



CSI

INTERNATIONAL

**©2008 by Connectivity Systems, Inc.
All Rights Reserved**

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

This material contains confidential and proprietary material of Connectivity Systems, Inc., and may not be used in any way without written authorization from Connectivity Systems, Inc. This material may not be reproduced, in whole or in part, in any way, without prior written permission from Connectivity Systems, Inc.

Permission is hereby granted to copy and distribute this document as follows:

- Each copy must be a complete and accurate copy.
- All copyright notices must be retained.
- No modifications may be made.
- Use of each copy is restricted to the evaluation and/or promotion of Connectivity Systems, Inc.'s *FileMarvel* product or in accordance with a license agreement.

**Batch Execution Guide and Reference Manual Version 3 Release 3
1st Edition June 2008**

Published by Connectivity Systems, Inc.
8120 State Route 138, Williamsport OH 43164
Phone: 800-795-4914 Fax: 740-986-6022

E-Mail: sales@csi-international.com, info@csi-international.com, support@csi-international.com

Internet: <http://www.csi-international.com/>

Table of Contents

COMPILE - Compile Copybook Function.....	1
COMPILE Statement.....	1
COPY Statement.....	2
COPYLIB Statement.....	3
DISPLAY Statement.....	3
FORMAT Statement.....	4
SYSIN Statement.....	5
SIGNON Statement.....	5
COMPILE Control Statement Examples.....	5
COMPARDS - Compare Datasets Function.....	7
COMPARDS Statement.....	8
SELECT Statement.....	13
SIGNON Statement.....	14
COMPARDS Control Statement Examples.....	14
COPYDS - Copy Dataset Function.....	16
COPYDS Statement.....	17
SKIPREC Statement.....	21
SELECT Statement.....	22
SIGNON Statement.....	23
COPYDS Control Statement Examples.....	23
LABEL - Tape and Cartridge Label Display Function.....	25
LABEL Statement.....	25
LABEL Control Statement Examples.....	26
LISTDIR – List Procedure, Copybook, Manual Function.....	27
LISTVOL – List Volume Dataset Function.....	28
LMODMAP – List Load Module Statistics Function.....	30
MANUAL - Print, delete or import manual function.....	32
PRINTDS - Print Dataset Function.....	35
PRINTDS Statement.....	35
SELECT Statement.....	40
SIGNON Statement.....	40
PRINTDS Control Statement Examples.....	41
SORTDS – Sort Dataset Function.....	43
SPACE – Print Volume Free Space Function.....	45
Batch Execution Language.....	48
Execution JCL.....	48
Execution JCL parms.....	50
Command entry rules.....	52
Symbolic value rules.....	55
String value rules.....	59
OFFLINE – Batch Prompt Panel Function.....	61
Offline Options Panel.....	61
SET - Set Jobcard Information.....	62
SOURCE – Source Storage Directory.....	63
MANUAL - Print Manuals Selection Panel.....	64
PRINTDS - Print File Entry Panel.....	65
CmplBOOK - Compile Copybooks Offline.....	68
SUBMIT - Submit Batch Jobs to the Offline Processor.....	69
LABEL - Display Cartridge or Tape Labels.....	70
COMPARDO - Compare Dataset Offline Prompt Panel.....	71
COPYDO - Copy Datasets Offline.....	73
LISTVTOO – Print Volume Dataset List Offline.....	77
SPACEO – Print Volume Free Space Offline.....	79
LMO DMAPO – Print Load Module Statistics & Code Reversion Offline.....	80

COMPILE - Compile Copybook Function

Compile control statements are used to define copybook templates. See Section 11 COPYBOOKS - Using Copybooks. Copybook templates can be generated automatically through Edit or Browse Entry screens (see Section 4.0 Edit Entry Panel), entered directly through the copybook directory (see Section 11.4.1 Creating and Editing Copybook Control), or entered directly into a PDS member using any editor. They can be compiled online automatically using the Edit or Browse entry panel, explicitly compiled using CMP command on the copybook directory, or compiled in batch (see Section 11.7 BATCH - Compile a Copybook in Batch).

Example:

```
COMPILE FUNC(COPYBOOK)      -
      LANGUAGE({COBOL|PLI})  -
      MEMBER(member)         -
      USERID(userid)        -
      TITLE(title)
FORMAT DISPLAY(STD)
TESTDSN PAYROLL.TEST.MARVEL
COPYLIB CICS.TEST.COBOLE
COPYLIB PAYROLL.TAX1994.COBOLE
COPY ADDTAX
SYSIN '05 TAX-EXEMPT X(9)'
DISPLAY TAX-EXEMPT WHEN EMPLOYEE-TYPE EQ 1
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
COPYLIB CICS.TEST.COBOLE
COPY ADDTAX
FORMAT DISPLAY(COMPACT) SKIP(ON) PAD(' ')
DISPLAY TAX-EXEMPT WHEN EMPLOYEE-TYPE EQ 1 - OR EMPLOYEE-AGE GT 120
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
DISPLAY ACCT-BUSINESS WHEN -
      ACCOUNT-TYPE EQ X'02' AND -
      ACCOUNT-BRANCH EQ X'03' -
      AND ACCT-IMPOUND NE 'YES'
```

COMPILE Statement

Use the batch COMPILE statement to compile record layouts in batch mode. Multiple compilations can be performed in a single job step. Each COMPILE statement is delimited by a MEND statement. A dash continues a statement to a succeeding line. The copybook COMPILE statements are used in on-line or batch compilations to identify the different layouts used with a file, the language of the layout definitions, and various other options.

The statements are specified in free format and can be continued across lines. Use a hyphen at the end of a line that has a continuation. At the end of the section on each statement type are a few examples. Following all the statement types is a section giving complete examples using several statement types.

```
COMPILE      FUNC(COPYBOOK)      -
              LANGUAGE({COBOL|PLI})  -
              MEMBER(member)         -
              USERID(userid)        -
              TITLE(title)
```

FUNC(COPYBOOK) Perform copybook compile.

LANGUAGE The language of the source members indicated by the COPY statements. Valid values are COBOL and PL/I.

COBOL Compile language is COBOL.
PLI Compile language is PLI.

MEMBER The member name of the compiled layout, preferably the same as the COPY statement member name.

USERID The TSO user ID assigned to the member. Use the reserved keyword GLOBAL to define it as a global member available to all users.

TITLE Attach this description to the copybook.

Examples

```
COMPILE FUNC(COPYBOOK)
    MEMBER(ACCTMAIN) <196>
    USERID(JOE012)
    TITLE(Major customer accounts)
COPYLIB PAYROLL.TAX1994.COBOL
COPY ADDTAX
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
MEND
```

```
COMPILE FUNC(COPYBOOK)
    MEMBER(PARTS)
    USERID(GLOBAL)
    TITLE(Manufacturing Parts)
COPYLIB PARTS.CICS.COBOL
COPY LOCPART
DISPLAY INDEPENDENT-PART WHEN USED-IN EQ 0
DISPLAY MAIN-PART WHEN USED-BY-PART EQ 0
DISPLAY SUB-PART WHEN USED-IN NE 0
MEND
```

COPY Statement

The COPY statement specifies a source PDS member that contains the COBOL or PL/1 copycode for the record or file layout. Multiple COPY statements are allowed, for multiple compilations.

Format:

COPY The name of the member in the COPYLIB library. Any number of COPY statements can be included.

Examples

```
COPY ADDTAX
COPY ABCDEF
```

COPYLIB Statement

The COPYLIB statement specifies a source PDS library to look for the COPY members. See Section A.1 COPY Statement. If multiple COPYLIB libraries are specified, they are searched in the given order.

Format:

COPYLIB *library* The library name, fully qualified, without quotes.

Example:

```
COPYLIB CICS.TEST.COBOL
COPYLIB PAYROLL.TAX1994.COBOL
```

DISPLAY Statement

See Help member CBOKCCTL

The DISPLAY statement is defines a complex multi-record type copybook template. Use the DISPLAY statement to identify which file layout to use for each record type.

Format:

DISPLAY *recordtype* **WHEN** *fieldname* **cond** *value*
 [**{AND|OR}** *fieldname* **cond** *value*]

where:

recordtyp The 01 COBOL level or a PLI structure included in the copybook template with a COPY statement.

fieldname The field containing a value used to determine which record layout to use.

cond The condition to compare the data in the dataset represented by *fieldname* to a value to determine which recordtype to use for this record.

EQ	Equals.
NE	Does not equal.
LE	Is less than or equal to.
LT	Is less than.
GE	Is greater than or equal to.
GT	Is greater than.

value The value to compare against *fieldname*.

B'value'	Bit value.
C'text'	Character string.
D'number'	Packed decimal number.
F'value'	Fixed decimal.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
HIGHVALUE	Reserved keyword indicating field filled with X'FF'.
LOWVALUE	Reserved keyword indicating field filled with X'00'

AND|OR Logical operators to construct a compound condition specifying multiple *fieldnames* and conditions.

Examples:

```
DISPLAY TAX-EXEMPT WHEN EMPLOYEE-TYPE EQ 1
OR EMPLOYEE-AGE GT 120
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
DISPLAY ACCT-BUSINESS WHEN
ACCOUNT-TYPE EQ X'02' <196>
AND ACCOUNT-BRANCH EQ X'03' <196>
AND ACCT-IMPOUND NE 'YES'
```

FORMAT Statement

The FORMAT statement determines the format the dataset will display.

FORMAT DISPLAY(({STD|FORMAT|COMPACT|HEX})
PAD(char)][SKIP({ON|OFF})]

Operands

DISPLAY	FORMAT	Use formatted mode.
	COMPACT	Use compact mode.
	STD	Use standard mode.
	HEX	Use hexadecimal mode.

PAD The padding character to use for edit operations, which require padding such as record length changes.
The default is the period (.).

SKIP ON - Skip to the next field when input is being typed and the cursor reaches the end of a field.
OFF - Do not skip to the next field.

Example:

```
FORMAT DISPLAY(STD)
FORMAT DISPLAY(COMPACT) PAD(.)
FORMAT DISPLAY(COMPACT) SKIP(ON) PAD(' ')
FORMAT DISPLAY(FORMAT) SKIP(OFF)
```

SYSIN Statement

The SYSIN statement adds COBOL statements to those specified in the member.

Format:

SYSIN {'statement'|DLM(string)}

statement The COBOL or PL/1 statement to add to those in the copybook(s) specified. It must be enclosed in quotes.

DLM Define the delimiter string. A delimiter provides a method to include multiple COBOL or PL/1 statements. The delimiter indicates the end of the statements.

string The delimiter indicating the end of the added COBOL or PL/1 statements.

Examples:

```
COPY NEWMEMB
SYSIN '05 ACCT-TYPE X(9)'
COPY MEMBA
SYSIN DLM(/*)
05 ACCT-TYPE X(9)
05 ACCT-SUPV X(9)
05 ACCT-PRTY X(9)
/*
COPY MEMBB
```

SIGNON Statement

If a copybook template is not marked as global (GBL), the SIGNON statement must precede the COMPARDS statement if copybook or reformat templates are used.

SIGNON **USERID**(userid)

where:

userid The userid for the TEMPLATE..

Example:

```
SIGNON USERID(TSO21)
COMPILE FUNC(COPYBOOK) -
LANGUAGE({COBOL|PLI}) -
MEMBER(member) -
USERID(userid) -
TITLE(title)
```

COMPILE Control Statement Examples

Examples:

```
COMPILE FUNC(COPYBOOK)
MEMBER(ACCTMAIN) <196>
USERID(JOE012)
TITLE(Major customer accounts)
COPYLIB PAYROLL.TAX1994.COBOL
COPY ADDTAX
```

DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
MEND

COMPILE FUNC(COPYBOOK)
 MEMBER(PARTS)
 USERID(GLOBAL)
 TITLE(Manufacturing Parts)
COPYLIB PARTS.CICS.COBOL
COPY LOCPART
DISPLAY INDEPENDENT-PART WHEN USED-IN EQ 0
DISPLAY MAIN-PART WHEN USED-BY-PART EQ 0
DISPLAY SUB-PART WHEN USED-IN NE 0
MEND

COMPILE FUNC(COPYBOOK) -
 LANGUAGE({COBOL|PLI}) -
 MEMBER(member) -
 USERID(userid) -
 TITLE(title)
FORMAT DISPLAY(STD)
TESTDSN PAYROLL.TEST.MARVEL
COPYLIB CICS.TEST.COBOL
COPYLIB PAYROLL.TAX1994.COBOL
COPY ADDTAX
SYSIN '05 TAX-EXEMPT X(9)'
DISPLAY TAX-EXEMPT WHEN EMPLOYEE-TYPE EQ 1
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
COPYLIB CICS.TEST.COBOL
COPY ADDTAX
FORMAT DISPLAY(COMPACT) SKIP(ON) PAD(')
DISPLAY TAX-EXEMPT WHEN EMPLOYEE-TYPE EQ 1 - OR EMPLOYEE-AGE GT 120
DISPLAY TAX-PAYMENTS WHEN PAYMENT-TOTAL GT 0
DISPLAY ACCT-BUSINESS WHEN -
 ACCOUNT-TYPE EQ X'02' AND -
 ACCOUNT-BRANCH EQ X'03' -
 AND ACCT-IMPOUND NE 'YES'

MEND

COMPARDS - Compare Datasets Function

Datasets may be compared via a batch job. The parameters and submission can be handled automatically by a panel (see Section 8.5 Compare File Entry Panel) or entered and submitted manually.

COMPARDS, the FileMarvel batch compare utility, compares two files, selective fields within records of a file or selective records of a dataset and prints the records in error in hexadecimal, character, or reformat template format modes. The COMPARDS command processes VSAM, ISAM, tape, PDS, keyed and non-keyed sequential datasets. Records are retrieved by sequence, key, relative record or RBA. Records are compared by key ranges or Boolean selection criteria. Entire partitioned datasets or member ranges may be compared. Error records are printed as line of error or entire records and in one of several optional formats: standard, ISPF hex format, reformat and hex reformat templates formats.

Specific fields within records may be compared using REFORMAT templates defined in any BROWSE, EDIT or COPYBOOK panels. A variety of operands are listed below to assist in tailoring the file compare. All options set as well as execution results are listed in the MESSAGE sysout dataset and all record discrepancies are formatted and printed in the REPORTS sysout dataset. Multiple COMPARDS commands may be entered in order to compare separate pairs of datasets.

COMPARDS Statement

COMPARDS

-
{File1(*ddname*) | DataSet1(*datasetname*)} -
[VOLUME1(*volser*)] -
{File2(*ddname*) | DataSet2(*datasetname*)} -
[VOLUME2(*volser*)] -
[COMPARE(Standard | ENHanced)] -
[COPYBOOK(*template* | *template:library*)] -
[REFORMAT(*template* | *template:library*)] -
[FORMAT({Hex | Standard | FORMAT | REFormat | HEXRef})] -
[REPORT({ERRor | REcOrd})] -
[KEYS({ON | OFF})] -
[WIDTH({Standard | MAXimum})] -
[ERRCHAR(*c*)] -
[LENERR({ON | OFF})] -
[RANGE(ASM | COB | JCL | PLD)] -
[RESYNCH({ON | OFF})] -
[SYNCHCNT(*nnnn*)] -
[STOPLMT(*nnnn*)] -
[STOPTOT(*nnnn*)] -
[KEYPOS(*nnnn nnnn*)] -
 [KEYPOS1(*nnnn nnnn*)] -
 [KEYPOS2(*nnnn nnnn*)] -
[RECORDS({count | ALL})] -
 [RECORDS1({count | ALL})] -
 [RECORDS2({count | ALL})] -
[StartREC(*rec*) [EndREC(*endrec*)] -
 [StartRE1(*rec*) [EndREC1(*endrec*)] -
 [StartRE2(*rec*) [EndREC2(*endrec*)] -
StartKEY(*key*) [EndKEY(*endkey*)] -
 StartKE1(*key*) [EndKEY1(*endkey*)] -
 StartKE2(*key*) [EndKEY2(*endkey*)] -
StartRBA(*rba*) [EndRBA(*endrba*)] -
 StartRB1(*rba*) [EndRBA1(*endrba*)] -
 StartRB2(*rba*) [EndRBA2(*endrba*)] -
StartREL(*rrn*) [EndREL(*endrrn*)] -
 StartRL1(*rrn*) [EndREL1(*endrrn*)] -
 StartRL2(*rrn*) [EndREL2(*endrrn*)] -
StartMEM(*member*) [EndMEM(*endmember*)] -
 StartMEM1(*member*) [EndMEM1(*endmember*)] -
 StartMEM2(*member*) [EndMEM2(*endmember*)] -
[FIELD(*copybook field name*) [FIELD(*copybook field name*)...] -
[SElect(*criteria*) [{AND|OR} SElect(*criteria*)...]

