOW/TURQUOISE/NEUTRAL}]
          [ASIS]
```

Up to 30 characters

ASIS depends on a CICS system setting for whether it actually works

34

© 2017 Creative Data Movers, Inc.

DFHMDF ATTRB Options



Protection Options

- ASKIP causes cursor to jump to the next unprotected field
- PROT means *no* input allowed here; the keyboard *will* lock
- UNPROT indicates an input field



Shift Option

- NUM *forces* the keyboard into numeric lock and right aligns input
- Does *not* work for some 3270 emulation software, so do *not* count on it

cont.

36

© 2017 Creative Data Movers, Inc.

DFHMDf ATTRB Options (cont.)



Intensity Options

- NORM makes the brightness normal intensity
- BRT makes the field bright
- DRK makes the field hidden (*not displayed*); for example, Password input



Other Options

- FSET turns on the Modified Data Tag (MDT) bit
- IC causes the cursor to be inserted in this field when map is sent

37

© 2017 Creative Data Movers, Inc.

BMS Example

```

PRINT NOGEN
CBOOKMS DFHMSD TYPE=&SYSPARM, X
                MODE=INOUT, X
                LANG=COBOL, X
                STORAGE=AUTO, X
                TERM=3270, X
                CTRL=FREEKB, X
                TIOAPFX=YES
BOOKMAP DFHMDI SIZE=(024,080), X
                MAPATTS=(COLOR,PS,HILIGHT,VALIDN), X
                DSATTS=(COLOR,PS,HILIGHT,VALIDN)
DFHMDf POS=(02,026), X
                ATTRB=(ASKIP,NORM), X
                COLOR=NEUTRAL, X
                LENGTH=021, X
                INITIAL='COURSE BOOKING SYSTEM'

```

cont.

38

© 2017 Creative Data Movers, Inc.

BMS Example (cont.)

```

DFHMDF POS=(05,002),           X
      ATTRB=(ASKIP,NORM),      X
      COLOR=RED,               X
      LENGTH=013,              X
      INITIAL='KEY IN EITHER'
DFHMDF POS=(07,003),           X
      ATTRB=(ASKIP,NORM),      X
      COLOR=NEUTRAL,           X
      LENGTH=011,              X
      INITIAL='COURSE KEY:'
CRSEKEY DFHMDF POS=(07,015),   X
      LENGTH=012,              X
      ATTRB=(UNPROT,IC,FSET,NORM), X
      COLOR=NEUTRAL
    
```

cont.

39

© 2017 Creative Data Movers, Inc.

How to count positions

```

DFHMDF POS=(07,003),           X
      ATTRB=(ASKIP,NORM),      X
      COLOR=NEUTRAL,           X
      LENGTH=011,              X
      INITIAL='COURSE KEY:'
CRSEKEY DFHMDF POS=(07,015),   X
    
```

length = 11

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Attribute byte

POS=(07,003)

Attribute byte

POS=(07,015)

40

© 2017 Creative Data Movers, Inc.

BMS Example (cont.)

```

TITLE   DFHMDF POS=(11,017),           X
        LENGTH=025,                   X
        ATTRB=(UNPROT,FSET,NORM),     X
        COLOR=NEUTRAL
        DFHMDF POS=(11,043),           X
        ATTRB=(ASKIP,NORM),LENGTH=000
        DFHMDF POS=(11,044),           X
        ATTRB=(ASKIP,NORM),           X
        COLOR=NEUTRAL,                X
        LENGTH=012,                    X
        INITIAL=' COURSE DATE: '
CRSEDAT DFHMDF POS=(11,057),           X
        LENGTH=010,                    X
        ATTRB=(UNPROT,FSET,NORM),     X
        COLOR=NEUTRAL
        DFHMDF POS=(11,068),           X
        ATTRB=(ASKIP,NORM),LENGTH=000

```

cont.

42

© 2017 Creative Data Movers, Inc.

BMS Example (cont.)

