

Keywords for EXTLAYOUT

DBD	LIB	REPLACEMEM
LAYOUTFILE	OUTFILE	XREFFILE

Keywords for EXTRACT

ACB	FORMAT	PADCHAR	SELRECIF, AND, OR
ACCUM	IF, AND, OR	PRINTLIM	SELSEG, AND, OR
CHANGE	INFILE	PRINTREC	SETRC
DBD	INLIM	PRINTSEG	SKIP
DBDLIB	INTERVAL	PSB	SKIPRECIF, AND, OR
DBRC	LAYOUTFILE	PSBLIB	SKIPSEGIF, AND, OR
EDIT	LINEPAGE	REPLACE	STARTKEY
EMPTYRC	MOVE	SEGLIM	STOP
ENV	NEXTREC	SEGMENT	TRUNCRC
ENVLIB	NOSELRC	SELECT	WRITE
EXTRACTFILE	OUTLIM	SELLIM	XREFFILE

Keywords for GENSOURCE

LIB	MEMBER	OUTFILE	REPLACEMEM
-----	--------	---------	------------

Keywords for INIT

DBD	DBRC	ENVLIB
DBDLIB	ENV	

Keywords for PRINT

ACB	IF, AND, OR	PRINTLIM	SKIP
ACCUM	INFILE	PSB	SKIPRECIF, AND, OR
CHANGE	INLIM	PSBLIB	SKIPSEGIF, AND, OR
DBD	INTERVAL	REPLACE	STARTKEY
DBDLIB	LAYOUTFILE	SEGLIM	STOP
DBRC	LINEPAGE	SEGMENT	TRUNCRC
EDIT	MOVE	SELECT	WRITE
EMPTYRC	NEXTREC	SELLIM	XREFFILE
ENV	NOSELREC	SELRECIF, AND, OR	
ENVLIB	OUTLIM	SELSEGIF, AND, OR	
FORMAT	PADCHAR	SETRC	

Keywords for PRINTLOG

FIELDDISPLAY	LAYOUTFILE	LOGFILE	SETRC
FORMAT	LINEPAGE	PRINTLIM	XREFFILE

Keywords for READ

ACB	INFILE	PRINTLIM	SKIPRECIF
ACCUM	INLIM	PRINTREC	SKIPSEGIF
DBD	INTERVAL	PRINTSEG	STARTKEY
DBDLIB	LAYOUTFILE	PSB	STOP
EMPTYRC	LINEPAGE	PSBLIB	TRUNCRC
ENV	MOVE	SELECT	WRITE
ENVLIB	NEXTREC	SELLIM	XREFFILE
FORMAT	OUTLIM	SETRC	
IF, AND, OR	PADCHAR	SKIP	

Keywords for RELOAD

ACB	FORMAT	OUTLIM	*SELRECIF, AND, OR
ACCUM	**IF, AND, OR	PADCHAR	*SELSEGIF, AND, OR
CHANGE	INLIM	PRINTLIM	SETRC
DBD	*INTERVAL	PRINTREC	SKIP
DBDLIB	LAYOUTFILE	PRINTSEG	*SKIPRECIF
DBRC	LAYOUTRC	PSB	*SKIPSEGIF
EDIT	LINEPAGE	PSBLIB	*STARTKEY
EMPTYRC	MODE	REPLACE	STOP
ENV	MOVE	REPLACEKEY	TRUNCRC
ENVLIB	NEXTREC	SEGMENT	WRITE
EXTRACTFILE	NOSELREC	*SELECT	XREFFILE

*The keyword can only apply to ROOT segments for the RELOAD command.

** When IF is used on the RELOAD command, all subsequent actions must act on the same segment as IF.

Keywords for UPDATE

ACB	FORMAT	PADCHAR	SETRC
ACCUM	IF, AND, OR	PRINTLIM	SKIP
CHANGE	INFILE	PRINTREC	SKIPRECIF, AND, OR
DBD	INLIM	PRINTSEG	SKIPSEGIF, AND, OR
DBDLIB	INSERT	PSB	STARTKEY
DBRC	INTERVAL	PSBLIB	STOP
DELETE	LAYOUTFILE	REPLACE	TRUNCRC
EDIT	LINEPAGE	ROOTKEY	WRITE
EMPTYRC	MOVE	SEGLIM	XREFFILE
ENV	NEXTREC	SELECT	
ENVLIB	OUTLIM	SELLIM	