Operands

FILE1(*ddname*) Pre-allocated 1- to 8-character DD name specifying the input dataset one to compare.
FILE1 can be abbreviated F1.

DATASET1(*datasetname*) 1- to 44-character input dataset name to compare. DATASET1 can be abbreviated DS1.

VOLUME1(*volser*) 1- to 6-character dataset volume serial name. Required if the dataset is not cataloged.
VOLUME1 can be abbreviated VOL1.

FILE2 (<i>ddname</i>)	Pre-allocated 1- to 8-character DD name specifying the input dataset two to compare. FILE2 can be abbreviated F2.
DATASET2 (<i>datasetname</i>)	1- to 44-character input dataset name to compare. DATASET2 can be abbreviated DS2.
VOLUME2 (<i>volser</i>)	1- to 6-character dataset volume serial name. Required if the dataset is not cataloged. VOLUME2 can be abbreviated VOL2.
COMPARE	Type of compare requested i.e., standard record for record or enhanced which attempts to synchronize imbedded records and resume the compare.
STANDARD	Compare record for record and isolate the discrepancies.
ENHANCED	Compare record for record and absorb a single record or a group of records that are imbedded and resume the compare.
RANGE	Restrict the compare to specific text fields for compilers avoiding the sequencing fields.
ASM	Compare assembler text columns 1 through 72, ignoring 73 through 80.
COB	Compare COBOL text columns 7 through 72 ignoring 1 through 6 and 73 through 80.
JCL	Compare columns 1 through 72, ignoring 73 through 80.
PLI	Compare columns 1 through 72, ignoring 73 through 80.
FORMAT	Report format type.
HEX	Format the record in an ISPF hexadecimal display. Can be abbreviated H.
STANDARD	Format the record in a standard display, translating all non-display data to periods. Can be abbreviated STD.
FORMAT	Compare and format the records in copybook format mode according to the template specified in the COPYBOOK() keyword.
REFORMAT	Format the records according to the template specified in the REFORMAT() keyword. Can be abbreviated as REF.
HEXREF	Format the records according to the template specified in the REFORMAT() keyword in ISPF hexadecimal format. Can be abbreviated as HEXR.

COPYBOOK parameters

FORMAT (FORMAT)	Compare the records according to the template specified in the COPYBOOK() keyword
COPYBOOK (template)	Name of a compiled copybook template. Required for FORMAT , COMPACT and for formatted FIELD statements. Add the library name if a SIGNON statement is not used. Example: COPYBOOK (COBBOOK : TSO21) Select record fields based on the copybook field names using one or more FIELD statements.
FIELD (field name) -	Field name specified in the COPYBOOK () keyword.
FIELD (field name)	Next field name

REFORMAT parameters

FORMAT(REFORMAT/HEXREF)	Format the records according to the template specified in the REFORMAT() keyword. Can be abbreviated as REF
REFORMAT(template)	Name of a compiled reformat template. Add the library if the SIGNON statement is not used. Required for REFORMAT formats. Example: REFORMAT(REFBOOK:TZO21)
WIDTH	Output format column width. WIDTH is ignored with special formatting. (Default: MAXIMUM.)
STANDARD	Truncate the format to an 80-character image. Can be abbreviated STD.
MAXIMUM	Format the output record to the full width of the page. Can be abbreviated MAX.
KEYS	Determines if the record key is displayed. (Default: ON.)
ON	Display record keys.
OFF	Do not display record keys.
REPORT	Report record type.
ERRor	Print the lines with discrepancies only. Do not print the entire record.
RECORD	Print the entire record if discrepancies occur.
ERRCHAR	A special character indicating the discrepancies between records. The underscore(_) is the default special character.
LENERR	Option determination for records compared when the record lengths are not the same.
ON	Treat different records lengths as error.
OFF	If records compare, treat different record length as a non-errors.
RESYNCH	Option to resynchronize keys if records keys are not the same.
ON	Yes, resynch the keys.
OFF	No, do not resynch the keys.
SYNCHCNT	The number of retries to synchronize a file when imbedded records occur on keyed files.
n	A number, 1-999.
KEYPOS	A two position parameter indication a common field sorted in ascending order in which sequential files may be synchronized. SYNCCNT will use this file position in an attempt to synchronize the files.
n	Position one is the relative position of the field based on 1. A number within the bounds of the record.

n	Position two is the length of the field beginning at position one. Example: KEYPOS(1,44)
STOPLMT	The number of consecutive record discrepancies at which the compare will be terminated.
n	A number, 1-999.
STOPTOT	The total number of record discrepancies at which the compare will be terminated.
n	A number, 1-999.
RECORDS(count)	The maximum number of records to compare. (Default: installation option.) 1 to 8 digits.
n	A number, 1-9999999.
ALL	Print all records in the file.
RECORDS1(count)	Same as RECORDS but applies to FILE1 only.
RECORDS2(count)	Same as RECORDS but applies to FILE2 only.

Record positioning keywords consist of the pairs STARTREC-ENDREC, STARTKEY-ENDKEY, STARTRBA-ENDRBA or (for PDFs only) STARTMEM-ENDMEN. Each of these pairs is mutually exclusive from the other pairs. In each case, the END of the pair is optional. The default is the end of the dataset. Each pair of options also contains a pair of controls designated for each file which will override the original control.

STARTREC(rec)	1- to 8-digit starting record number. STARTREC can be abbreviated SREC.
STARTRE1(count)	Same as STARTREC but applies to FILE1 only.
STARTRE2(count)	Same as STARTREC but applies to FILE2 only.
ENDREC(endrec)	1- to 8-digit ending record number. ENDREC can be abbreviated EREC.
ENDREC1(count)	Same as ENDREC but applies to FILE1 only.
ENDREC2(count)	Same as ENDREC but applies to FILE2 only.
STARTKEY(key)	1- to 64-byte starting key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes, and/or any combination. Exact or generic keys are located depending on the length of the key. STARTKEY can be abbreviated SKEY.
	Where <i>key</i> :
	B'value' Binary value.
	C'text' Character string.
	X'value' Hexadecimal value.
	string Character string without imbedded blanks
	'string' Character string with imbedded blanks
	X'hex'char Mixed hex and character mode
STARTKE1(count)	Same as STARTKEY but applies to FILE1 only.
STARTKE2(count)	Same as STARTKEY but applies to FILE2 only.

ENDKEY(endkey)	1- to 64-byte ending key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes and/or any combination. Exact or generic keys are located depending on the length of the key. ENDKEY can be abbreviated EKEY.
-----------------------	---

Where *key*:

B'value' Binary value.
C'text' Character string.
X'value' Hexadecimal value.
string Character string without imbedded blanks
'string' Character string with imbedded blanks
X'hex'char Mixed hex and character mode

ENDKEY1(count) Same as ENDKEY but applies to FILE1 only.
ENDKEY2(count) Same as ENDKEY but applies to FILE2 only.

STARTRBA(rba) Starting hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-than conditions are not applicable to RBA processing.) STARTRBA can be abbreviated SRBA.

STARTRB1(count) Same as STARTRBA but applies to FILE1 only.
STARTRB2(count) Same as STARTRBA but applies to FILE2 only.

ENDRBA(endrba) Ending hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-than conditions are not applicable to RBA processing.) ENDRBA can be abbreviated ERBA.

ENDRBA1(count) Same as ENDRBA but applies to FILE1 only.
ENDRBA2(count) Same as ENDRBA but applies to FILE2 only.

STARTREL(rrn) 1- to 8-digit starting relative record number. STARTREL can be abbreviated SREL.

STARTRL1(rrn) Same as STARTREL but applies to FILE1 only.
STARTRL2(rrn) Same as STARTREL but applies to FILE2 only.

ENDREL(rrn) 1- to 8-digit ending relative record number. ENDREL can be abbreviated EREL.

ENDREL1(rrn) Same as ENDREL but applies to FILE1 only.
ENDREL2(rrn) Same as ENDREL but applies to FILE2 only.

STARTMEM(member) Starting member, for partitioned datasets only. The member specified must actually exist in the PDS. STARTMEN can be abbreviated SMEN.

STARTME1(member) Same as STARTMEM but applies to FILE1 only.
STARTME2(member) Same as STARTMEM but applies to FILE2 only.

ENDMEM(member) Ending member, for partitioned datasets only. The member specified must actually exist in the PDS. ENDMEN can be abbreviated EMEM.

ENDMEM1(member) Same as ENDMEM but applies to FILE1 only.
ENDMEM2(member) Same as ENDMEM but applies to FILE2 only.

SELECT Statement

Select records based on their contents using one or more SELECT statements. If more than one SELECT statement is specified, they must be separated by AND or OR keywords.

Format:

```
SELECT(FIND structure value cond)  
    [{AND|OR}  
SELECT ...
```

where:

value The value to compare.

B'value'	Binary value.
C'text'	Character string.
D'number'	Packed decimal number.
Q'value'	Bit value.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode
HIGHVALUE	Reserved keyword indicating field filled with X'FF'.
LOWVALUE	Reserved keyword indicating field filled with X'00'

cond The condition to compare the data in the dataset represented.

EQ	Equals.
NE	Does not equal.
LE	Is less than or equal to.
LT	Is less than.
GE	Is greater than or equal to.
GT	Is greater than.

AND|OR Logical operators to construct a compound condition specifying multiple *fieldnames* and conditions.

Sequential dataset

```
COMPARDS DS1(dataset) DS2( dataset2) -  
SELECT('811203' 1 10 )    AND -  
SELECT('811117' 1 10 )    OR -  
SELECT('820128')    AND -  
SELECT('820220') -  
RECORDS(10)
```

SIGNON Statement

If a copybook template is not marked as global (GBL), the SIGNON statement must precede the COMPARDS statement if copybook or reformat templates are used.

SIGNON **USERID**(*userid*)

where:

userid The userid for the TEMPLATE..

Example:

```
SIGNON USERID(TSO21)
COMPARDS DS1(ACCOUNTS.PAY02.VSAM.JAN) -
DS2(DATA.SET.TWO) -
REFORMAT($MYREFMT) -
FORMAT(STD) -
STARTKEY(X'00'CONTROL '01') -
RECORDS(1)
```

COMPARDS Control Statement Examples

Examples:

Sequential dataset with external files

```
COMPARDS FILE1(DDN1) FILE2(DDN2) -
REPORT(ERROR) -
STOPLMT(99) -
STOPTOT(999) -
ERRCHAR(+) -
KEYPOS(10 15) -
SYNCCNT(25) -
STARTRE1(10) -
STARTRE2(25) -
RECORDS(500)

//DDN1
//DDN2
```

Unformatted sequential dataset

```
COMPARDS DS1(Dataset) DS2(dataset2) -
FORMAT(FORMAT) COPYBOOK(CBKCOB:TSO09) -
FIELD(COB-FIELD1) -
FIELD(COB-FIELD2) -
FIELD(COB-FIELD3)
```

Unformatted sequential dataset

```
COMPARDS DS1(Dataset) DS2(dataset2)
```

Unformatted sequential dataset

```
COMPARDS DS2(Dataset) DS2(dataset2) -
RECORDS(10) -
FORMAT(HEX) WIDTH(MAXIMUM) -
SELECT(X'00'CONTROL.P'025C' EQ 1 50) OR -
SELECT(CONTROL EQ 50 100)        AND -
SELECT(P'025C' GT 100)
```

VSAM dataset comparing specific fields

COMPARDS DS1(Dataset) DS2(Dataset2) -
REFORMAT(template) -
FORMAT(HEXREF) -
STARTKEY(X'00'C'CONTROL 'X'01') -
RECORDS(25)

VSAM KSDS dataset

COMPARDS DS1(Dataset) DS2(Dataset2) -
STARTKEY(X'00'C'CONTROL 'X'01') -
ENDKEY(X'01'C'EXECUTE') -
FORMAT(STANDARD) WIDTH(MAXIMUM)

VSAM ESDS dataset

COMPARDS DS1(Dataset) DS2(Dataset2) -
FORMAT(STANDARD) -
WIDTH(STANDARD) -
SRBA1(x'00000050') -
ERBA2(x'fefefefe')

VSAM RRDS dataset

COMPARDS DS1(Dataset) DS2(Dataset2) -
STARTREC(15) -
SREL1(50) -
SREL2(55) -
RECORDS(05) -
FORMAT(HEX) -
WIDTH(MAXIMUM)

Selected members of a PDS

COMPARDS DS1(Dataset) DS2(Dataset2) -
STARTMEM(mem1) -
ENDMEM(memn)

Particular member of a PDS

COMPARDS DS1(Dataset(mem)) DS2(Dataset2(mem)) -
RECORDS(30) -
STATS(OFF)

Select all members of a PDS

COMPARDS DS1(Dataset(*)) DS2(Dataset2(*))

Compare PDS directory only

COMPARDS DS1(Dataset) DS2(Dataset2)

COPYDS - Copy Dataset Function

Datasets may be copied via a batch job. The parameters and submission can be handled automatically by an online function COPYDO or entered and submitted manually. You can use the COPYDS command to copy entire files or selected records. The input and output files can differ in organization. COPYDS is a convenient method to merge keyed files. Invoke COPYDS repeatedly and specify a different source dataset to create a merged file.

Target record lengths may either be truncated or elongated. Installation specified pad characters can be used to fill the target record length.

Records may be expanded, reduced, fields inserted within records and reformatted as desired. Fields within a record may be moved around using the REFORMAT technique. A field for example, a date may be expanded. For example a file containing a field with the format MMDDYY may be expanded to MMDDYYYY. The REFORMAT technique may be used to create test files as well. Another example may be moving COBOL source sequence from position 73-80 to position 1-6. Specific records may be selected from one file to another using multiple Boolean logic selection criteria.

The following combinations are permitted:

Sequential FB/VB/U	to a	Sequential FB/VB/U
Sequential FB/VB/U	to a	VSAM KSDS/ESDS/RRDS
Sequential FB/VB/U	to a	PDS member FB/VB
PDS member FB/VB	to a	PDS member FB/VB
PDS member U	to a	PDS member U
PDS member FB/VB/U	to a	Sequential FB/VB/U
VSAM KSDS/ESDS/RRDS	to a	VSAM KSDS/ESDS/RRDS
VSAM KSDS/ESDS/RRDS	to a	Sequential FB/VB/U

The following combinations are not permitted:

Sequential FB/VB/U	to a	PDS member U
PDS member FB/VB/U	to a	VSAM KSDS/RRDS
PDS member FB/VB	to a	PDS member U

COPYDS Statement

COPYDS -
 {**IFile**(*ddname*) | **IDataSet**(*datasetname*)} -
 [**IVOLume**(*volser*)] -
 {**OFile**(*ddname*) | **ODataSet**(*datasetname*)} -
 [**OVOLume**(*volser*)] -
 [**COPYBOOK**(*template* | *template:library*)] -
 [**REFORMAT**(*template* | *template:library*)] -
 [**FORMAT**(({Hex | **STandard** | **FORMAT** | **REFormat** | **HEXRef**})|)] -
 [**FIELD**({**ON** | **OFF**})] -
 [**REPORT**({**ERRor** | **RECord**})] -
 [**KEYS**({**ON** | **OFF**})] -
 [**WIDTH**({**STandard** | **MAXimum**})] -
 [**LENERR**({**ON** | **OFF**})] -
 [**STOPLMT**(*nnnn*)] -
 [**KEYPOS**(*nnnn nnnn*)] -
 [**RECORDs**({*count* | **ALL**})] -
 [**StartREC**(*rec*) [**EndREC**(*endrec*)] -
 StartKEY(*key*) [**EndKEY**(*endkey*)] -
 StartRBA(*rba*) [**EndRBA**(*endrba*)] -
 StartREL(*rrn*) [**EndREL**(*endrrn*)] -
 StartMEM(*member*) [**EndMEM**(*endmember*)] -
 SKipRec(*rnum* | *srec-erec*) -
 [**SElect**(*criteria*) [**{AND|OR} SElect**(*criteria*)...]

Operands

IFILE(*ddname*) Pre-allocated 1- to 8-character DD name specifying the input dataset one to copy. IFILE can be abbreviated IF.

IDATASET(*datasetname*) 1- to 44-character input dataset name to copy. IDATASET can be abbreviated IDS.

IVOLUME(*volser*) 1- to 6-character dataset volume serial name. Required if the dataset is not cataloged. IVOLUME can be abbreviated IVOL.

OFILE(*ddname*) Pre-allocated 1- to 8-character DD name specifying the input dataset two to copy. OFILE can be abbreviated OF.

ODATASET(*datasetname*) 1- to 44-character input dataset name to copy. ODATASET can be abbreviated ODS.

OVOLUME(*volser*) 1- to 6-character dataset volume serial name. Required if the dataset is not cataloged. OVOLUME can be abbreviated OVOL.

FORMAT Report format type.