```

        DFHMDF POS=(24,029),           X
        ATTRB=(ASKIP,NORM),           X
        COLOR=NEUTRAL,                X
        LENGTH=015,                    X
        INITIAL='CLEAR KEY: EXIT'
        DFHMDF POS=(24,045),           X
        ATTRB=(ASKIP,NORM),           X
        COLOR=NEUTRAL,                X
        LENGTH=014,                    X
        INITIAL='ENTER: PROCEED'
        DFHMDF TYPE=FINAL
        END

```

45

© 2017 Creative Data Movers, Inc.

Symbolic Map Example

```

01 BOOKMAPI.
03 FILLER                PIC X(12).
03 CRSEKEYL             PIC S9(4) COMP.
03 CRSEKEYF            PIC X.
03 CRSEKEYA REDEFINES CRSEKEYF          PIC X.
03 FILLER                PIC X(4).
03 CRSEKEYI            PIC X(12).
03 TITLEL              PIC S9(4) COMP.
03 TITLEF              PIC X.
03 TITLEA REDEFINES TITLEF              PIC X.
03 FILLER                PIC X(4).
03 TITLEI              PIC X(25).
03 CRSEDATL            PIC S9(4) COMP.
03 CRSEDATF           PIC X.
03 CRSEDATA REDEFINES CRSEDATF          PIC X.
03 FILLER                PIC X(4).
03 CRSEDATI            PIC X(12).

```

cont.

47

© 2017 Creative Data Movers, Inc.

Symbolic Map Example (cont.)

```

03 CLNAMEL             PIC S9(4) COMP.
03 CLNAMEF            PIC X.
03 CLNAMEA REDEFINES CLNAMEF          PIC X.
03 FILLER              PIC X(4).
03 CLNAMEI            PIC X(30).
03 ACTMSGL            PIC S9(4) COMP.
03 ACTMSGF           PIC X.
03 ACTMSGA REDEFINES ACTMSGF          PIC X.
03 FILLER              PIC X(4).
03 ACTMSGI            PIC X(67).
03 ERRMSGL            PIC S9(4) COMP.
03 ERRMSGF           PIC X.
03 ERRMSGA REDEFINES ERRMSGF          PIC X.
03 FILLER              PIC X(4).
03 ERRMSGI            PIC X(67).

```

cont.

48

© 2017 Creative Data Movers, Inc.

Symbolic Map Example (cont.)

```

01 BOOKMAPO REDEFINES BOOKMAPI.
  03 FILLER                PIC X(12).
  03 FILLER                PIC X(3).
  03 CRSEKEYC              PIC X.
  03 CRSEKEYP              PIC X.
  03 CRSEKEYH              PIC X.
  03 CRSEKEYV              PIC X.
  03 CRSEKEYO              PIC X(12).
  03 FILLER                PIC X(3).
  03 TITLEC                PIC X.
  03 TITLEP                PIC X.
  03 TITLEH                PIC X.
  03 TITLEV                PIC X.
  03 TITL EO               PIC X(25).

```

cont.

49

© 2017 Creative Data Movers, Inc.

Symbolic Map Example (cont.)

```

  03 FILLER                PIC X(3).
  03 CRSEDATEC             PIC X.
  03 CRSEDATEP             PIC X.
  03 CRSEDATEH             PIC X.
  03 CRSEDATEV             PIC X.
  03 CRSEDATEO             PIC X(12).
  03 FILLER                PIC X(3).
  03 CLNAMEC                PIC X.
  03 CLNAMEP                PIC X.
  03 CLNAMEH                PIC X.
  03 CLNAMEV                PIC X.
  03 CLNAMEO                PIC X(30).

```

cont.

50

© 2017 Creative Data Movers, Inc.

Symbolic Map Example (cont.)