Keyword Abbreviation/Syntax Table

Keyword	Abbreviation	Syntax
ACB	ACB	(<i>acbname</i> [, <i>pcb_sequence_number</i>])
ACCUM	ACCUM	(<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>length</i> [, <i>decimal-positions</i>], <i>data-type</i>]} [, ' <i>description</i> '])
CHANGE	CHA	(<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>scan-length</i>]} , <i>operator</i> , <i>from-data</i> , { <i>to-data INFILE</i> (<i>infile-position</i> , <i>length</i>)} [, ALL])
DBD	DBD	(<i>dbdname</i>)
DBDLIB	DBDLIB	(DBDLIB <i>dsname</i> <i>ddname</i>)
DBRC	DBRC	(Y N)
DELETE	DEL	(<i>{segment-name ROOT}</i>)
EDIT	EDIT	(<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>scan-length</i>]} , <i>operator</i> , <i>from-data</i> , { <i>to-data INFILE</i> (<i>infile-position</i> , <i>length</i>)} [, ALL])
EMPTYRC	EMPTYRC	(4 <i>return-code</i>)
ENV	ENV	(<i>envname</i>)
ENVLIB	ENVLIB	(ENVLIB <i>dsname</i> <i>ddname</i>)
EXTRACTFILE	EF	(<i>{[EXTRACT dsname ddname]}</i>)
FIELDDISPLAY	FD	(A M)
FORMAT	FORMAT	(<i>{[CHARACTER HEX SINGLE]}</i>)

IF, AND, OR	IF, AND, OR	(<i>{segment-name ROOT</i> [, PRESENT NOTPRESENT]) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>scan-length</i>]} , <i>operator</i> , { <i>data INFILE</i> (<i>infile-position</i> , <i>length</i>)} [, INANY INALL INNONE]) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> , <i>scan-length</i> } , <i>operator</i> , { <i>field-name position</i> }) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> , <i>scan-length</i> } , <i>operator-type</i>)
INFILE	INFILE	(<i>{[INFILE dsname</i> (<i>member</i>)] <i>ddname</i> [, RESET RETAIN]}) (0 <i>number</i>)
INSERT	INSERT	(<i>{segment-name ROOT}</i>)
INTERVAL	INTERVAL	(<i>{segment-name ROOT}</i> , <i>number_in_group</i> , <i>number_to_skip</i>)
LAYOUTFILE	LF	(LAYOUT <i>dsname</i> <i>ddname</i>)
LAYOUTRC	LAYOUTRC	(4 <i>return-code</i>)
LIB	LIB	(DBDLIB <i>dsname</i> <i>ddname</i>)
LINEPAGE	LP	(60 <i>number</i>)
LOGFILE	LOGFILE	(LOGFILE <i>dsname</i> <i>ddname</i>)
MEMBER	MEM	(<i>[member]</i> [<i>pattern</i>] [<i>startmember-endmember</i>] [, ...])
MODE	MODE	(<i>{LOAD UPDATE}</i>)
MOVE	MOVE	(<i>to-position</i> , <i>INFILE</i> (<i>infile-position</i> , <i>length</i>)) (<i>to-position</i> , <i>to-data</i>) (<i>to-position</i> , { <i>from-segment-name ROOT</i> } , { <i>length</i> , <i>from-position</i> <i>from-field-name</i> }) (CLEAR)
NEXTREC	NEXTREC	
NOSELRC	NOSELRC	(4 <i>return-code</i>)
OUTFILE	OUTFILE	(<i>{[OUTFILE dsname ddname]}</i>)
OUTLIM	OUTLIM	(0 <i>number</i>)
PADCHAR	PADC	(X'00' <i>char-data</i>)
PRINTLIM	PRINTLIM	(0 <i>number</i>)
PRINTREC	PR	
PRINTSEG	PS	(<i>{segment-name ROOT}</i>)
PSB	PSB	(<i>psbname</i> [, <i>pcb_sequence_number</i>])
PSBLIB	PSBLIB	(<i>{PSBLIB dsname ddname}</i>)
REPLACE	REP	(<i>{segment-name ROOT}</i> , { <i>field-name position</i> } , { <i>to-data INFILE</i> (<i>infile-position</i> , <i>length</i>)}) REPLACE (<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>scan-length</i>]} , <i>operator</i> , <i>from-data</i> , { <i>to-data INFILE</i> (<i>infile-position</i> , <i>length</i>)} [, ALL])
REPLACEKEY	RK	(<i>{N Y D}</i>)
REPLACEMEM	RM	(N Y)
ROOTKEY	ROOTKEY	(<i>key-position</i>)
SEGLIM	SEGLIM	(<i>{segment-name ROOT}</i> , <i>number</i>)
SEGMENT	SEG	(<i>{segment-name ROOT}</i> [, ...])
SELECT	SELECT	(<i>{segment-name ROOT}</i> , <i>number</i>)
SELLIM	SELLIM	(0 <i>number</i>)
SELRECIF	SELR	(<i>{segment-name ROOT</i> [, PRESENT NOTPRESENT]) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> [, <i>scan-length</i>]} , <i>operator</i> , { <i>data INFILE</i> (<i>infile-position</i> , <i>length</i>)} [, INANY INALL INNONE]) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> , <i>scan-length</i> } , <i>operator</i> , { <i>field-name position</i> }) (<i>{segment-name ROOT}</i> , { <i>field-name position</i> , <i>scan-length</i> } , <i>operator-type</i>)