HEX Format the record in an ISPF hexadecimal display. Can be abbreviated H.

STANDARD Format the record in a standard display, translating all non-display data to periods. Can be abbreviated STD.

FORMAT Format the records according to the template specified in the COPYBOOK() keyword.

REFORMAT Format the records according to the template specified in the REFORMAT() keyword.

Can be abbreviated as REF.

HEXREF Format the records according to the template specified in the REFORMAT() keyword in ISPF hexadecimal format. Can be abbreviated as HEXR.

COPYBOOK parameters

FORMAT(FORMAT) Copy the records according to the template specified in the COPYBOOK() keyword

COPYBOOK(template) Name of a compiled copybook template. Required for FORMAT, COMPACT and for formatted SELECT statements. Add the library name if a SIGNON statement is not used. Example: COPYBOOK(COBBOOK:TSO21)

FIELD Determines to use the SELECT copybook field names to selective field comparisons (Default: OFF.)

ON Use the SELECT field names to copy on field values.
OFF Use the SELECT field names for record criteria search.

REFORMAT parameters

FORMAT(REFORMAT/HEXREF) Format the records according to the template specified in the REFORMAT() keyword. Can be abbreviated as REF

REFORMAT(template) Name of a compiled reformat template. Add the library if the SIGNON statement is not used. Required for REFORMAT formats. Example: REFORMAT(REFBOOK:TSO21)

WIDTH Output format column width. WIDTH is ignored with special formatting. (Default: MAXIMUM.)

STANDARD Truncate the format to an 80-character image. Can be abbreviated STD.

MAXIMUM Format the output record to the full width of the page. Can be abbreviated MAX.

KEYS Determines if the record key is displayed. (Default: ON.)

ON Display record keys.
OFF Do not display record keys.

REPORT Report record type.

ERRor Print the lines with discrepancies only. Do not print the entire record.

RECORD Print the entire record if discrepancies occur.

LENERR Option determination for records copyd when the record lengths are not the same.

ON Treat different records lengths as error.

OFF If records copy, treat different record length as a non-errors.

KEYPOS	A two positional parameter indication a common field sorted in ascending order in which sequential files may be synchronized. SYNCCNT will use this file position in an attempt to synchronize the files.
n	Position one is the relative position of the field based on 1. A number within the bounds of the record.
n	Position two is the length of the field beginning at position one.
STOPLMT	The number of consecutive record discrepancies at which the copy will be terminated.
n	A number, 1-999.
RECORDS(count)	The maximum number of records to copy. (Default: installation option.) 1 to 8 digits.
n	A number, 1-9999999.
ALL	Print all records in the file.

Record positioning keywords consist of the pairs STARTREC-ENDREC, STARTKEY-ENDKEY, STARTRBA-ENDRBA or (for PDSs only) STARTMEM-ENDMEN. Each of these pairs is mutually exclusive from the other pairs. In each case, the END of the pair is optional. The default is the end of the dataset. Each pair of options also contains a pair of controls designated for each file which will override the original control.

STARTREC(rec) 1- to 8-digit starting record number. STARTREC can be abbreviated SREC.

ENDREC(endrec) 1- to 8-digit ending record number. ENDREC can be abbreviated EREC.

STARTKEY(key) 1- to 64-byte starting key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes, and/or any combination. Exact or generic keys are located depending on the length of the key. STARTKEY can be abbreviated SKEY.

Where *key*:

B'value'	Binary value.
C'text'	Character string.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode

ENDKEY(endkey) 1- to 64-byte ending key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes and/or any combination. Exact or generic keys are located depending on the length of the key. ENDKEY can be abbreviated EKEY.

Where *key*:

B'value'	Binary value.
C'text'	Character string.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode

STARTRBA(rba) Starting hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-

than conditions are not applicable to RBA processing.) STARTRBA can be abbreviated SRBA.

- ENDRBA(*endrba*)** Ending hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-than conditions are not applicable to RBA processing.) ENDRBA can be abbreviated ERBA.
- STARTREL(*rrn*)** 1- to 8-digit starting relative record number. STARTREL can be abbreviated SREL.
- ENDREL(*rrn*)** 1- to 8-digit ending relative record number. ENDREL can be abbreviated EREL.
- STARTMEM(*member*)** Starting member, for partitioned datasets only. The member specified must actually exist in the PDS. STARTMEN can be abbreviated SMEN.
- ENDMEM(*member*)** Ending member, for partitioned datasets only. The member specified must actually exist in the PDS. ENDMEN can be abbreviated EMEM.

SKIPREC Statement

Input records may be skipped writing to the output dataset using the keyword SKIPREC. A single record or a block of records may be skipped in each entry of SKIPREC. Records skipped may be printed using the REPORT(RECORD) option.

Format:

SKIPREC(rnum) -
or
SKIPREC(srec-erec) -

SKIPREC

Rnum	The record number of a single record to be skipped
Srec	The start record number of a block of records to be skipped
Erec	The end record number of a block of records to be skipped
SKIPREC	Next skip statement

SELECT Statement

Select records based on their contents using one or more SELECT statements. If more than one SELECT statement is specified, they must be separated by AND or OR keywords.

Format:

```
SELECT(FIND structure value cond)  
    [{AND|OR}  
SELECT ...
```

where:

value The value to copy.

B'value'	Binary value.
C'text'	Character string.
D'number'	Packed decimal number.
Q'value'	Bit value.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode
HIGHVALUE	Reserved keyword indicating field filled with X'FF'.
LOWVALUE	Reserved keyword indicating field filled with X'00'

cond The condition to copy the data in the dataset represented.

EQ	Equals.
NE	Does not equal.
LE	Is less than or equal to.
LT	Is less than.
GE	Is greater than or equal to.
GT	Is greater than.

AND|OR Logical operators to construct a compound condition specifying multiple *fieldnames* and conditions.

Sequential dataset

```
COPYDS IDS(dataset) ODS( dataset2) -  
SELECT('811203' 1 10 )    AND -  
SELECT('811117' 1 10 )    OR -  
SELECT('820128')    AND -  
SELECT('820220') -  
RECORDS(10)
```

SIGNON Statement

If a copybook template is not marked as global (GBL), the SIGNON statement must precede the COPYDS statement if copybook or reformat templates are used.

SIGNON **USERID**(*userid*)

where:

userid The userid for the TEMPLATE..

Example:

```
SIGNON USERID(TSO21)
COPYDS  IDS(ACCOUNTS.PAY02.VSAM.JAN) -
ODS(DATA.SET.TWO) -
REFORMAT($MYREFMT) -
FORMAT(STD) -
STARTKEY(X'00'CONTROL '01') -
RECORDS(1)
```

COPYDS Control Statement Examples

Examples:

Sequential dataset with external files

```
    COPYDS IFILE(DDN1) OFILE(DDN2) -
    REPORT(ERROR) -
    STOPLMT(99) -
    STOPTOT(999) -
    ERRCHAR(+) -
    KEYPOS(10 15) -
    SYNCCNT(25) -
    STARTRE1(10) -
    STARTRE2(25) -
    RECORDS(500)

//DDN1
//DDN2
```

Unformatted sequential dataset

```
    COPYDS IDS(Dataset) ODS(dataset2)
```

Unformatted sequential dataset

```
    COPYDS IDS(Dataset) ODS(dataset2) -
    RECORDS(10) -
    FORMAT(HEX) WIDTH(MAXIMUM) -
    SELECT(X'00'CONTROL.P'025C' EQ 1 50) OR -
    SELECT(CONTROL EQ 50 100)        AND -
    SELECT(P'025C' GT 100)
```

VSAM dataset comparing specific fields

```
    COPYDS IDS(Dataset) ODS(Dataset2) -
    REFORMAT(template) -
    FORMAT(HEX) -
    STARTKEY(X'00'C'CONTROL 'X'O1') -
    RECORDS(25)
```

VSAM KSDS dataset

COPYDS IDS(Dataset) ODS(Dataset2) -
STARTKEY(X'00'C'CONTROL 'X'O1') -
ENDKEY(X'O1'C'EXECUTE') -
FORMAT(STANDARD) WIDTH(MAXIMUM)

VSAM ESDS dataset

COPYDS IDS(Dataset) ODS(Dataset2) -
FORMAT(STANDARD) -
WIDTH(STANDARD) -
SRBA1(x'00000050') -
ERBA2(x'fefefefe')

VSAM RRDS dataset

COPYDS IDS(Dataset) ODS(Dataset2) -
STARTREC(15) -
SREL1(50) -
SREL2(55) -
RECORDS(05) -
FORMAT(HEX) -
WIDTH(MAXIMUM)

Selected members of a PDS

COPYDS IDS(Dataset) ODS(Dataset2) -
STARTMEM(meml) -
ENDMEM(memn)

Particular member of a PDS

COPYDS IDS(Dataset(mem)) ODS(Dataset2(mem)) -
RECORDS(30) -
STATS(OFF)

Select all members of a PDS

COPYDS IDS(Dataset(*)) ODS(Dataset2(*))

Copy PDS directory only

COPYDS IDS(Dataset) ODS(Dataset2)

LABEL - Tape and Cartridge Label Display Function

LABEL function displays tape or cartridge labels for standard labeled files. The batch processor formats and optionally prints the labels in the desired format. The label display may optionally be displayed as a TSO notification. See the FileMarvel administrator to activate TSO notification authorization.

LABEL Statement

LABEL {FILE(*ddname*)} -
 {DSN(*dataset name*)} -
 VOLUME(*volser*) -
 UNIT(*unitname*) -
 SEQ(*sequence number*) -
 [FORMAT{OFF | STD | HEX | HEXD}]]
 [WIDTH({STandarD | MAXimum})]
 [NOTIFY({OFF | TSO | WTO})]

Operands

FILE(*ddname*) Optional, pre-allocated 1- to 8-character DD name specifying the input dataset to format. If FILE is omitted, then the file is dynamically allocated.

Example FILE(DD1)

```
//DD1    DD DISP=(OLD,KEEP),EXPDT=98000,  
          //      LABEL=(013,BLP),  
          //      VOL=SER=111115,  
          //      UNIT=CART,  
          //      DSN=MOUNT.VOLUME.LABEL
```

DSN() Optional 1 - 44 character dataset name used as the mount message.

Example DSN(MOUNT.VOLUME.LABEL)

VOLUME() Required 6-character volume serial name for the tape or cartridge of the standard labeled unit.

Example VOLUME(123115)

UNIT() Required 1 - character esoteric unit name for the volume.

Example UNIT(CART)

SEQ() Required 1 - numeric file sequence for the dataset label to be displayed. This sequence is the logical file number and not the actual file sequence. The correct file is calculated by FileMarvel.

Example SEQ(3) (3rd dataset on the device)

FORMAT() Optional request to print the actual label records in the desired format.

OFF Do not print the actual labels

STD Format the labels in standard mode

HEX	Format the labels in ISPF hex mode
HEXD	Format the labels in hex dump mode
WIDTH()	Optional request to print the formatted records in either 80 characters or the max line width.
STD	Format the labels in standard 80 character width.
MAX	Format the labels in maximum line width.
NOTIFY()	Optional request to notify the requestor on completion of label processing.
OFF	Do not notify the TSO requestor.
TSO	Send the formatted label as a TSO notification. This option requires authorization by the administrator.
WTO	Send the formatted label as a message on the job JES log.

LABEL Control Statement Examples

Examples:

```
SIGNON USERID(TSODG )
```

```
LABEL -
  DSN(MOUNT.VOLUME.LABEL) -
  VOL(115206) -
  SEQ(7) -
  UNIT(CART) -
  FORMAT(STD) -
  NOTIFY(TSO)
```

```
LABEL -
  FILE(LABEL1) -
  FORMAT(HEX) -
  NOTIFY(WTO)
```

```
//LABEL1 DD DISP=(OLD,KEEP),EXPDT=98000,
// LABEL=(013,BLP),
// VOL=SER=111115,
// UNIT=CART,
// DSN=MOUNT.VOLUME.LABEL
```

LISTDIR – List Procedure, Copybook, Manual Function

LISTDIR prints an entire or selective procedure, copybook or manual directories.

Command format

```
LISTDIR      FUNC(PROC/COPYBOOK/MANUAL)      -  
            OFILE(ddname)                    -  
            MEMBER(member*) USERID(userid)   -  
            GENERIC
```

Keywords

FUNCTION() Specifies the requested function. If the MEMBER is omitted, a directory list is produced.
FUNC() Default: PROC

PROC	List the procedure directory
COPYBOOK	List the copybook directory
MANUAL	List the manual directory

OFILE(ddname) Pre-allocated report file ddname.
OF() Default: FMVREPTS

MEMBER() Optional 1- to 8-character member name. Entering a member prefix followed by an *, will
limit the directory list to the string specified. Refer to the examples below.

USERID() Optional 1 - 8 TSO userid.

GENERIC To list all members beginning with the member specified. Another technique to
* list a specific group of members is to suffix the member with an asterisk.

Examples

```
LISTDIR FUNC(PROC) /* print the entire procedure */  
                /* directory */
```

```
LISTDIR FUNC(PROC) - /* print the directory */  
    MEMBER(mem*) /* members prefixed by "mem" */
```

```
LISTDIR FUNC(COPYBOOK) /* print the entire directory */
```

```
LISTDIR FUNC(MANUAL) /* print the entire directory */
```

LISTVOL – List Volume Dataset Function

LISTVOL prints an entire or selected volumes for free space or volume dataset list.

Command format

```
LISTVOL      FUNC(SPACE/VTOC)      -
             OFILE(report ddname)   -
             OPTION()                -
             FORMAT()                -
             COPYBOOK()             -
             SORTFLD()              -
             STARTREC()             -
             RECORDS()              -
             VOL()
```

VOL(volser)

Keywords all functions

FUNCTION()	Specifies the requested function.
FUNC()	
SPACE	List the volume free space.
VTOC	List the volume datasets.
OFILE(ddname)	Pre-allocated report file ddname.
OF()	Default: FMVREPTS
VOLUME()	Required volume, generic volume or volume types
VOL()	
volser / volser*	1- to 6-character volume or generic volume
PRIVATE	display all private mounted volumes
PUBLIC	display all public mounted volumes
STORAGE	display all storage mounted volumes
ALL	all volumes

Keywords VTOC

OPTION()	Display option
LIST	display datasets and summary information
DATASET	display datasets only
SUMMARY	display summary only
FORMAT()	Display format
STANDARD	display dataset line - default
HEX	display each vtoc record in hex mode
DUMP	display each vtoc record in hex dump mode
FORMAT	display each vtoc record in copybook format

COPYBOOK(*copybook*) Name of a compiled copybook template. Required for FORMAT.

DEFAULT: COPYBOOK(CBKVTOC:SYSTEM)

SORTFLD() Sort the volume display according to the field name specified. You can sort by any field name in the dataset selection list.

DATASET	dataset name
DSORG	dataset organization
RECFM	record format
LRECL	record length
BLKSZ	block size
EXTN	extents
USETRK	used tracks
CREATED	creation date
VOLUME	volume serial

STARTREC(*rec*) 1- to 8-digit starting record number. STARTREC can be abbreviated SREC.

RECORDS(*count*) The maximum number of records to print. (Default: installation option.) 1 to 8 digits.

Examples

```
LISTVOL FUNC(VTOC)      - /* print volume dataset list */
VOL(volser)
```

```
LISTVOL FUNC(VTOC) -
OPTION(LIST) -
FORMAT(FORMAT) -
COPYBOOK(CBKVTOC:SYSTEM) -
SORTFLD(DATASET) -
STARTREC(25) -
RECORDS(10) -
VOL(VEND02 )
```

```
LISTVOL FUNC(SPACE)    - /* print volume free space */
VOL(volser)
```

```
LISTVOL FUNC(SPACE)    - /* print volume free space */
VOL(SYS*)              /* for volumes starting with SYS */
```

LMODMAP – List Load Module Statistics Function

LMODMAP prints load module statistics, formatted hex dumps for csects and reverts the csect from machine code to source code.

Command format

```
LMODMAP FUNCtion(ALL | INFO | HEXDUMP | REVERT) -  
  DataSet(loadlib dataset name) -  
  MEMber(member name) -  
  CSECT(csect within member) -  
  MEMber(next member) -  
  StartMEM(member) EndMEM(endmember)
```

Keywords

FUNCTION() Specifies the requested function.
FUNC()

ALL	List all report types
INFO	List module attributes only
HEXDUMP	List hexadecimal format for csect
REVERT	Revert csect machine code to source

DATASET() 1 - 44 character dataset name or symbolic. The dataset name may contain a member name
DS() enclosed in brackets

Example : DATASET(dataset(member))

MEMBER() Optional member name used as an alternative to specifying the member as part of the dataset
MEM() name keyword.