```

03 FILLER          PIC X(3).
03 ACTMSGC        PIC X.
03 ACTMSGP        PIC X.
03 ACTMSGH        PIC X.
03 ACTMSGV        PIC X.
03 ACTMSGO        PIC X(67).
03 FILLER          PIC X(3).
03 ERRMSGC        PIC X.
03 ERRMSGP        PIC X.
03 ERRMSGH        PIC X.
03 ERRMSGV        PIC X.
03 ERRMSGO        PIC X(67).

```

51

© 2017 Creative Data Movers, Inc.

Sample BMS JCL

```

//MYIDJ JOB ('ACCTG'), 'your-name', CLASS=A, MSGCLASS=A,
//          NOTIFY=&SYSUID
//*****
//* BMS ASSEMBLE
//STEP1 EXEC PROC=DFPHASVMS, PARM.ASSEM='SYSPARM(MAP)'
//SYSPUNCH DD DSN=&&TEMP, DCB=(RECFM=FB, BLKSIZE=2960),
//          SPACE=(2960, (10, 10)), UNIT=SYSDA, DISP=(NEW, PASS)
//SYSIN DD DSN=MYID.CLASS.BMS(MYBMS), DISP=SHR <== change
//*****
//* LINK
//STEP2 EXEC PROC=DFHLNKVS, PARM='LIST, LET, XREF'
//SYSMOD DD DSN=?????.YOUR.loadlib, DISP=SHR <== change
//SYSLIN DD DSN=&&TEMP, DISP=(OLD, DELETE)
//          DD *
//          MODE RMODE(ANY|24) <== choose
//          NAME PHONM90(R) <== change
//*****
//* DISASSEMBLE
//STEP3 EXEC PROC=DFPHASVMS, PARM.ASSEM='SYSPARM(DSECT)'
//SYSPUNCH DD SYSOUT=*
//COPYLIB DD DSN=?????.yourlib.copypub, DISP=SHR <== change
//SYSIN DD DSN=MYID.CLASS.BMS(MYBMS), DISP=SHR <== change

```



52

© 2017 Creative Data Movers, Inc.

CICS Programming with BMS Objects

✓ Embed CICS commands in your COBOL source to SEND and RECEIVE MAPs

© 2015 Creative Data Movers, Inc.

Pseudo-Conversational Programming

cutd

SEND MAP
MOVE '1' TO COMMAREA
RETURN with COMMAREA
program ends

Book code: ci02999999

starts again
RECEIVE MAP ci02999999
READ ci02999999 RECORD
prepare map

course update
cutd ci02
Date: 10/01/12
Book info: CICS pgm

SEND MAP with book information
MOVE '2' & book code to COMMAREA
RETURN with COMMAREA
program ends

User changes book info

starts again
RECEIVE MAP with updates
READ and REWRITE updates
prepare map

course update
cutd ci02
Date: 11/11/16
Book info: CICS
RECORD UPDATED
** CLEAR to END **

SEND map
MOVE '3' to COMMAREA
RETURN with COMMAREA
program ends

User presses CLEAR key

starts again
RETURN without transid or commarea
program and transaction both end

© 2016 Creative Data Movers, Inc.

Using a BMS mapset in a COBOL Program

SEND MAP places the physical and/or Symbolic map data on the screen

Programs can alter attribute values prior to a SEND MAP

Programs can override some control options like FREEKB, ALARM, FRSET

RECEIVE MAP retrieves data from the screen fields that have MDT on and places it in the corresponding variables in the COBOL program symbolic map

BMS will **NEVER** transmit LOW-VALUES even if a MDT tag is on.

Check length if MDT is on.



© 2016 Creative Data Movers, Inc.

SEND MAP Format

```
EXEC CICS SEND
      MAP ('dfhmdi name')
      [MAPSET('dfhmsd name')]
      [FROM(data name)]
      [{MAPONLY/DATAONLY}]
      [{ERASE/ERASEAUP}]
      [CURSOR [(number)]]
      [FREEKB]
      [ALARM]
      [FRSET]
      [RESP (dataname)]
END-EXEC
```

© 2016 Creative Data Movers, Inc.

SEND MAP Example