SELSEGIF	SELS	{{segment-name ROOT[,PRESENT NOTPRESENT]}} {{segment-name ROOT},{field-name position[,scan-length]},operator,{data INFILE(infile-position,length)}[INANY INALL INNONE]} {{segment-name ROOT},{field-name position,scan-length},operator,{field-name position}} {{segment-name ROOT},{field-name position,scan-length},operator-type)
SETRC	SETRC	(return-code)
SKIP	SKIP	(number)
SKIPRECIF	SKIPR	{{segment-name ROOT[,PRESENT NOTPRESENT]}} {{segment-name ROOT},{field-name position[,scan-length]},operator,{data INFILE(infile-position,length)}[INANY INALL INNONE]} {{segment-name ROOT},{field-name position,scan-length},operator,{field-name position}} {{segment-name ROOT},{field-name position,scan-length},operator-type)
SKIPSEGIF	SKIPS	{{segment-name[,PRESENT NOTPRESENT] ROOT}} {{segment-name ROOT},{field-name position[,scan-length]},operator,{data INFILE(infile-position,length)}[INANY INALL INNONE]} {{segment-name ROOT},{field-name position,scan-length},operator,{field-name position}} {{segment-name ROOT},{field-name position,scan-length},operator-type)
STARTKEY	STARTK	(root-key)
STOP	STOP	[((RESET RETAIN))]
SYNCD	SYNCD	(100 number)
SYNCI	SYNCI	(100 number)
SYNCU	SYNCU	(100 number)
TRUNCRC	TRUNCRC	(4 return-code)
WRITE	WR	(ddname{,...})
XREFFILE	XF	((XREF ddname dsname))

Selection Criteria - Dynamic Record Selection Criteria

You can filter IMS database records using selection criteria. Some examples of valid selection criteria are listed in the following table:

You can elect to Browse the selected segment's entire IMS database record, or the selected segment's parental path and child segments.

Selection Criteria	Description
IFREC(segname)	Applies selection criteria to a specific record. If no IFREC or IFSEG is specified, dynamic selection criteria apply to the ROOT segment. The entire IMS database record is processed if a segment matches the selection criteria.
IFSEG(segname)	Applies selection criteria to a specific segment name. If no IFREC or IFSEG is specified, dynamic selection criteria apply to the ROOT segment. The selected segment is included, along with all its children and I of the segments in its parental path. All children of the segments in the parental path are also included. No twins of the selected segment or of the segments in the parental path or any of their children are included.
80 EQ C'NY'	Position 80 for a length of 2 equals 'NY'