Example: MEMBER(member)

CSECT() Optional csect name within the load module. Omitting the csect results in all csects being
processed.

Example: CSECT(csect name)

STARTMEM() Optional 1 - 8 byte start member that exists on a partitioned dataset
SMEM()

ENDMEM() Optional 1 - 8 byte end member that exists on a partitioned dataset
EMEM()

Examples

```
LMODMAP FUNC(ALL ) - /* all report type */  
  DATASET( LOAD.LIBRARY ) -  
  MEMBER(FMVALDD) -  
    CSECT(FMVALDD) -  
    CSECT(MODIENX) -  
  MEMBER(FMVALOC) -
```

```
        CSECT(MODCMDT ) -  
        CSECT(MODALEX ) -  
MEMBER(FMVAMS ) -  
        CSECT(AMSGDG ) -  
        CSECT(AMSNONV ) -  
MEMBER(FMVAUTO ) -  
        CSECT(MODAULT ) -  
        CSECT(MODAULO ) -  
MEMBER(FMVBASC ) -  
        CSECT(BSCCTRL ) -  
MEMBER(FMVBASE )
```

```
LMODMAP FUNC(REVERT) -  
        DATASET( dataset.name(FMVALDD) )
```

```
LMODMAP FUNC(INFO ) -  
        DATASET( dataset.name ) - /* module stats for */  
        STARTMEM(FMVALDD ) - /* member group  
        ENDMEM(FMVAUTO )
```

MANUAL - Print, delete or import manual function

MANUAL is a multi-function command which prints, deletes and imports manuals as well as lists manual directories and text members. The specific function is requested by means of the FUNCTION/FUNC keyword listed below supported by the required parameters for the function. There are three components to a manual i.e., the MANUAL, the TOPIC text and the HELP text. The MANUAL is a list of controls and text members which indicate to MANUAL the order, parts and text to be printed, deleted or listed. The TOPIC text is a member containing the text to be printed, deleted or listed. The HELP text is a member which can be used as part of a manual but is primarily used for online assistance.

Command format

MANUAL	FUNC(IMPORT)	-
	MANUAL(member) USERID(userid)	-
	DS(dataset.name)	-
	TITLE(description)	
MANUAL	FUNC(PRINT)	-
	MANUAL(member) USERID(userid)	
MANUAL	FUNC(PRINT)	-
	MANUAL(member) USERID(userid)	
MANUAL	FUNC(LISTDIR)	-
	MANUAL(member) USERID(userid)	-
	TOPIC(member) USERID(userid)	-
	HELP(member) USERID(userid)	-
	GENERIC	
MANUAL	FUNC(LIST)	-
	MANUAL(member) USERID(userid)	-
	TOPIC(member) USERID(userid)	-
	HELP(member) USERID(userid)	-
	GENERIC	
MANUAL	FUNC(DELETE)	-
	MANUAL(member) USERID(userid)	-
	TOPIC(member) USERID(userid)	-
	HELP(member) USERID(userid)	-
	GENERIC ALL	

Keywords

FUNCTION(DELETE)	Specifies a request to delete MANUAL, TOPIC or HELP members
FUNC()	.
MANUAL()	Optional 1- to 8-character MANUAL member.
or	
TOPIC()	Optional 1- to 8-character TOPIC text member.
or	
HELP()	Optional 1- to 8-character HELP text member.
GENERIC	Optional indicator for generic member groups appended to the member names for
*	MANUAL, TOPIC and HELP keywords. See the examples.

ALL Optional indicator to DELETE all topics associated with the MANUAL.

----- Function IMPORT -----

FUNCTION(IMPORT) Specifies a request to import MANUAL, TOPIC, or HELP members.
FUNC()

DATASET() Required 1 - 44 character partitioned dataset name. Do not specify the member name but
DS() rather use the MANUAL, TOPIC or HELP keywords.

MANUAL() Optional 1- to 8-character MANUAL member. Saves the TOPIC and HELP members
specified in the controls and creates a MANUAL member.

or
TOPIC() Optional 1- to 8-character TOPIC text member. Saves the manual members specified in the
controls but does not create a MANUAL member.

or
HELP() Optional 1- to 8-character HELP text member. Saves the HELP members specified in the
controls but does not create a MANUAL member.

AUTH() Optional 1 - 8 function authorization used to control security access.

TITLE() Optional 1 - 50 character MANUAL description used for identification.

----- Function LIST -----

FUNCTION(LIST) Specifies a request to list MANUAL, TOPIC or HELP members.
FUNC()

MANUAL() Optional 1- to 8-character MANUAL member.

or
TOPIC() Optional 1- to 8-character TOPIC text member.

or
HELP() Optional 1- to 8-character HELP text member.

GENERIC Optional indicator for generic member groups appended to the member names for *
MANUAL, TOPIC and HELP keywords. See the examples.

----- Function LISTDIR -----

FUNCTION(LISTDIR) Specifies a request to list the directory of available manuals
FUNC()

FILE() Optional 1- to 8-character preallocated ddname as the target for the report
F() Default: reports file

MANUAL() Optional 1- to 8-character manual generic MANUAL member name.

GENERIC Optional indicator for generic member groups appended to the member names for the
MANUAL keyword. See the examples.

----- Function PRINT -----

FUNCTION(PRINT) Specifies a request to print a manual.
FUNC()

FILE() Optional 1- to 8-character preallocated ddname as the target for the report.
F() Default: reports file

MANUAL() Required 1- to 8-character manual MANUAL name. Use FUNC(LISTDIR) to determine the manuals available.

TOPIC() TOPIC is a keyword used interchangeably with the keyword MANUAL.

Examples

```
/*  IMPORT MANUALS  */

MANUAL  FUNC(IMPORT)  MANUAL(FMINST)  -
          DS(PDS.TEXT.MANUALS)        -
          TITLE(FMINST: FM Installation Manual)
MANUAL  FUNC(IMPORT)  MANUAL(FMUSER)  -
          DS(PDS.TEXT.MANUALS)        -
          TITLE(FMUSER: FM User Manual)

/*  PRINT MANUALS  */

MANUAL  FUNC(PRINT)  MANUAL(FMINST)
MANUAL  FUNC(PRINT)  TOPIC(FMUSER)

/*  LIST DIRECTORY  */

MANUAL  FUNC(LISTDIR)
MANUAL  FUNC(LISTDIR)  MANUAL(FMU*)

/*  LIST TOPIC MEMBER  */

MANUAL  FUNC(LIST)  MANUAL(FMUSER)
MANUAL  FUNC(LIST)  TOPIC(FMU*)
MANUAL  FUNC(LIST)  HELP(FMU*)

/*  DELETE MEMBER(S)  */

MANUAL  FUNC(DELETE)  MANUAL(FMINST)
MANUAL  FUNC(DELETE)  TOPIC(FMC*)
MANUAL  FUNC(DELETE)  HELP(FMC*)
```

PRINTDS - Print Dataset Function

Dataset printing is done via a batch job. The parameters and submission can be handled automatically by a panel (see Section 8.3 Print File Entry Panel) or entered and submitted manually.

PRINTDS, the FileMarvel batch print utility, formats and prints entire or selective records of a dataset in hexadecimal, character, block, or copybook format modes. The PRINTDS command processes VSAM, ISAM, tape, PDS, keyed and non-keyed sequential as well as special format datasets like PDS directories, volume VTOCs and VVDSs. Records are retrieved by sequence, key, relative record or RBA. Records are printed by key ranges or Boolean selection criteria. Entire partitioned datasets or member ranges are searched with selection arguments. Records are printed in one of several optional formats: asis, standard, ISPF hex format, hex dump format, reformat and hex reformat templates and copybook format and compact template formats.

Once the appropriate copybook template is compiled (either on-line in FileMarvel or using the batch compilation procedure) use it to format dataset records. Extended copybook formats display the field name, level, length, offset, usage, converted field value, and actual value. The standard format is used to generate reports for the end user reducing the need for some report programs.

If a copybook template is not marked as global (GBL), a USERID statement must precede the PRINTDS statement. The syntax and usage of the USERID statement is found in Section B.2 - USERID Statement.

PRINTDS Statement

PRINTDS {File(*ddname*) | DataSet(*datasetname*)} -
[VOLume(*volser*)] -
[OFIle(*outddname*)] -
[COPIBOOK(*copybook* | *copybook:library*)] -
[REFORMAT(*template* | *template:library*)] -
[TITLE(*title*)] -
[STATS({ON | OFF})] -
[EJECT] -
[WIDTH({STandarD | MAXimum})] -
[KEYS({ON | OFF})] -
[FORMAT({Block | Default |Hex | KEYS | STandarD | CoMPact |
ForMaT | HEXDump | REFormat | HEXRef})] -
[FieldLOC(*startbyte*) [FieldLEN(*length*)] -
[RECORDS({count | ALL})] -
[StartREC(*rec*) [EndREC(*endrec*)] -
StartKEY(*key*) [EndKEY(*endkey*)] -
StartREL(*rrn*) [EndREL(*endrrn*)] -
StartRBA(*rba*) [EndRBA(*endrba*)] -
StartMEM(*member*) [EndMEM(*endmember*)] -
[SElect(*criteria*) [{AND|OR} SElect(*criteria*)...]

Operands

FILE(*ddname*) Pre-allocated 1- to 8-character DD name specifying the input dataset to format. FILE can be abbreviated F.

DATASET(*datasetname*) 1- to 44-character input dataset name to format. DATASET can be abbreviated DS.

VOLUME(*volser*) 1- to 6-character dataset volume serial name. Required if the dataset is not cataloged. VOLUME can be abbreviated VOL.

OFIle(*outddname*) 1- to 8-character output file DD name containing the formatted results. The output file can be a SYSOUT file or a dataset. If OFIle is not specified, the formatted record is set to the REPORTS DD name in the JCL procedure. (Default: JCL procedure report file.)

FORMAT	Report format type. (Default: DEFAULT. Under DEFAULT, formatting depends on dataset type, as explained below.)
BLOCK	Print the entire physical block in hexadecimal mode. Useful to validate the physical contents of the block. Can be abbreviated B.
DEFAULT	Format is determined by dataset type. Data files are formatted in hexadecimal. PDS directories are formatted by member. Can be abbreviated D.
HEX	Format the record in an ISPF hexadecimal display. Can be abbreviated H.
KEYS	Do not format the record, but print the keys of the record only.
STANDARD	Format the record in a standard display, translating all non-display data to periods. Can be abbreviated STD.
COMPACT	Format the record in the compact field definition format. Valid only if COPYBOOK is specified. Can be abbreviated CMP.
FORMAT	Format the record in the formatted field definition format, with each field specifying the level, offset, length, usage and value. Valid only if COPYBOOK is specified. Can be abbreviated FMT.
HEXDUMP	Format the record in a hexadecimal dump. Can be abbreviated HEXD.
REFORMAT	Format the records according to the template specified in the REFORMAT() keyword. Can be abbreviated as REF.
HEXREF	Format the records according to the template specified in the REFORMAT() keyword in ISPF hexadecimal format. Can be abbreviated as HEXR.

COPYBOOK parameters

FORMAT(FORMAT/COMPACT) Compare the records according to the template specified in the COPYBOOK() keyword

COPYBOOK(*copybook*) Name of a compiled copybook template. Required for FORMAT, COMPACT and for formatted SELECT statements. Add the library name if a SIGNON statement is not used. Example: COPYBOOK(COBBOOK:TSO21)

REFORMAT parameters

FORMAT(REFORMAT/HEXREF) Format the records according to the template specified in the REFORMAT() keyword. Can be abbreviated as REF

REFORMAT(*template*) Name of a compiled reformat template. Add the library if the SIGNON statement is not used. Required for REFORMAT formats. Example: REFORMAT(REFBOOK:TSO21)

TITLE(*title*) Text string to print in the heading of report pages.

STATS Determines if dataset statistics are printed at the end of the PRINTDS output. (Default: ON.)

ON
OFF

Display file statistics.
Do not display file statistics.

EJECT

Eject the print page prior to formatting a new print record.

WIDTH	Output format column width. WIDTH is ignored with special formatting. (Default: MAXIMUM.)
STANDARD	Truncate the format to an 80-character image. Can be abbreviated STD.
MAXIMUM	Format the output record to the full width of the page. Can be abbreviated MAX.
KEYS	Determines if the record key is displayed. (Default: ON.)
ON	Display record keys.
OFF	Do not display record keys.
FIELDLOC(<i>startbyte</i>)	Starting point within the record to print. (Default: start of record.) FIELDLOC can be abbreviated FLOC.
FIELDLEN(<i>length</i>)	Number of bytes to print. (Default: end of record.) FIELDLEN can be abbreviated FLEN.
RECORDS(<i>count</i>)	The maximum number of records to print. (Default: installation option.) 1 to 8 digits.
ALL	Print all records in the file.

Record positioning keywords consist of the pairs STARTREC-ENDREC, STARTKEY-ENDKEY, STARTRBA-ENDRBA, STARTTTR-ENDTTR, STARTABS-ENDABS, or (for PDSs only) STARTMEM-ENDMEN. Each of these pairs is mutually exclusive from the other pairs. In each case, the END of the pair is optional. The default is the end of the dataset.

STARTREC(*rec*) 1- to 8-digit starting record number. STARTREC can be abbreviated SREC.

ENDREC(*endrec*) 1- to 8-digit ending record number. ENDREC can be abbreviated EREC.

STARTKEY(*key*) 1- to 64-byte starting key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes, and/or any combination. Exact or generic keys are located depending on the length of the key. STARTKEY can be abbreviated SKEY.

Where *key*:

B'value'	Binary value.
C'text'	Character string.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode

ENDKEY(*endkey*) 1- to 64-byte ending key, for keyed datasets only. The key can be specified in character, hexadecimal, packed, binary modes and/or any combination. Exact or generic keys are located depending on the length of the key. ENDKEY can be abbreviated EKEY.

Where *key*:

B'value'	Binary value.
C'text'	Character string.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode

STARTRBA(*rba*) Starting hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-than conditions are not applicable to RBA processing.) STARTRBA can be abbreviated SRBA.

ENDRBA(*endrba*) Ending hexadecimal RBA address, for VSAM files only. The RBA address must be specified in the format X'xxxxxxxx' and must actually exist in the file. (Generic or greater-than conditions are not applicable to RBA processing.) ENDRBA can be abbreviated ERBA.

STARTREL(*rrn*) 1- to 8-digit starting relative record number. STARTREL can be abbreviated SREL.

ENDREL(*rrn*) 1- to 8-digit ending relative record number. ENDREL can be abbreviated EREL.

STARTMEM(*member*) Starting member, for partitioned datasets only. The member specified must actually exist in the PDS. STARTMEN can be abbreviated SMEN.

ENDMEM(*endmember*) Ending member, for partitioned datasets only. The member specified must actually exist in the PDS. ENDMEN can be abbreviated EMEM.

SELECT Statement

Select records based on their contents using one or more SELECT statements. If more than one SELECT statement is specified, they must be separated by AND or OR keywords.

Format:

```
SELECT(fieldname cond value)
      [{AND|OR}
SELECT ...
```

where:

recordtyp The 01 COBOL level or a PL/I structure included in the copybook template statement.

fieldname The field containing a value used to determine which record layout to use.

cond The condition to compare the data in the dataset represented by *fieldname* to a value to determine which recordtype to use for this record.

EQ	Equals.
NE	Does not equal.
LE	Is less than or equal to.
LT	Is less than.
GE	Is greater than or equal to.
GT	Is greater than.

value The value to compare against *fieldname*.

B'value'	Binary value.
C'text'	Character string.
D'number'	Packed decimal number.
Q'value'	Bit value.
X'value'	Hexadecimal value.
string	Character string without imbedded blanks
'string'	Character string with imbedded blanks
X'hex'char	Mixed hex and character mode
HIGHVALUE	Reserved keyword indicating field filled with X'FF'.
LOWVALUE	Reserved keyword indicating field filled with X'00'

AND|OR Logical operators to construct a compound condition specifying multiple *fieldnames* and conditions.

Sequential dataset

```
PRINTDS DS(dataset) -
COPYBOOK(copybook) -
SELECT(M-CRT-DATE EQ +811203) AND -
SELECT(H-DATE-DUE EQ +811117) OR -
SELECT(M-CRT-DATE EQ +820128) AND -
SELECT(H-DATE-DUE EQ +820220) -
RECORDS(10)
```

SIGNON Statement

If a copybook template is not marked as global (GBL), the SIGNON statement must precede the PRINTDS statement if copybook or reformat templates are used.

SIGNON USERID(*userid*)

where:

userid The userid for the copybook creator.