```
EXEC CICS SEND MAP ('CUSTMAP')
          MAPSET ('CUSTSCR')
          FRSET
          ERASE
          RESP (WS-RESP-CD)
END-EXEC
```



© 2011 Creative Data Movers, Inc.

RECEIVE MAP Format

```
EXEC CICS RECEIVE
          MAP ('dfhmdi name')
          [MAPSET('dfhmsd name')]
          [INTO(dataname)/SET(ptr-ref)]
          [ASIS]
          [RESP (dataname)]
END-EXEC
```

(ADDRESS of data)

- After RECEIVE MAP, WS-RESP-CD contains:
 - NORMAL or EOC - Successful
 - MAPFAIL - All input data has a length of zero
 - Other - Failure
- EIBLK values are filled in

© 2016 Creative Data Movers, Inc.

RECEIVE MAP – Two Ways

```
EXEC CICS RECEIVE  MAP ('CRSEMAP')  
                  MAPSET ('CUSTSCR')  
                  RESP (WS-RESP-CD)  
END-EXEC
```

```
EXEC CICS RECEIVE  MAP ('CRSEMAP')  
                  MAPSET ('CUSTSCR')  
                  SET (ADDRESS OF CRSEMAPI)  
                  RESP (WS-RESP-CD)  
END-EXEC
```

© 2011 Creative Data Movers, Inc.

After RECEIVE MAP

L - Length

- Contains actual length or zero



F/A - Field/Attribute

- If not low-values, then user erased input with ERASE EOF and turned on MDT



I - Input

- The actual text or data



© 2011 Creative Data Movers, Inc.

Altering Attributes

- ✓ Set attribute bytes in your program using the fields in the symbolic map that end with 'A'
- ✓ Values in the symbolic map sent to BMS will override defaults originally defined for the field
- ✓ Attribute byte x'00' is never sent
- ✓ Attributes are bit flags
- ✓ CICS comes with a copybook called DFHBMSCA containing bit pattern values for various attributes
- ✓ COPY DFHBMSCA – or code your own attributes

© 2016 Creative Data Movers, Inc.

Attribute Byte Bit Flag Configuration

Bits are numbered 01 23 45 67

Attribute byte bits 0 and 1 depend on the values in bits 2 to 7

Bits 2 and 3 are for Protection and Shift (numeric):

- 00 is Unprotected Alphanumeric
- 01 is Unprotected Numeric
- 10 is Protected
- 11 is Autoskip



© 2016 Creative Data Movers, Inc.

Attribute Byte Bit Flag Configuration (cont.)

Bits 4 and 5 determine intensity:

- 00 is Normal
- 01 is also Normal
- 10 is Bright
- 11 is Dark



Bit 6 is always 0 (but not always)

Bit 7 is the Modify Data Tag (MDT):

- 0 means the field has not been changed so CICS will *NOT* return data for that field
- 1 means a change was made, so CICS will return that data

© 2016 Creative Data Movers, Inc.

Manually setting Attributes

For unprot, alphanumeric, norm, MDT on

- bits 2, 3 = 00 (unprot, alpha)
- bits 4, 5 = 00 or 01 (normal intensity)
- bit 6 = 0 (usually off)
- bit 7 = 1 (MDT on)
- 0000 0101 = x'05'
- MOVE x'05 TO fieldA

If all bits become zero, make bit 6 = 1

- 0000 0100 = x'04
- MOVE X'04 to fieldA

Program Example: Modifying Attributes

```

IF CUSTNUM1 NOT NUMERIC
  MOVE DFHMBRY TO CUSTNUMA
  MOVE 'CUSTOMER NUMBER MUST BE NUMERIC '
    TO ERRMSGO
  GO TO SEND-MAP-RTN.
.
.
SEND-MAP-RTN.
  EXEC CICS SEND      MAP ('CRSEMAP')
                    MAPSET('CUSTSCR')
                    ERASE
                    RESP(WS-RESP-CD)