STATE-CODE NE 'NY'	Field STATE-CODE not equal 'NY'(requires LAYOUT info) STATE-CODE is equal to 'NY' or 'TX'
STATE-CODE = C'NY,TX' 80 = C'NY,TX' AND Position 80 for a length of 2 equals 'NY' or 'TX', AND 100 = C"100,000" position 100 for a length of 7 equals '100,000'	Field TRAN-DD is '02' '03' '04', or '05' Position 10 for a length of 4 is a valid packed value 2-byte packed field equals P'0' or P'1' or P'999' 'new york' (any case) anywhere in position 101-120
TRAN-DD GT '01' AND TRAN-DD LT '06' 10(4) EQP	Start segment selection with the root segment whose key is GTEQ to '56789' Stop reading after reading 5000 IMS database records
100(2) = P'0,1,999' 101(20) CO 'NEW YORK'	Only select 1000 IMS database records based on the selection criteria Compound condition (OR also supported) Compound condition with fields
STARTKEY='56789'	Specify multiple literals by listing the literals or by listing the literal values within apostrophes separated by a comma. Some examples of conditions with multiple literal values follow:
INLIM=5000	80 = C'NY' C'NJ' C'MA' Position 80 equals one of the three values
SELLIM=1000	80 = C'NY,NJ,MA' Position 80 equals one of the three values
80 = C'NY' AND 100 = C'001'	80 = C"NY,NJ,MA" Position 80 equals value 'NY,NJ,MA' (with commas)
STATE-CODE EQ C'NY' AND COUNTY-CODE EQ C'001'	100(2) = P'0,1,999' 2-byte packed field equals one of three values
Specify multiple literals by listing the literals or by listing the literal values within apostrophes separated by a comma. Some examples of conditions with multiple literal values follow:	'abc,123,xyz' Segment contains one of the three values
	"abc,123,xyz" Segment contains value 'abc,123,xyz' (with commas)

When comparing to another field within a segment, the part of the condition following the operator specifies a numeric field position within the segment or a segment layout field name.

Examples:

80(2)= 84 means position 80 for a length of 2 equals position 84 and 85

STATE-CODE EQ PREV-STATE-CODE means field STATE-CODE equals PREV-STATE-CODE

A condition to select segments based on the presence of a character string anywhere within a field uses the CO (contains) operand.

Example:

101(20) CO C'NEW YORK' Looks for the character string in columns 101 - 120

When looking for a character string anywhere within a segment, specify the literal value:

C'NEW YORK'	Looks for 'NEW YORK' (upper case) anywhere in a segment
C'new york'	Looks for 'new york' (lower case) anywhere in a segment
C"100,000"	Looks for '100,000' (including comma) anywhere in a segment
'NEW YORK' or 'new york'	Looks for 'new york' (any case) anywhere in a segment
X'FFFF'	Looks for hex FFFF anywhere in a segment
P'0'	Packed zero value (length 1-16) anywhere in a segment

INTERVAL(3,2) Selects three segments starting with the first segment and then skips the next two. Continues this selection process until the end of database is reached.

Quick Reference

CA File Master™ Plus for IMS

Release 9.1

Menu Options

Option	Description
0 SETUP	Setup and processing parameters
0.1 DEFAULT	Processing defaults and jobcard
0.2 PRINT	Print output control parameters
0.3 PARM FILES	Definition and update of processing data sets
0.4 XREF	Segment Layout Cross Reference
0.5 IMS ENV	DLI & BMP Environments
1 BROWSE	Browse IMS Database
2 EDIT	Edit IMS Database
3 UTILITIES	Utility functions
3.1 EXTRACT	Extract of IMS Database to Sequential File
3.2 RELOAD	Reload of IMS Database from Extract File
3.3 INIT	Initialization of IMS Database
3.4 DBCHART	Presentation of IMS Hierarchy from DBD Member
3.5 SOURCE	Generate ACB, DBD, MDA and PSB Source
3.6 Ext Layout	Generate Extract File Record Layouts
4 PRINT	Print data from IMS Database
5 FILTER	Define or update selection criteria
6 LAYOUT	View or update record layout
D DSN LIST	Select DSN for processing
X EXIT	Exit
Z ABOUT	About the product

Batch Commands and Keywords

The following table shows an alphabetical listing of the primary commands and functions performed using CA File Master Plus for batch:

Command	Description
DBCHART	Prints the hierarchical chart of an IMS database
EXTLAYOUT	The Extract Layout utility creates record layouts
EXTRACT	Extracts segments from a database and writes them to a sequential file
GENSOURCE	Generates source for an ACB, DBD, PSB or MDA
INIT	Initializes uninitialized files associated with a primary database
PRINT	Prints segments, records and reports
PRINTLOG	Prints the Edit Change Log report
READ	Reads segments in a sequential manner and acts on these segments
RELOAD	Populates a database with segments from an extract file
UPDATE	Changes, inserts or deletes segments in a database

Keywords for DBCHART

DBD	LIB	LINEPAGE
-----	-----	----------

Printing Instructions: Set the laser printer Print Properties to Legal (8.5x14), 2-sided/Duplex, 'Flip' or 'Rotate' on Short Edge. After printing, fold the paper between the three columns accordion style to create an easy-to-read quick reference.

All product names referenced herein are either trademarks or registered trademarks of their respective companies.
Copyright © 2015 CA. All rights reserved