Example:

```
SIGNON USERID(TSO21)
PRINTDS DS(ACCOUNTS.PAY02.VSAM.JAN) -
COPYBOOK($MYCBKCOB) -
FORMAT(FORMAT) -
STARTKEY(X'00'CONTROL '0I') -
RECORDS(1)
```

PRINTDS Control Statement Examples

Examples:

Sequential dataset

```
PRINTDS DS(dataset) -
COPYBOOK(copybook) -
SELECT(M-CRT-DATE    EQ +811203)    AND -
SELECT(H-DATE-DUE    EQ +811117)    OR -
SELECT(M-CRT-DATE    EQ +820128)    AND -
SELECT(H-DATE-DUE    EQ +820220) -
RECORDS(10)
```

VSAM dataset

```
PRINTDS DS(dataset) -
COPYBOOK(copybook) -
FORMAT(FORMAT) -
STARTKEY(X'00'C'CONTROL 'X'0I') -
RECORDS(1)
```

Unformatted sequential dataset

```
PRINTDS DS(dataset) -
RECORDS(10) -
FIELD(1) -
FIELDLEN(50) -
FORMAT(HEX) WIDTH(MAXIMUM) -
SELECT(X'00'CONTROL.P'025C' EQ 1 50) OR -
SELECT(CONTROL EQ 50 100)            AND -
SELECT(P'025C' GT 100)
```

Unformatted sequential dataset

```
PRINTDS DS(dataset) -
RECORDS(10) -
FORMAT(HEX) WIDTH(STANDARD)
```

VSAM KSDS dataset

```
PRINTDS DS(dataset) -
OFILE(PRNTFILE) -
STARTKEY(X'00'C'CONTROL 'X'0I') -
ENDKEY(X'01'C'EXECUTE') -
```

FORMAT(STANDARD) WIDTH(MAXIMUM)

VSAM ESDS dataset

PRINTDS DS(dataset) -
FORMAT(STANDARD) -
WIDTH(STANDARD) -
FIELDLOC(51) -
FIELDLEN(50)

VSAM RRDS dataset

PRINTDS DS(dataset) -
STARTREC(15) -
FIELDLOC(44) -
FIELDLEN(64) -
RECORDS(05) -
FORMAT(HEX) -
WIDTH(MAXIMUM)

Selected members of a PDS

PRINTDS DS(dataset(*)) -
STARTMEM(meml) -
ENDMEM(memn) -
RECORDS(30) -
SELECT(CONTROL EQ 1 50) OR -
SELECT(ARGUMENT)

Particular member of a PDS

PRINTDS DS(dataset(member)) -
RECORDS(30) -
STATS(OFF)

Load library PDS directory record

PRINTDS DS(dataset) -
STARTMEM(meml) -
ENDMEM(memn) -
RECORDS(30) -
STATS(OFF)

SORTDS – Sort Dataset Function

SORTDS reorganizes records of a file into desired ascending or descending field sequences. Files may be sorted into new target files or originating source files. The SORT keywords and operands are based on the standard sort utility currently installed in your data center in an effort to eliminate confusion. Many of these keywords, however, need not be used since the SORT command generates them internally.

Copybook template fields may also be used as the sort fields eliminating the need to calculate field displacements, lengths and types. The copybooks must be of course compiled and linked prior to using this technique. Refer to the COMPILE copybook command for the process.

On completion of a successful sort, no messages are displayed unless the SYSOUT ddname is present. On unsuccessful sorts, error messages will be displayed on the internally allocated SYSOUT ddname.

Keywords

DATASET() DS()	Required 1 - 44 character dataset name of the input file to be sorted.
ODATASET() ODS()	Optional 1 - 44 character dataset name of the target file containing the result of the sort. If the ODS() keyword is omitted, the target of the sort defaults to the input file DS().
FIELDS() FIELD()	Required set or sets of sort sequence fields ordered in the same format as the batch utility sort format. Each set must contain the field start, field length, field type and sort direction with each field separated by a comma.
field start	Required 1 - 5 numeric relative to 1 indicating the sort starting location. If the file type is variable, the 4 byte rdw must be accounted for in the field start.
field length	Required 1 - 5 numeric indicating the length of the sort field.
field type	Required 2 characters indicating the type of the sort field.
BI	field type is binary
CH	field type is character
PD	field type is packed decimal
ZD	field type is zoned decimal
direction	Required 1 character indicating the direction of the sort for the field.
A	Direction is ascending
D	Direction is descending
	Example: FIELDS(1,5,CH,A,25,3,PD,D)
LENGTH() LEN()	Optional 1 - 5 numeric indicating the file record length. If the file type is variable, the maximum record length including the rdw must be specified. If this keyword is omitted, the value will be set internally.
SIZE()	Optional 1 - 8 numeric indicating the estimated number of records to be sorted. If this keyword is omitted, the value will be set internally.
TYPE()	Optional 1 character indicating the file type. If this keyword is omitted, the value will be set internally.
F	file type is fixed
V	file type is variable

WORK() Optional 1 - 5 numeric indicating the number of sort work files to be allocated. If this keyword is omitted, the value will be set internally.

----- Copybook Template Keywords -----

COPYBOOK()
CB() Optional 1- to 8-character member of a compiled and linked copybook template. The copybook sort field names are extracted from this member.

FIELDS() If the COPYBOOK keyword is specified the sort fields may be specified by copybook data names and sort directions with each field separated by a comma or blank. Field lengths and types will be extracted from the copybook.

field start Required copybook field name

direction Required 1 character indicating the direction of the sort for the field.

A Direction is ascending
D Direction is descending

Example: FIELDS(REC-KEY,D -
ITEM-NUMBER,A -
STOCK-NUMBER,D)

Examples

```
/*          */
/* standard sort */
/*          */
SORTDS DS(sequential.dataset) -
      ODS(target.dataset) -
      FIELDS(1,5,CH,A,10,2,PD,D)

/*          */
/* variable sort into source dataset */
/*          */
SORTDS DS(sequential.dataset) -
      FIELDS(5,5,CH,A,15,2,PD,D)

/*          */
/* copybook sort */
/*          */
SORTDS DS(sequential.dataset) -
      ODS(target.dataset) -
      COPYBOOK(STOCKORD) -
      FIELDS(REC-KEY,D -
            ITEM-NUMBER,A -
            STOCK-NUMBER,D)
```

SPACE – Print Volume Free Space Function

SPACE is a multi-function command which either prints direct access free space statistics or formats the free space on behalf of another function such as the LSPACE command. Specific, generic or DASD mount volume types may be used to generate selective reports.

If the OFFILE keyword is not used, the report will be directed to the report file of the JCL procedure.

Command format

```
SPACE FUNC(PRINT)          -
    VOL(volser)            -
    UNIT(unit address)     -
    UCB(in core address)   -
    OFFILE(report ddname)  -
    TITLE(title heading)   -
    NOTFOUND label        -
    ONERROR label

SPACE FUNC(FORMAT)         -
    VOL(volser)            -
    UNIT(unit address)     -
    UCB(in core address)   -
    NOTFOUND label        -
    ONERROR label
```

Keywords

FUNCTION()	Perform the function requested by the operand.
FUNC()	
PRINT	Print the space report depending on the value of VOLUME, UNIT or UCB keyword.
FORMAT	Format the unit passed by the LSPACE command depending on the UCB address of the UCB keyword.
OFFILE()	Optional 1- to 8-character output file ddname which will contain the formatted output. The output file may be a sysout file or a dataset.
VOLUME()	Required volume, generic volume or volume types
VOL()	
volser	1- to 6-character volume or generic volume
PRIVATE	display all private mounted volumes
PUBLIC	display all public mounted volumes
STORAGE	display all storage mounted volumes
ALL/*	all volumes
or	
UNIT()	A 1 - 3 character unit address or symbolic.
or	
UCB()	An absolute ucb address or symbolic containing the address.
TITLE()	A heading set in either the report file or the OFFILE ddname.
NOTFOUND label	On a not found condition, branch to label
ONERROR label	On an error condition, branch to label

Symbolics

&ZSPCTCY	Decimal value specifying the total free cylinders available
&ZSPCTTK	Decimal value specifying the total free tracks available
&ZSPCTEX	Decimal value specifying the total free extents available
&ZSPCCCY	Decimal value specifying the total free contiguous cylinders available
&ZSPCCTK	Decimal value specifying the total free contiguous tracks available
&ZVOL	Current volume serial number
&ZUNIT	Current unit name
&ZUCLASS	Current unit device class
&ZUMOUNT	Current unit mount status
&ZUOPEN	Current unit open count
&ZURESrv	Current unit reserve count
&ZUCB	Current unit in core address
&ZLSPACE	Current unit lspace value
&ZLINE	Current unit formatted space line

Examples

```
/*          */
/* display free space */
/*          */
SPACE VOL(*)
SPACE VOL(ALL)
SPACE VOL(STG)
SPACE VOL(STG110)
SPACE VOL(PRIVATE)

/*          */
/* direct free space to a file */
/*          */
ALLOC F(PRNTFILE) SYSOUT(5) -
      RECFM(FBA) LRECL(133) BLKSIZE(1310)
OPEN F(PRNTFILE) OUTPUT
SPACE VOL(PUBLIC) OFILE(PRNTFILE) -
      TITLE(All public volumes report)
CLOSE F(PRNTFILE)
FREE F(PRNTFILE)

/*-----*/
/* FORMAT DASD DEVICE BASED ON PASSED &GDEVVOL */
/* PARAMETER USING THE UCBS CAN FUNCTION. */
/*-----*/
```

SPACE-INIT:

```
UCBS CAN FUNC(INITIA TE) DEVTYPE(DASD) ONLINE
```

SPACE-SCAN:

```
UCBS CAN FUNC(SCAN) ATEND GOTO SPACE-RETURN
SET &GUNIT &ZUNIT
SET &GUCB &ZUCB
SET &GVOL &ZVOL
IF &GDEVVOL = &ZNULL -
  OR = * -
  OR = ALL -
  OR = &ZVOL TRUNC -
```

```
OR = &ZUMOUNT -
THEN GOTO SPACE-LSPACE
IF &GDEVMNT = &ZUMOUNT GOTO SPACE-LSPACE
GOTO SPACE-SCAN
/*-----*/
/* ISSUE LSPACE AND FORMAT THE RESULTS */
/*-----*/
```

SPACE-LSPACE:

```
LSPACE UCB(&GUCB) ONERR GOTO SPACE-ERROR
SPACE FUNC(FORMAT) UCB(&GUCB) OFILE(&GPRINT)
FLOW OFF
GOTO SPACE-SCAN
```

```
/*-----*/
/* RETURN */
/*-----*/
```

SPACE-RETURN:

```
RETURN
```

Batch Execution Language

Batch Execution Language is a language which performs specialized computer functions at a higher level than the standard assemblers and compilers associated with the operating system. BASIC was designed to assist the individual in accomplishing complicated and time consuming tasks without the need to know system or compiler language requirements.

Functional commands and associated operands are processed by BASIC and in turn generate the required code to execute the request. Certain commands as the PRINTDS command generate lower level in stream commands in order to accomplish the higher level command request. Multiple commands may be coded into a logical order and executed in stream or compiled into executable procedures, saved and recalled for execution by other procedures.

A wide range of commands are available with BASIC which will assist in simplifying the demands of data processing. BASIC contains a variety of features which have been developed to simplify and reduce the coding effort in order to accomplish on going as well as one time functions.

Consider the following features:

A simple high-level language permitting the desired results without the need to know the internals required to accomplish tasks.

The capability to perform functions in an easy fashion as well as perform one time functions without having to write extensive programs to accomplish the same results.

Print DASD and tape files in hex, character, COBOL and PLI copybook modes. Entire files or selected records can be formatted by means of random key selection or by Boolean logic selection criteria.

Entire PDS datasets as well as member ranges may be searched for with selective arguments.

Print DASD PDS directories, VTOCs and datasets in formatted as well as hex formats.

Validate file records according to copybook templates and display non-conforming data isolating potential violations and production batch abends.

A full range of commands to access, modify and manipulate VSAM, ISAM, PDS, DIRECT, SEQUENTIAL files without the need to know the internals of each file type.

A facility to modify, generate, insert, reduce, delete, expand and manipulate records and fields within records at your discretion.

The capability to sort files with standard field formats as well as copybook data names eliminating the need to calculate field displacements, lengths or types.

A full range of program control commands to execute and manage programs.

Format desired core addresses in hex by means of table scans or pointer tracking for debugging purposes.

Execution JCL

Execution is accomplished by executing a JCL procedure called FMVBATCH, which resides in a system JCL procedure library. Commands are entered as an input stream via the SYSIN or FMVSYIN ddnames. The FMVSYIN ddname must be used for the input procedure stream only when the SYSIN ddname is required by a function within the procedure. If both SYSIN and FMVSYIN are omitted, the system will generate a SYSIN ddname for the job.

During the installation process, system option defaults are generated but may be overridden using the PARM on the execute statement. The options control the compile and execution of the input stream. Refer to the section EXECUTION PARAMETERS in this manual for the definition options.

The FileMarvel JCL procedure may have been tailored during installation to conform to system standards and may not appear as listed below. However, a sample of the JCL procedure, ddname specifications and execution examples are listed below.

```
//FMVBATCH PROC RGN=4096K,      /* Region size          */
//  FLQ=FMARVEL,              /* Installation prefix   */
//  SLQ=V?R?,                 /* Installation level    */
//  LLIB=LLIB,                /* Load module library  */
//  REPORT='*',               /* Function report      */
//  PRINT='*',                 /* Stream echo sysout   */
//  DUMP='*'                   /* Dump sysout          */
/*-----*/
/*  FileMarvel BATCH EXECUTION PROCEDURE */
/*-----*/
//FMVBATCH EXEC PGM=FMV?BTCH,REGION=&RGN
//STEPLIB DD DSN=&FLQ.&SLQ.&LLIB,DISP=SHR
//FMVPRINT DD SYSOUT=&PRINT
//FMVREPTS DD SYSOUT=&REPORT
//SYSUDUMP DD SYSOUT=&DUMP
//  PEND
```

DDNAME	DESCRIPTION
STEPLIB	resident load module library
FMVPRINT	command echo, compile and execution sysout
FMVREPTS	default command execution report sysout
FMVSYISIN	command input stream sysin or (SYSIN)
SYSUDUMP	abnormal end snap dump sysin
SYSIN	command input stream sysin or (FMVSYISIN)

Note: Since FileMarvel treats a "/" as a comment, entry in position 1 of the FMVSYISIN input will cause an end of text condition by the operating system. If comments are coded in position 1, then the FMVSYISIN ddname should be coded as follows:

```
//FMVSYISIN DD DATA,DLM='##'
```

Examples:

```
//JOB CARD
//PRINTDS EXEC FMVBATCH

/*          */
/* print files */
/*          */
PRINTDS DS(test.seq) /* seq file */

PRINTDS DS(vsam.file) - /* vsam file */
STARTKEY(X'00',C'KEY'X'01') -
FORMAT(HEX) -
WIDTH(STANDARD)

//JOB CARD
//FMVBATCH EXEC FMVBATCH,PARM='NOLINK,EXEC'
//FMVSYISIN DD DATA,DLM='##'

/*          */
/* allocate sysin */
/*          */
ALLOC F(SYSIN) DS(TEST.FILE) SHR
```

```

/*          */
/* print seq files */
/*          */
PRINTDS DS(test.seq)

/*          */
/* print pds files */
/*          */
PRINTDS DS(pds.dir)

/*          */
/* print vsam files */
/*          */
PRINTDS DS(vsam.file) -
COPYBOOK(member) -
ENDKEY(X'00',C'KEY'X'01') -
RECS(25) -
FIELDLOC(16) -
FIELDLEN(32) -
FORMAT(HEX) -
WIDTH(MAXIMUM)

```

Execution JCL parms

The compile and execution of procedures are controlled by options set either by the installation options, parameter overrides on the execute statement of the JCL or by the COMPILE command within the input stream. System default options may be overridden by inserting the proper parameter in the PARM keyword of the BASIC JCL. For example, BASIC may have been installed without the copybook compile cross reference report and one may be desired. By merely inserting the PARM='CBXREF' on the execute statement, the report will be generated as a part of the compile process.

The parameters are described in detail below.

Parameter

SOURCE	Generates a compile echo of the input stream and the source print of any members copied in a Copybook compile request. SOURCE is usually the default.
NOSOURCE	Do not produce a compiled source report.
CBXREF	Generates a copybook compile report which displays the copybook fields, levels, lengths, offsets, usage, and initial values. This report may be used to validate the copybook template prior to the actual file record formatting.
NOCBXREF	Do not produce a copybook cross reference report.
LINK	This parameter applies to compiled procedures or copybooks using the COMPILE command. On successful completion of a compiled member, the generated object will be linked in the BASIC procedure load library. Linked procedures may be called by other in stream commands or procedures. Linked copybooks may be used to format dataset records.
NOLINK	Do not link the compiled procedure or copybook. NOLINK may be used for the first pass of a compile and execution to obtain the proper results prior to overlaying existing members.
LTORG	Generates a compile report which displays the values, symbolics and keyword operands with lengths and displacements. This report may be used to validate data or symbolics used prior to the actual execution of the commands.