END-EXEC

```

© 2012 Creative Data Movers, Inc.

Cursor Positioning Techniques



IC option on the ATTRB parameter can supply the cursor location



Programs can override IC by specifying CURSOR on SEND



There are two forms of the CURSOR option



Symbolic cursor positioning lets us flag the desired field by moving -1 to the length field and specifying CURSOR



Direct cursor positioning is *inadvisable*. CURSOR is followed by a number in parentheses that represents displacement to cursor position.

© 2015 Creative Data Movers, Inc.

Symbolic Cursor Positioning Example

```

MOVE DFHBMFSE TO CUSTNUMA
IF CUSTNUM1 IS NOT NUMERIC
  MOVE -1      TO CUSTNUML
  MOVE DFHBMBRY TO CUSTNUMA
  MOVE 'CUST NUM MUST BE NUMERIC'
                TO ERRMSGO
  PERFORM SEND-MAP-RTN
.
SEND-MAP-RTN.
  EXEC CICS SEND MAP ('CUSTMAP')
                MAPSET ('CUSTSCR')
                ERASE
                CURSOR
                RESP (WS-RESP-CD)

END-EXEC
IF. . .

```

© 2011 Creative Data Movers, Inc.

Before SEND MAP

Field A

- Move new value to attribute bytes



Field O

- Move data to output fields



Field L

- Move -1 to set cursor to this field



- If error, MOVE message to ERRMSGO

© 2015 Creative Data Movers, Inc.

DFHMDF GRPNAME – A Complex Field

Group definition in DFHMDF

```
MM      DFHMDF POS=(10,1),LENGTH=2,ATTRB=BRT,GRPNAME=DATE
HYPHEN1 DFHMDF POS=(10,3),LENGTH=1,GRPNAME=DATE,INITIAL='- '
DD      DFHMDF POS=(10,4),LENGTH=2,GRPNAME=DATE
HYPHEN2 DFHMDF POS=(10,6),LENGTH=1,GRPNAME=DATE,INITIAL='- '
YY      DFHMDF POS=(10,7),LENGTH=2,GRPNAME=DATE
```

Generated copybook

```
02 DATE.
   03 FILLER  PIC X(2) .
   03 MMA     PIC X.
   03 MMO     PIC X(2) .
   03 HYPHEN1 PIC X(1) .
   03 DDO     PIC X(2) .
   03 HYPHEN2 PIC X(1) .
77  03 YRO     PIC X(2) .
```

© 2017 Creative Data Movers, Inc.

DFHMDF OCCURS – a Repeated Field

DFHMDF

```
MAPFLD DFHMDF POS=(7,1),LENGTH=9,ATTRB=NORM,OCCURS=40
```

COPYBOOK

```
02 MAPFLDG OCCURS 40.
   03 FILLER  PIC X(2) .
   03 MAPFLDA PIC X.
   03 MAPFLDO PIC X(9) .
```

COBOL PROGRAM CODE

```
PERFORM VARYING I FROM 1 BY 1 UNTIL I > 40
  MOVE INPUT-FLD(I) TO MAPFLDO (I)
  IF DAYS-USED(I) < 30 MOVE DFHMBRY to MAPFLDA(I)
END-PERFORM
```

78.

© 2017 Creative Data Movers, Inc.

References

IBM manual: access to CICS via Browser

- <https://www-03.ibm.com/systems/z/os/zvse/solutions/cicsaccess.html>

BMS CICS TS v5.2

- https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.2.0/com.ibm.cics.ts.applicationprogramming.doc/topics/dfhp370.html

BMS CICS TS v5.3

- https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.3.0/com.ibm.cics.ts.applicationprogramming.doc/topics/dfhp373.html

86

© 2017 Creative Data Movers, Inc.

Questions and Answers

Thank You
Mary Abdill



Creative Data Movers, Inc.
www.CreativeDataMovers.com
mary.abdill@CreativeDataMovers.com

87

© 2017 Creative Data Movers, Inc.