NOLTORG Do not produce a compiled ltorg report.

SAVE This parameter applies to compiled procedures or copybooks using the COMPILE command. On successful completion of a compiled member, the in stream source applicable to the member will be saved in the BASIC source library.

NOSAVE Do not save the compiled source.

XREF
for Generates a compile report which displays the commands, keywords and keyword operands ready for execution. This report may be used to validate expected execution prior to the actual execution of the commands.

NOXREF Do not produce a compiled cross reference report.

----- Execution Time Parameters -----

EXEC On completion of a single or multiple member compile and/or link, execute the in stream or compiled procedures. EXEC is usually the default.

NOEXEC Do not execute the compiled procedure. NOEXEC may be used to compile and link a series of procedures later to be executed by other procedures.

FLOW Generates an execution report similar to the XREF report. During execution commands, the command is printed with resolved values and symbolics. The FLOW report is handy in debugging procedures. Caution should be observed for the report may be very large depending on tight code loops.

NOFLOW Do not generate an execution report. NOFLOW is usually the default.

IGNORERR On execution errors, continue processing with the next command.

LCASE Print all reports in lower case. LCASE is usually the default.

NOLCASE Print all reports in upper case. NOLCASE should be used when printers do not support lower case trains.

POOL
procedure Generates an execution symbolic resolution report. At the conclusion of the execution of a the symbolic pools and final values are printed. The POOL report is handy for debugging purposes.

NOPOOL Do not generate a symbolic report. NOPOOL is usually the default.

CALL() A 1- to 8-character member name of a compiled and linked procedure. A userid may be appended to the procedure name to execute a specific user proc by separating the proc and the userid with a colon. Ex. proc:userid. The member specified in the CALL parameter is executed immediately without the need of in-stream commands or procedures.

Examples

```
//JOB CARD
//FMVBATCH EXEC FMVBATCH

/*                      */
/* in stream commands are executed */
/*                      */
```

```

PRINTDS DS(vsam.file) -
      COPYBOOK(book1) -
      STARTKEY(X'00',C'KEY'X'01')

```

```

//JOB CARD
//FMVBATCH EXEC FMVBATCH,PARM='NOLINK,CBXREF'

```

```

/*                                     */
/* copybooks are compiled, not linked and xref is produced */
/*                                     */

```

```

COMPILE FUNC(COPYBOOK) -
      MEMBER(book1) -
      DESC(copybook compilation)
COPYLIB DS(TEST.COPYLIB)
COPY MEM(COBBMEM1)
MEND

```

```

COMPILE FUNC(COPYBOOK) -
      MEMBER(book2) -
      DESC(copybook compilation)
COPYLIB DS(TEST.COPYLIB)
COPY MEM(COBBMEM2)
MEND

```

```

//JOB CARD
//FMVBATCH EXEC FMVBATCH,PARM='XREF,LTORG,FLOW,POOL'

```

```

/*                                     */
/* procedures compiled, linked and executed with */
/* various reports generated */
/*                                     */

```

```

COMPILE MEMBER(proc1) -
      OPTIONS(XREF LINK EXEC FLOW) -
      DESC(this is a test case)
ALLOC F(*) DS(TEST.DATASET) SHR
CALL PGM(pgm1)
CALL PROC(proc1)
FREE F(*)

```

```

COMPILE MEMBER(proc2) -
      DESC(this is a test case)
ALLOC F(*) DS(TEST.DATASET) SHR
CALL PROC(proc1)
FREE F(*)

```

```

/*                                     */
/* execute a proc from the parm field */
/*                                     */

```

```

//JOB CARD
//FMVBATCH EXEC FMVBATCH,PARM='CALL(PROC:USERID),FLOW'

```

Command entry rules

Command entry may be in stream, compiled procedures and/or compiled copybooks. In stream commands and compiled procedures and copybooks may interspersed. Multiple compile members may be entered within the same

stream and are differentiated by means of the COMPILE command. In stream commands are executed on completion of successful validation and are assigned the member name of *SYSIN*.

Compiled procedures are a series of commands under the control of the COMPILE command and assigned the member name specified in the COMPILE command. These members may be linked and called by other in stream commands and procedures. If in stream commands and compiled procedures are to be interspersed, the MEND command must be used to indicate the end of the compile. The next command beyond the MEND command reverts back to the *SYSIN* member.

Compiled copybooks follow the same rules as the compiled procedures. These compiles are non-executable. The results of the compile are used by other commands such as the PRINTDS command.

Requests are entered as commands, keywords and keyword operands via the ddname FMVSYISIN. Commands are free form entries on one or more lines beginning in column 2 through 71. Command continuations must contain a hyphen at the point of the continuation followed by the continuation operand on the next line beginning in at least column 2. Comments are indicated by means of a "/*" followed by the comment and ending with a "*/" to close it out. Another form of a comment is an asterisk in column one. Command development components are described below.

Component

command 1- to 8-character verb specifying the desired function.

Example: PRINTDS

keyword
processed. 1- to 8-character command keyword indicating to the command that special operands must be

Example: MANUAL FUNCTION()

keyword operand, 1 to nn characters prefixed by the keyword and enclosed by left and right parenthesis indicating to the command that parameters are being passed by means of the keyword.

Example: PRINTDS DATASET(dataset.name)

hyphen indicates to the command continuation keywords associated with the command follow on the next card.

Example: PRINTDS -
 DATASET(dataset.name)

/* indicates to the command that the data following the "/*" should be treated as comments. Comments may be set as a unique line or to the right of a command or operand.

Example: PRINTDS - /* this is a test */
 DATASET(dataset.name)

*/ indicates to the command that the comment is being terminated.

Example: /* this is a test */
 PRINTDS -
 DATASET(dataset.name)

 PRINTDS - /* this is a test */
 DATASET(dataset.name)

* an asterisk in column 1 indicates the entire card is to be treated as a comment. This technique may be handy in commenting alternative keywords on a temporary basis.

Example: PRINTDS -
 * DATASET(dataset.one)
 * DATASET(dataset.two)
 DATASET(dataset.nnn)

completed Upon execution of each command, conditions are set to indicate whether the function was properly. Each command may be tested to determine what action to take on completion of the command. The common condition testing keywords are listed below.

Example: OPEN (SYSUT1) -
 ONERROR GOTO label

Examples

ONERROR ONERR	On an error condition, go to the specified label address.
ONNOERR NOERROR NOERR	On an a satisfactory completion, go to the specified label address
ONMSG MSG	On a satisfactory completion with a pending message ,go to the specified label.
WARNING WARN	On a warning condition, go to the specified label address.
ONEND ATEND	On an end condition, go to the specified label address. Used to read files.
NOTFOUND NTFD	On a not found condition, go to the specified label address. Used to locate members.
ONNULL NULL	On a null condition specified by a symbolic access, go to the specified address.

Examples

```

/*          */
/* print files */
/*          */
* PRINTDS DS(test.seq)      /* comment card */
PRINTDS DS(pds.file(*)) - /* fb pds file */
    STARTMEM(BCS) -
    ENDMEM(BLD) -
    RECORDS(30)
PRINTDS DS(vsam.file) - /* vsam file */
    STARTKEY(X'00',C'KEYX'01') -
    FORMAT(HEX) -
    WIDTH(NARROW)

/*          */
/* print vsam files */
/*          */
PRINTDS DS(vsam.file) -

```

```

*      ENDKEY(X'00',C'KEY'X'01') - /* comment card */
      RECS(25)      -
      FIELDLOC(16)  -
      WIDTH(WIDE)

```

Symbolic value rules

Symbolics are basically vehicles for communication. They are storage units set by commands or internally by FM and accessible to other commands and called procedures. Values, counters, core addresses or record buffer addresses may be stored, altered or accessed by commands.

Symbolics are identified as 1- to 8-character operands preceded by the character "&" i.e. &ZDATE. Symbolics may be set as operands or keyword operands and will be resolved prior to the execution of the command. For example, the symbolic &DSN in the DS keyword of the command PRINTDS DS(&DSN) will be resolved to the dataset name set by a previous command prior to executing the PRINTDS command. Each symbolic may contain any value length.

Locations within symbolic values may be accessed, inserted, extended or modified by means of (displacement, length) operands attached to a symbolic. For example, the fourth byte of a symbolic that happens to be the contents of a record read from a file or just being generated, requires alteration. Then the access of the fourth byte is accomplished by &SYMBOL(4,2) for two bytes. Data values may be moved to that location and the record replaced to the file. See the displacement operands below. Also see the SET and IF commands and RECORD GENERATION topics.

Symbolic pools are available in order to preserve the symbolic values depending on the requirements of commands or procedures. For example, shared data between procedures require GLOBAL pools in order for called procedures to set symbolics, retire and make the data available to calling procedures. Symbolic pool targeting is accomplished explicitly with special operands or implicitly by depending on the name of the symbolic.

Symbolics preceded with the character &Z are reserved symbolic names and are directed to the GLOBAL or SYSTEM pools. The reserved &Z symbolics should not be altered by the user due to system impact but may be accessed at any time.

Certain system symbolics &Z have special meanings or contain data available to commands automatically. For example, &ZDATE contains the current date and &ZTIME contains the current time and may be used by commands. The symbolic &ZNULL may be used to test a symbolic to determine if data is available. A list and description of special symbolics is provided in the topic SPECIAL SYMBOLICS.

Certain commands set &Z symbolics upon successful and execution pass data to subsequent commands. For example, the ALLOC command will pass the ddname in the symbolic &ZFILE. Other commands set symbolics to null after execution in order to indicate an unavailable resource. For example, the FREE command sets the symbolic &ZFILE to null to indicate the resource is unavailable. If applicable, each command description contains a list of symbolics and values which are set after execution.

Symbolics preceded with the characters &G are available for general use. However, these symbolics will be directed to the GLOBAL pools automatically without explicit direction.

Refer to the SET command for the details on setting symbolic values and to the IF command for testing symbolic values. Use of symbolics by commands depends basically on the symbolic characteristic and pool residency. A description of each is listed below.

Symbolics

BINARY	B''	The symbolic contains a full word of binary data only. Example: &BINDATA contains B'0AEB080F'
BIT	Q''	The symbolic contains bit data only. Example: &BINDATA contains Q'01010101'

ADDRESS	X'' The symbolic contains a 4 byte core address. Example: &ZCVT contains X'00000010'
BUFFER commands.	X'' The symbolic contains a 4-byte core buffer address making the buffer available to Example: &F1BUFF contains X'00F8C510'
DECIMAL	D'' The symbolic contains packed decimal numeric data only. Example: &COUNTER contains D'0000575C'
HEX	X'' The symbolic contains hexadecimal data only. Example: &HEXDATA contains X'0700ECD8'
NUMERIC	nnn The symbolic contains zoned decimal numeric data only. Example: &COUNTER contains 575
MIXED STRING	The symbolic contains mixed values described in the characteristics above. Example: &MIXED contains (Z'12345'C'ABCD'X'1234'B'1111'P'123')
VALUE	vvv The symbolic contains character data only. Example: &ZDATE contains mm/dd/yy
ZONED	Z'' The symbolic contains zoned decimal numeric data only. Example: &COUNTER contains Z'+0000575'

Operands

&symbol(d,l) Specific fields may be accessed within the value of the symbol by using displacements and lengths. Numeric, &symbols or keywords may be used to access the specific locations and lengths.

DISPLACEMENT(d) Specifies the displacement within a value of a symbol. A location beyond the value may result in a null value or an error message depending on its use. A displacement without a length operand indicates a length to the end of the value.

nnnn A numeric location relative to 1 may be used.

&symbol A symbol may be used containing the displacement.

PREFIX A keyword indicating the field one character prior to the first field of the value. Data may be prefixed using PREFIX keyword.

SUFFIX A keyword indicating the field one character beyond the last field of the value. Data may be appended using SUFFIX keyword.

LENGTH(l) Specifies the length of access for the value of a symbol. A length beyond the value will result in truncation.

nnnn A numeric length from the displacement.

&symbol A symbol may be used containing the length.

END A keyword indicating to the end of the value.

INSERT A keyword indicating a data insert is requested.

Examples: &ZREC(10,20)
 &ZREC(10,END)
 &ZREC(&LOCAL,&LENGTH)
 &ZREC(PREFIX,&LENGTH)
 &ZREC(SUFFIX,10)

symbol	operand	results
&symbol(offset) PREFIX		The symbol is set to the explicit offset value for the length of either the remainder of the record set to the value length of source data. Records are extended if source values extend beyond the end of the symbol.
value(&ZNULL)		If the SET data value is null, then the record is truncated from the offset to the end of the record. Example: &symb(50) = 'abcdef' &symb(50) = &ZNULL
&symbol(offset,length) PREFIX		The symbol is set to the explicit offset value for the specified length Records are extended if source values extend beyond the end of the symbol
value(&ZNULL)		If the SET data value is null, then the data at the offset is deleted for the length specified and the remainder of the data is shifted to the left. Example: &symb(PREFIX,25) = 'abcdef' &symb(50,6) = 'abcdef' &symb(50,25) = &ZNULL
&symbol(offset,INSERT)		Source value data is inserted at the explicit offset for the length of source data and the remainder of the data is shifted to the right of the data inserted. Records are elongated after the insertion is complete.
value(&ZNULL)		If the SET data value is null, then the data at the offset is deleted to the remainder of the data. Example: &symb(25,INSERT) = 'abcdef' &symb(50,INSERT) = &ZNULL
&symbol(PREFIX,INSERT)		Source value data is inserted at the start of the record and the entire record is shifted to the right. Records are elongated after the insertion is complete.
value(&ZNULL)		If the SET data value is null, then the symbol is set to null. Example: &symb(PREFIX,INSERT) = 'ab'

&symbol(offset,END) The symbol is set to the explicit offset value for the length of either the remainder of the record or set to the value length of source data. Records are not extended if the source extends beyond the end of the symbol and the source value is truncated.

PREFIX

value(&ZNULL) If the SET data value is null, then the data at the offset is deleted to the remainder of the data.

Example: &symb(25,END) = 'abcdef'
&symb(50,END) = &ZNULL

&symbol(SUFFIX,length) The symbol is appended with the value date for either the explicit length or SET with the source data value. Records are elongated after the data is appended.

value(&ZNULL) If the SET data value is null, then no action is taken.

Example: &symb(SUFFIX) = 'abcdef'
&symb(SUFFIX,5) = &record1

symbolic pool description

LOCAL Symbolics residing in the LOCAL pool are available to all in stream commands or the Currently executing procedure only. Other called procedures may not access the pool but rather are assigned their own LOCAL pool.

GLOBAL
called Symbolics residing in the GLOBAL pool are available to all in stream commands or all procedures. Both user and system symbolics may reside in the GLOBAL pool.

PROFILE Symbolics residing in the PROFILE pool are owned by a Signon and values are retained across signons.

SYSTEM Symbolics residing in the SYSTEM pool are owned by the system and may be shared by all signons.

OPTIONS Symbolics residing in the OPTIONS pool are owned by the system and contains the system Option settings. The OPTIONS pool may be accessed by all signons but not altered.

Examples

```

/* PRINT A DATASET */

SET &DSN = TEST.DATASET
PRINTDS DS(&DSN) -
STARTKEY(X'00'C'KEY'X'01')

/* A COUNTER TEST */

SET -
&M1 = (1.B'11110010'X'F3'C'4'Z'5'6.B'11110111'8)
SET &C3 = 3 POOL(LOCAL)
SET &COUNTER = &COUNTER+1

```

```

IF    &COUNTER > 5          GOTO    RETURN

/*  GENERATE FILE RECORDS  */

ALLOC DS(TEST.DATASET1) SHR
SET   &FILE1    &ZFILE
ALLOC DS(TEST.DATASET2) SHR
SET   &FILE2    &ZFILE
ALLOC DS(TEST.DATASET3)    -
      NEW      CATLG      -
      UNIT(DASD) SPACE(TRK 10 10) -
      LRECL(150) BLKSIZE(1500
SET   &FILE3    &ZFILE

OPEN  F(&FILE1)  INPUT
OPEN  F(&FILE2)  INPUT
OPEN  F(&FILE3)  OUTPUT

READ:
READ  F(&FILE1)  ATEND  GOTO    CLOSE
SET   &REC1     &ZREC
READ  F(&FILE2)  ATEND  GOTO    CLOSE
SET   &REC1     &ZREC
SET   &CTR1     &CTR1+1
SET   &REC3     -
      (&CTR1&REC1(10,25)&REC2(60,25).X'0000'.&REC1(300,END))
WRITE F(&FILE3)  REC(&REC3)
GOTO  READ

```

String value rules

Certain command keyword operands will accept character, hex, packed, binary, zoned decimal or mixed mode strings in order to represent the internal data value. Such an example is the PRINTDS record positioning keywords such as STARTKEY, ENDKEY, STARTRBA, ENDRBA etc. The specified value is converted from the external to the internal representation prior to execution of the command. Double ampersands and quotes are resolved to one unless they are enclosed in quotes. Ampersands followed by a 1- to 8-character name are considered symbolics and an attempt is made to resolve them unless they are enclosed in quotes. Values in quotes are not resolved and treated as is. A character value string or symbolic followed by a period "." indicates the termination of the specified value type. The value representations are listed below.

Values

BINARY must be preceded by the indicator B and the value must be enclosed in quotes. The value must be numeric in the range of a 0 or a 1 representing the bit string. The bit string need not fill an entire byte for it will be left justified and zero filled to the right.

Examples: B'1'
 B'01'
 B'100010001000'

CHARACTER may be specified as an implied character string, enclosed in quotes or preceded by the Indicator C and enclosed in quotes.

Examples: ABC implied character string
 'A B C' 'character string'
 C'A&&&C' C'character string'

HEXADECIMAL must be preceded by the indicator X and the value must be enclosed in quotes. The hex

value must be within the range of 0123456789ABCDEF and must be paired i.e., two external characters represent one internal character.

Examples: X'F2F1'
X'88256C'
X'F20102'

PACKED DECIMAL must be preceded by the indicator P and the value must be enclosed in quotes. The value must be numeric in the range of 0123456789. Positive/negative values are indicated with "+" or "-" signs as the first character following the quote.

Examples: P'21'
P'+3456'
P'-23456'

ZONED DECIMAL must be preceded by the indicator Z and the value must be enclosed in quotes. The value must be numeric in the range of 0123456789. Positive/negative values are indicated with "+" or "-" signs as the first character following the quote.

Examples: Z'21'
Z'+3456'
Z'-23456'

MIXED MODE is a combination of the values described above. Each value type must be prefixed with the type indicator i.e., B", H", P" or Z" with the exception of the character value. Strings must be enclosed in brackets.

Examples: (Z'12345'C'ABCD'X'1234'B'1111'P'123')
(12345.X'1234'C' 'B'11110011'P'-123')
(1.B'1101'X'F3'C'4'Z'5'&ZDATE&ZTIME)

Examples

```
/*          */  
/* print files */  
/*          */
```

```
PRINTDS DS(vsam.file) -  
    STARTKEY(X'00',C'KEY'X'01')
```

```
PRINTDS DS(vsam.file) -  
    STARTKEY(1.B'1101'X'F3'C'4'Z'5')
```

```
PRINTDS DS(esds.file) -  
    STARTRBA(X'00000200')
```

OFFLINE – Batch Prompt Panel Function

Offline Options Panel

The Offline Options Panel may be obtained by entering option 5 from the FileMarvel Primary Options Panel or entering the command OFFLINE from any FileMarvel panel. The OFFLINE function permits the user to submit various batch functions. The JOBCARD is required for any job submission so it should be primed prior to any other selection.

Fastpath command **OFFLINE**

```
FMV/VxRx----- FileMarvel Offline Functions -----  
OPTION ==>  
  
0 SET           - Modify the jobcard  
1 SOURCE       - Offline source library  
2 MANUAL       - Print FileMarvel manuals  
3 PRINTDS      - Print selected records or entire files  
4 CmplBOOK     - Compile copybooks offline  
5 SUBMIT       - Submit batch jobs from a PDS data set  
6 LABEL        - Display cartridge or tape labels  
7 COMPARDO     - Compare selected records or entire files  
8 COPYDO       - Copy selected records or entire files  
9 LISTVTOO     - Print volume data set list  
10 SPACEO      - Print volume free space  
11 LMODMAPO    - Load module statistics and code reversion
```

SET	Set jobcard Information.
SOURCE	Offline source storage directory.
MANUAL	Print FileMarvel Manuals.
PRINTDS	Print Entire or Partial Files.
CMPLBOOK	Compile Copybooks in Batch Mode.
SUBMIT	Submit batch JCL to the OFFLINE processor. .
LABEL	Display cartridge or tape labels within OFFLINE processing.
COMPARDO	Compare two datasets in batch mode.
COPYDO	Copy one dataset or records of one dataset to another dataset.
LISTVTOO	Print volume dataset list.
SPACEO	Print volume free space.
LMODMAPO	Print load module statistics and revert code

SET - Set Jobcard Information

Select SET from the OFFLINE, then select JOBCARD from the SET selection menu. The jobcard information is put to use when any a batch is submitted.

Fastpath command **JOBCARD**

```
FMV/VxRx-JOBCARD-GKH1----- Jobcard Modification Panel -----  
COMMAND ==>  
Main Commands: COPY, CANCEL, END, SAVE  
-----1-----2-----3-----4-----5-----6-----7-|-----  
//INTT006B JOB (X,X,X,X),INTT006 ',CLASS=X,MSGCLASS=A,  
//      NOTIFY=INTT006  
//*      COMMENT  
//PROCLIB DD DISP=SHR,DSN=FMARVEL.PROCLIB  
-----1-----2-----3-----4-----5-----6-----7-|-----
```

The JOBCARD JCL is used anytime FileMarvel submits a batch
To perform a background task (for example printing files).
Enter your JOBCARD JCL using //* as JCL comments for cards you
do not use.

Note: Enter the COPY command to copy JCL from a data set.

A jobcard statement of four lines is supported. The jobcard is automatically initialized using your userid and installation PROCLIB. Fill in installation appropriate parameters. The Jobcard is retained between ISPF sessions.

SOURCE – Source Storage Directory

Select SOURCE from the Offline Option Menu. SOURCE is a facility that permits the saving and recall of generated members from the other functions of OFFLINE. For example, the generated JCL from COPYDO may be saved, then recalled for modification and submission at a later time. A sample of a selection menu is displayed below.

Fastpath command **SOURCE**

```
FMV/L03-SOURCE-----Source Library Directory -----
COMMAND ==>                               Scroll ==> CSR
Commands: FT oper, L(A,OC mem), SelAll, Sel mem, SG
Linecmds: DEL, GBL, GRP, Ren, Sel
  Member  Userid  Action  Created  Changed  Used Description
----- Top of List -----
... ATEST1  TSODG      2004/07/29 2004/07/31  13 test job 1
... BTEST2  TSODG      2004/07/29 2004/08/01  10 test job 2
... CTEST3  TSODG      2004/07/29 2004/07/30   8 test job 3
... COMPARE TSODG      2004/07/28 2004/07/28   1 compare data set 1
... COMPARDO      2004/07/30 2004/07/30   1 compare data set 2
... COPY    TSODG      2004/07/28 2004/07/28   1 copy all files
... COPYO   TSODG      2004/07/30 2004/07/30   1 copy file 1 file2
... LABEL   TSODG      2004/07/31 2004/07/31   1 print labels
... MANUAL  TSODG      2004/07/31 2004/07/31   1 print manuals
```

MANUAL - Print Manuals Selection Panel

Select MANUAL from the Offline Option Menu. Print specific manuals by entering the selection for the desired manual. The Jobcard is required to be primed for this selection. The selected manuals are generated from the online HELP topics. A sample of a selection menu is displayed below.

Fastpath command **MANUAL**

```
FMV/VxRx-MANUAL----- Manual Selection Panel -----  
COMMAND ==>  
  
   1 BATCH      - FileMarvel Batch User Manual  
   2 DEMO       - FileMarvel Demo Walkthrough  
   3 EDIT       - FileMarvel Edit Commands  
   4 MANUAL     - FileMarvel Manual Generation  
   5 PRINTDS    - FileMarvel Print Data Set Manual  
   6 USERREF   - FileMarvel User Reference
```

PRINTDS - Print File Entry Panel

Select PRINTDS from the Offline Option Menu. Print all or a portion of records in a browse or edit space. Printing is performed by a batch job submitted through this entry panel. The PRINTDS batch job has a variety of keywords and parameters. For more information see Appendix B: Batch Print Parameters. To submit a batch print manually see Section 8.1 Submit Batch Print Manually.

Fastpath command **PRINTDS**

```
FMV/L02-PRINTDS----- Print Data Set Offline -----
COMMAND ==>
Commands: CAN, END, GEN, JOB, SUB
Data set name . . 'YOUR.DATASET'
      Data set name, * or wildcard pattern
Volume . .      For uncataloged data sets
Format mode . . S   S=Standard, H=Hex, F=Format, C=Compact D=Hexdump
Print width . . M   S=Standard, M=Max print format width

COMPILE or USE with FORMAT, COMPACT or REFORMAT display mode:
Copybook . .    Member name, *=Member list, BLANK=default
Compile . . N   Y=yes, N=no, E=extended
Language . . C   C=COBOL, P=PLI, copybook compile language
Copylib . .

APPLY CRITERIA . . N   Y=yes, N=no, E=extended
Start key . .
End key . .
Start record . . 00000001 Starting record number, if key not specified
Record count . . 00005000 Number of records to process
Record offset . . 00000001 Print record segment offset
Record length . . 99999999 Print record segment offset length
Enter the parameters, then enter SUBMIT to submit the JCL
```

Commands available:

- SUBMIT** - Submits job using fields input on this panel.
- JCL** - Edit the JCL generated from this panel before submission.
- JOBCARD**- Edit the default Jobcard (see Section 3.6 JOBCARD - Set Jobcard Information).

Fields:

DATA SET NAME The dataset name to be printed.

VOLUME Volume serial number for uncataloged non-VSAM datasets.

FORMAT MODE F= Format. Use the copybook information below to format the print by fields.
S= Standard.
H=Hex. Prints the file in an ISPF hex format.
C=Compact. Same as "F" except some field descriptions are omitted to print more fields per page.
D=Hex dump. Prints the file in a "dump" format.

PRINT WIDTH S =Standard. Truncate the format to an 80-character image. Can be abbreviated STD. M=Maximum print format width. Format the output record to the full width of the page. Can be abbreviated MAX.

COPYBOOK Required for Print Modes Format and Compact. The copybook source member name or compiled copybook name to use to format the print. Enter the copybook source member name here if FileMarvel has not compiled the copycode. If a file is already associated with a copybook template, do not specify this

field. If no copybook has been associated with the file and this field is left blank, the file is displayed in standard display mode regardless of the Display Mode specified. Enter “*” for a copybook selection list.

COMPILE Indicates whether FileMarvel will compile the copybook. When the copycode is compiled it creates a “copycode template” and is saved in the FileMarvel CONTROL file. Compilation is required only once. Subsequent file accesses using this copybook name use the copybook template most recently compiled.
Y - The Copybook will be compiled by FileMarvel before the print job is submitted.
N - The Copybook will not be compiled.
E - Will provide the Extended Copybook Compile Panel before the print job is submitted. See Section 4.19 Extended Copybook Compile Panel.

LANGUAGE Indicates the language of the source copybook member.
C - COBOL
P - PL/I
Do not specify this field for a file already associated with a copybook.

COPY LIBRARY The PDS where the source copybook member specified in the Copybook parameter resides. Specify only the dataset name. Do not include the member name here. This field is required only if the copybook is to be compiled. If the copybook has already been compiled, the copycode template is obtained from the FileMarvel CONTROL dataset.

APPLY CRITERIA Controls whether record/member selection is used.
Y - Record selection will occur. The selection criteria is specified in the fields **FROM KEY/TO KEY, FROM RECORD, or RECORD COUNT**.
E - Record selection will occur. A Record Selection Extended panel is displayed so selection criteria can be specified. See Section 4.1 Extended Search Prompt.
N - No record selection will occur. The entire dataset will be printed.

FROM KEY Used with **TO KEY** to specify a range of records to select from a VSAM KSDS dataset. Enter the lowest key to include in the range of records to print. A partial key may be entered. If the key contains non-displayable characters, enter the key in hex in the following format:

X’hhhhhh...’ Where hhhhhh = the hexadecimal representation of the key or partial key.

TO KEY is required if **FROM KEY** is specified.

TO KEY Used with **FROM KEY** to specify a range of records to select from a VSAM KSDS dataset. Enter the highest key to include in the range of records to print. A partial key may be entered. If the key contains non-displayable characters, enter the key in hex in the following format:

X’hhhhhh...’ Where hhhhhh = the hexadecimal representation of the key or partial key.

FROM RECORD Specifies the starting record number to select.

RECORD COUNT If **FROM RECORD** is specified, represents the maximum number of records to select.

To submit a batch print job, use the procedure “FMVBATCH”. The PRINTDS batch job has a variety of keywords and parameters. For more information see Appendix B: Batch Print Parameters.

An example print step is listed below:

```
// <<<JOB CARD>>>
//STEP EXEC FMVBATCH

SIGNON USERID(DAVID02)
PRINTDS DS(ACCOUNTS.PAY02.VSAM.JAN) -
COPYBOOK($CBKCOB) -
FORMAT(FORMAT) -
STARTKEY(X'00'C'CONTROL 'X'O'I) -
```

RECORDS(1)

/*

CMPLBOOK - Compile Copybooks Offline

Select CMPLBOOK from the Offline Option Menu. Submit the copybook for compilation in batch mode. The required compile controls are primed and must be altered as desired. The CMPLBOOK batch job has a variety of keywords and parameters. For more information on the required control, see Appendix A: Compile Control Parameters. A sample display is listed below.

Several options are available to perform a FileMarvel copybook template compile. See Section Compiling Copybooks. To compile a complex copybook or several copybook templates at once, use the batch utility. To perform a batch compilation, submit a job using the cataloged JCL procedure FMVBATCH.

Fastpath command **CMPLBOOK**

```
FMV/VxRx-CMPLBOOK----- Compile Copybooks Offline -----  
COMMAND ==>  
Commands: COPY, SAVE, SUBMIT  
----- Top of Data -----  
00001 //<JOB CARD>  
00002 //PRINTDS EXEC FMVBATCH  
00003  
00004 SIGNON USERID(TSODG )  
00005 COMPILE FUNC(COPYBOOK) -  
00006 LANGUAGE(COBOL) -  
00007 TITLE(?Description) -  
00008 MEMBER(?MEMBER ) -  
00009 USERID(TSODG )
```

COPY Enter the COPY command to copy controls already defined in the COPYBOOK COMPILE directory.

SAVE Enter the SAVE command to save the JCL and CONTROLS to an external dataset.

SUBMIT Enter the SUBMIT command to release the compile for batch offline processing.

SUBMIT - Submit Batch Jobs to the Offline Processor

The SUBMIT function submits JCL from a partitioned dataset. If a PDS dataset and member is entered the, the member JCL will be displayed for validation and the SUBMIT command will release the job for batch processing. If a PDS data is entered without a member, the directory is displayed for member selection. Only PDS, external vendor directories or UNIX files are the only type of datasets permitted.

Fastpath command **SUBMIT**

```

FMV/VxRx-SUBMIT----- Submit Batch JCL from a Data Set -----
COMMAND ==>

Data set name 'YDG.V3R3.PDS'
                PDS data Set name and member containing JCL to be submitted

Volume         Volume of the data set, if not cataloged

Enter the data set name and press enter to execute the request.
    
```

The SUBMIT function may be entered on any command entry line of any function or may be submitted as an AUTOEXEC command. Set an entry in the AUTOEXEC table as displayed below and then enter the AUTOEXEC command on any command entry line and the SUBMIT will be processed.

Short command	Exec	Command executed
-----	----	-----
COMPILE1	ON	SUBMIT 'TEST.PN09.SOURCE(COMPILE1)'
COMPILE2	ON	SUBMIT 'TEST.PN09.SOURCE(COMPILE2)'

LABEL - Display Cartridge or Tape Labels

The LABEL function displays tape or cartridge labels for standard labeled files. After priming the required fields, the generated JCL is displayed ready for batch submission. The batch processor formats the desired labels and optionally prints the labels in the desired print format. The formatted label display is printed on the print file and optionally as a TSO notification. See the FileMarvel administrator to activate TSO notification authorization.

The online form of LABEL generates JCL using standard allocation. See the BASIC topic of LABEL to generate the batch controls for dynamic allocation.

Fastpath command **LABEL**

```
FMV/VxRx-LABEL----- Display Tape Volume Label -----  
COMMAND ==>
```

Volume . .	Volume serial number
Unit . . CART	Device type name
Fileseq . . 001	File sequence number
Format . . N	N=no, S=standard, H=hex, D=hexdump format labels
Notify . . N	N=no, T=tso W=wto display on completion

VOLUME The volume serial number of the tape or cartridge standard labeled device.

Example 124589

UNIT The device type on which the VOLUME will be mounted on.

Example CART
 TAPE

FILESEQ The file sequence for the dataset label to be displayed. This sequence is the logical file number and not the physical or actual file sequence. The correct file sequence is calculated by FileMarvel.

FORMAT Optionally, format the actual label records in the desired format.

N=no	Do not print the actual labels
S=standard	Format the labels in standard mode
H=hex	Format the labels in ISPF hex mode
D=hex	Format the labels in hex dump mode

NOTIFY Optionally, notify the TSO requestor on completion of the label processing.

N=no	Do not notify the TSO requestor.
T=TSO	Send the formatted label as a TSO notification. This option requires authorization by the administrator.
W=WTO	Send the formatted label as a message on the job JES log.

COMPARDO - Compare Dataset Offline Prompt Panel

The COMPARDO function generates the controls that permit the comparison of two entire files, selective records within files or selective fields within records in the offline processor. The Dataset organizations do not have to match. Specific records may be selected for comparison using Multiple Selection Criteria. Selected fields within selected records may also be compared. Specific fields within records may be compared using a REFORMAT template. Check Appendix C for optional control formats. Prime the fields then enter GEN to create the required JCL.

Fastpath command: **COMPARDO**

```

FMV/VxRx-COMPARDO ----- Compare Data Sets Offline -----
COMMAND ==>
Commands: CAN, END, GEN, JOB, SUB
Data set one . . 'FMV.VxRx.DEMOFILE'
                   Data set name, * or wildcard pattern
Volume . .       For uncataloged data sets

Data set two . . 'TEST.FILE.DATASET'
                   Data set name, * or wildcard pattern
Volume . .       For uncataloged data sets

Compile or Use with REFORMAT compare mode for field compares:
Format mode . . . S          S=standard, H=hex, R=reformat, X=hex reformat
Template . .     CBKRFM     member name, *=member list, BLANK=default
Compile . .     Y           Y=yes, N=no, E=extended

Apply criteria . . Y           Y=yes, N=no, E=extended, R=reformat
Start key . .
End key . .
Start record . . 00000001   Starting record number, if key not specified
Record count . . 99999999   Number of records to process
Enter the parameters, then enter GEN to generate the JCL

```

DATA SET ONE Specifies the dataset name to be compared. Enter a partial name with an * to receive a dataset selection list. The following file types are supported:

- Sequential files
- PDS files with member specified
- PDS file for a member selection list
- VSAM files (KSDS, ESDS, RRDS)
- BDAM files
- GDG files

VOLUME Specify the volume if the dataset is not cataloged.

DATA SET TWO Specifies the second dataset name to be compared. Enter a partial name with an * to receive a dataset selection list. The following file types are supported:

- Sequential files
- PDS files with member specified
- PDS file for a member selection list
- VSAM files (KSDS, ESDS, RRDS)
- BDAM files
- GDG files

VOLUME Specify the volume if the dataset is not cataloged.

The following fields are used to create or use a reformat template in order to compare specific fields with in records.

COMPILE or USE with REFORMAT

FORMAT MODE Field mode determines the type of compare to be executed i.e., field vs. entire record and what format the record discrepancies are to be printed.

- S** Standard executes an entire record compare and formats discrepancies in standard mode
- H** Hexadecimal executes an entire record compare and formats discrepancies in ISPF hexadecimal mode
- R** Reformat requires a template name and compares the specified fields in the template
- X** Hexadecimal reformat requires a template name and compares in REFORMAT mode

TEMPLATE Member name, *=member list, BLANK=default

COMPILE Create or modify existing templates

- Y** Yes, execute a compile
- N** No, do not compile

APPLY CRITERIA The remaining fields pertain to selecting a subset of the files to be compared.

Y - Record/member selection will occur. The selection criteria is specified in the fields

START KEY/END KEY, START RECORD and RECORD COUNT.

E - Record selection will occur. A Record Selection Extended panel is displayed where multiple selection criteria can be specified. See Section 4.1 Extended Search Prompt.

N - No record selection will occur. The entire dataset is presented.

START KEY Use with **To Key** to specify a range of records to select from a VSAM KSDS dataset. Enter the lowest key to include in the range of records to compare. A partial key may be entered. If the key contains non-displayable characters enter the key in hex in the following format:

X'hhhhhh...' Where hhhhhh = the hexadecimal representation of the key or partial key.

If this field is specified, the **To Key** must also be specified..

END KEY Use with **From Key** to specify a range of records to select from a VSAM KSDS dataset. Enter the highest key to include in the range of records to compare. A partial key may be entered. If the key contains non-displayable characters, enter the key in hex in the following format:

X'hhhhhh...' Where hhhhhh = the hexadecimal representation of the key or partial key.

START RECORD Specifies the starting record number to be compared.

RECORD COUNT If **START RECORD** is specified, this is the maximum number of records to compare.

COPYDO - Copy Datasets Offline

Datasets may be copied via a batch job. The parameters and submission can be handled automatically by an online function COPYDS or entered and submitted manually. You can use the COPYDS command to copy entire files or selected records. The input and output files can differ in organization. COPYDS is a convenient method to merge keyed files. Invoke COPYDS repeatedly and specify a different source dataset to create a merged file.

Target record lengths may either be truncated or elongated. Installation specified pad characters can be used to fill the target record length.

Records may be expanded, reduced, fields inserted within records and reformatted as desired using COPYBOOK or REFORMAT templates. Fields within a record may be moved around using the REFORMAT technique. A field for example, a date may be expanded. For example a file containing a field with the format MMDDYY may be expanded to MMDDYYYY. The REFORMAT technique may be used to create test files as well. Another example may be moving COBOL source sequence from position 73-80 to position 1-6. Specific records may be selected from one file to another using multiple Boolean logic selection criteria.

The following combinations are permitted:

Sequential FB/VB/U	to a	Sequential FB/VB/U
Sequential FB/VB/U	to a	VSAM KSDS/ESDS/RRDS
VSAM KSDS/ESDS/RRDS	to a	VSAM KSDS/ESDS/RRDS
VSAM KSDS/ESDS/RRDS	to a	Sequential FB/VB/U

The following combinations are not permitted:

Sequential FB/VB/U	to a	PDS member U
PDS member FB/VB/U	to a	VSAM KSDS/RRDS
PDS member FB/VB	to a	PDS member U

Fastpath command **COPYDO**

FMV/VxRx-COPYDO----- Copy Data Sets Offline -----

Command ==>

Commands: CAN, END, GEN, JOB, SUB

Data set one . . 'YDG.V3R3.KSDS'
Source data set name, * or wildcard pattern
Volume . . For uncataloged data sets

Data set two . . 'YDG.V3R3.SEQ'
Target data set name, * or wildcard pattern
Volume . . For uncataloged data sets

File mode . . S S=standard, R=replace, E=extend, A=allocate
Format mode . . S S=standard, H=hex, R=reformat, X=hex reformat
Record pad . . 00 Two byte hexadecimal record expansion character

COMPILE or USE with REFORMAT compare mode for field compares:

Template . . CBKCOB Member name, *=member list, BLANK=default
Compile . . N Y=yes, N=no, E=extended

APPLY CRITERIA . . N Y=yes, N=no, E=extended, F=field

Start key/mem. . .

End key/mem. . . .

Start record . . . 00000001 Starting record number, if key not specified

Record count . . . 00000001 Number of records to process

Enter the parameters, then enter GEN to generate the JCL

DATA SET ONE Specifies the source dataset name to be copied. Enter a partial name with an * to receive a dataset selection list. The following file types are supported:

- Sequential files
- PDS files with member specified
- PDS file for a member selection list
- VSAM files (KSDS, ESDS, RRDS)
- BDAM files
- GDG files

VOLUME Specify the volume if the dataset is not cataloged.

DATA SET TWO Specifies the target dataset name to be copied to. Enter a partial name with an * to receive a dataset selection list. The following file types are supported:

- Sequential files
- PDS files with member specified
- PDS file for a member selection list
- VSAM files (KSDS, ESDS, RRDS)
- BDAM files
- GDG files

VOLUME Specify the volume if the dataset is not cataloged.

FILE MODE Controls how the target file will be saved or replaced.

A - allocate Permits the allocation of a new dataset. See Section 6.3 ALLOC - Allocate a Dataset

- E - extend** A save executed in this edit session will attempt to write the records to the end of the file. In VSAM/KSDS files the records are added to the file by key and existing records are not replaced.
- S - standard** This mode will attempt to write the file sequentially from the source file. If records exist in the file, sequential files are overwritten, PDS members are not replaced, VSAM/ESDS files are extended or VSAM/KSDS/RRDS are loaded as in an initial load.
- R - replace** This mode will attempt to write the file sequentially from the source file. If records exist in the file, sequential files are overwritten, PDS members are replaced, VSAM/ESDS files are extended or duplicate keys VSAM/KSDS/RRDS are replaced.

FORMAT MODE Controls whether the records are to be reformatted prior to saving the file or when errors occur print the records in the specified format.

F - Format Provides an edit session using a COPYCODE template to format the dataset. The Copybook parameter must be set.

S - Standard Save the records without modification.

H - Hex Provides an edit session in ISPF hexadecimal format with both the printable characters and their hex representations.

R - Reformat Provides a reorganized view of a record as defined by the reformat template. This view permits fields or groups of fields moved for ease of viewing.

D - Hexdump Provides an edit session in a “dump” format with both the printable characters and their hex representations.

RECORD PAD The hexadecimal pad character used to expand a target record if the target record is larger than the source record.

COPYBOOK Enter the copybook or reformat template member name is to be used or compiled during this record selection session. If a file is already associated with a copybook template, this field is not needed. Enter '*' for a list of available compiled copybooks.

COMPILE Indicates whether the copybook or reformat template specified is to be compiled for use by FileMarvel. When the copycode is compiled it will create a “copycode template” and will be saved in the FileMarvel CONTROL file. The copybook is required to be compiled only once. Subsequent file accesses using this copybook name will use the copybook template most recently compiled.

The copybook or reformat template will be obtained from the dataset specified on the Copybook Dataset parameter on the Copybook Option Settings Panel. See Section 3.3 COPYBOOK - Set Copybook Default Options.

APPLY CRITERIA The remaining fields pertain to selecting a subset of the file. When browsing files (of any type) or editing keyed files, a subset of the file can be processed. Specify selection criteria and choose whether or not to apply the criteria. PDS members may also be selected.

Y - Record/member selection will occur. If the dataset is a PDS, the selection criteria are applied to the member name. The selection criteria are specified in the fields **START KEY/END KEY, START RECORD and RECORD COUNT.**

N - No record selection will occur. The entire dataset is presented.

E - Record selection will occur. A Record Selection Extended panel is displayed where multiple selection criteria can be specified. See Section 4.1 Extended Search Prompt.

START KEY Use with **To Key** to specify a range of records to select from a VSAM KSDS dataset. Enter the lowest key to include in the range of records to edit. A partial key may be entered. If the key contains non-displayable

characters enter the key in hex in the following format:

X'hhhhhh...' Where hhhhhh = the hexadecimal representation of the key or partial key.

Use to specify a range of members to select from a PDS. Enter the name of the first member to include. A partial name may be entered.

If this field is specified, the **To Key** must also be specified..

END KEY Use with **From Key** to specify a range of records to select from a VSAM KSDS dataset. Enter the highest key to include in the range of records to edit. A partial key may be entered. If the key contains non-displayable characters, enter the key in hex in the following format:

X'hhhhhh...' Where hhhhhh = the hexadecimal representation of the key or partial key.

Use to specify a range of members to select from a PDS. Enter the name of the last member to include. A partial name may be entered.

If this field is specified, the **From Key** must also be specified.

START RECORD Specifies the starting record number to include in the edit session. If the dataset is a PDS, this is the starting member number relative to 1 included in the PDS member selection panel.

RECORD COUNT If **START RECORD** is specified, this is the maximum number of records to include in the edit session. If the dataset is a PDS, this is the maximum number of members displayed in the PDS member selection panel.

LISTVTOO – Print Volume Dataset List Offline

The LISTVTOO function displays datasets residing on direct access volumes. Datasets from a single, a generic or a group of volumes may be displayed on a single request. Selections can also be made from a volume list. On entry to the result of a request, dataset lists are displayed as encountered on the volume table of contents. Fields may be sorted according to datasets, dates, organizations, etc. Displays can be selected from three separate types of formats.

Fast path command: LISTVTOO

```
FMV/L02-LISTVTOO----- Volume Data Set Display -----
Command ==>
Commands: CAN, END, GEN, JOB, SUB

Volume . . . . . Volume, group, class or "*" for volume list

                        Classes: PRIVATE, PUBLIC, STORAGE
                        Example: STG001 STG* PRIVATE

Display option . . . L  L=dslist, D=data sets only, S=summary
Format mode . . . S   S=standard H=hex D=hexdump F=format

Sort field option. . DATASET  Sort field name or blank

APPLY CRITERIA . . . N    N=No, Y=Yes - use criteria below

From record. . . . 00000001 Starting record number
Record count . . . .99999999 Number of records to process
```

VOLUME Enter a volume, volume list or mount class of volumes.

* display volume unit name selection list

volume 6-character volume serial name

ex. VOL001

volume group 1- to 6-character volume serial name followed by an asterisk to delineate the end of a volume name string. All volumes containing the volume prefix will be scanned for datasets or criteria.

ex. VOL0*

DISPLAY OPTION Dataset entry display type

L List volume summary and dataset list

D List dataset list only

S List volume summary only

FORMAT MODE Indicates the file formatting mode.

S Standard formatting mode, which lists the datasets residing on the volume.

H Hexadecimal formatting mode prints each vtoc record.

D Hexadecimal dump formatting mode prints each vtoc records.

F Format records using a compiled copybook and prints each vtoc record.

SORT FIELD OPTION Sort the volume display according to the field name specified. Issue a

request without the sort field to determine the optional field names.

ASIS blank	do not sort
DATASET	dataset name
DSORG	dataset organization
RECFM	record format
LRECL	record length
BLKSZ	block size
EXTN	extents
USETRK	used tracks
CREATED	creations date
VOLUME	volume serial

APPLY CRITERIA Whether the selection criteria specified below are to be applied.

Y	Use the selection criteria specified below.
N	Specifies all selection requests are ignored.

FROM RECORD Specifies the starting record number to process.

RECORD COUNT Specifies the maximum number of records to process.

Commands:

CANCEL | CAN Cancel the function, subfunction, panel or edit session without any changes.

END Verify and prime the arguments and return to the previous panel and/or save the edit session. END is usually assigned to PF3.

GENERATE | GEN Generate JCL for offline submission

JOBCARD | JOB Display & alter jobcard for offline submission

SUBMIT | SUB Submit the edit session JCL stream for batch or offline execution

SPACEO – Print Volume Free Space Offline

The SPACEO function lists direct access free space. A formatted list of volumes is displayed with free space statistics. The volume request may be a single volume, group of volumes or by mount status.

Fastpath command **SPACEO**

```
FMV/L03-SPACEO----- Dasd Free Space Utility -----
Command ==>
Commands: CAN, END, GEN, JOB, SUB

Volume . . .      Enter volume, volume list, group or class

                   Classes: PRIVATE, PUBLIC, STORAGE or ALL
                   Ex.   STG001 STG* PRIVATE

                   Free space may be displayed for a specific
                   volume, a volume group or a class group.
```

VOLUME Enter a volume, volume list or mount class of volumes.

volume a single volume as STG001

volume list a list group delineated by an asterisk such as STG*

mount class volume mount class i.e., PRIVATE, PUBLIC or STORAGE

LMODMAPO – Print Load Module Statistics & Code Reversion Offline

The LMODMAP function displays the directory for a link edit partitioned datasets. Members are selected for offline to print module statistics, csect member hex dumps or reverted source code.

Fastpath command **LMODMAPO**

```

FMV/L03-LMODMAPO----- Load Module Analysis -----
Command ==>
Commands: CAN, END, GEN, JOB, SAVE, SUBmit

LOADLIB data set name . .
Volume. . . . .           For uncataloged data sets

Request . . . . . ALL      Function request
                          ALL, INFO, HEXDUMP, REVERT

  Limit member list

APPLY CRITERIA . . . . N   Y=yes, N=no
Start mem . . . . .       Set member start
End mem . . . . .         Set member end
  
```

DATA SET NAME enter a fully qualified dataset name in quotes of a linkedit PDS

VOLUME enter the volser for the dataset if not catalogued

REQUEST enter the display request type

- ALL display all request types
- INFO display linkedit load module statistics
- HEXDUMP display csect code in hexadecimal format
- REVERT display csect code in reverted source code

APPLY CRITERIA whether the selection criteria specified below is to be applied

- Y Use the selection criteria specified below to select members
- N Specifies all selection requests are ignored

START MEM Specify the lowest member or generic member to begin the display

Example FMVC

END MEM Specify the highest member or generic member to end the display

Example FMVDU

You can specify the member range to limit the number for directory display

Example specific FMVALOC
 generic FMVA