

CA-IDMS[®]

Programming Quick Reference

15.0



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED, OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

First Edition, December 2000

© 2000 Computer Associates International, Inc.
One Computer Associates Plaza, Islandia, NY 11749
All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.

Contents

| | |
|---|------|
| How to Use This Guide | v |
| Chapter 1. COBOL syntax | 1-1 |
| 1.1 COBOL Precompiler options | 1-2 |
| 1.2 COBOL Precompiler-Directive statements | 1-3 |
| 1.3 COBOL DML statements | 1-6 |
| Chapter 2. FORTRAN | 2-1 |
| 2.1 Compiler options statements | 2-2 |
| 2.2 Compiler-directive statements | 2-3 |
| 2.3 Control statements | 2-4 |
| 2.4 Retrieval statements | 2-5 |
| 2.5 Modification statements | 2-6 |
| 2.6 Save statements | 2-7 |
| 2.7 Logical Record Facility statements | 2-8 |
| 2.8 The DMLR Precompiler | 2-9 |
| Chapter 3. PL/1 syntax | 3-1 |
| 3.1 PL/1 Precompiler options | 3-2 |
| 3.2 PL/1 Precompiler-Directive statements | 3-3 |
| 3.3 PL/1 DML statements | 3-6 |
| Chapter 4. Assembler Syntax | 4-1 |
| 4.1 Assembler Precompiler options | 4-2 |
| 4.2 Assembler Precompiler-Directive statements | 4-4 |
| 4.3 Assembler DML statements | 4-6 |
| Chapter 5. DML statement function table | 5-1 |
| Chapter 6. Call formats | 6-1 |
| 6.1 CA-IDMS/DB call formats | 6-2 |
| 6.2 CA-IDMS/DC call formats | 6-10 |
| Chapter 7. IDMS-DB/DC Communications block | 7-1 |
| 7.1 16-byte IDMS communications block | 7-2 |
| 7.2 IDMS communications block fields | 7-3 |
| 7.3 LRC block | 7-5 |
| 7.4 IDMS/DC communications block | 7-6 |
| Chapter 8. Run-time error-status codes | 8-1 |
| 8.1 Major DB status codes | 8-2 |
| 8.2 Minor DB status codes | 8-3 |
| 8.3 Major DC status codes | 8-8 |
| 8.4 Minor DC status codes | 8-9 |
| 8.5 ERROR-STATUS condition names | 8-13 |
| 8.6 IDMS-DC/UCF DMLA run-time R(15) values | 8-14 |

| | |
|--|------|
| Chapter 9. TP monitor coding requirements | 9-1 |
| Chapter 10. Record/set representation | 10-1 |
| Chapter 11. EMPLOYEE database structure diagram | 11-1 |
| Chapter 12. Online debugger syntax | 12-1 |
| 12.1 Debugger symbols | 12-2 |
| 12.2 User symbols | 12-3 |
| 12.3 Program symbols | 12-4 |
| 12.4 Expression operators | 12-5 |
| 12.5 Delimiters | 12-6 |
| 12.6 Debugger commands | 12-7 |

How to Use This Guide

This Quick Reference applies to Release 15.0 of CA-IDMS and accompanies the following reference guides:

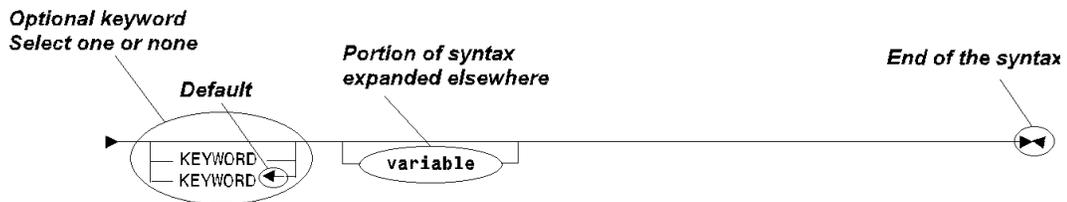
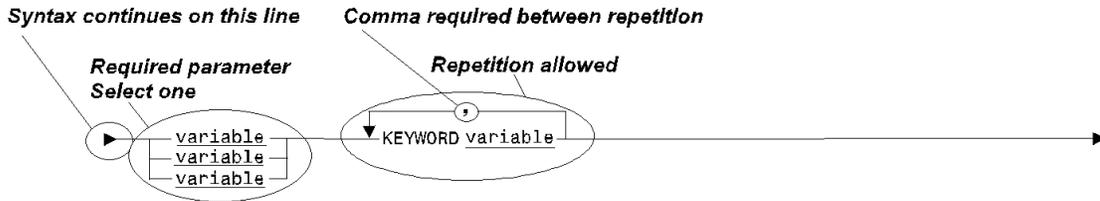
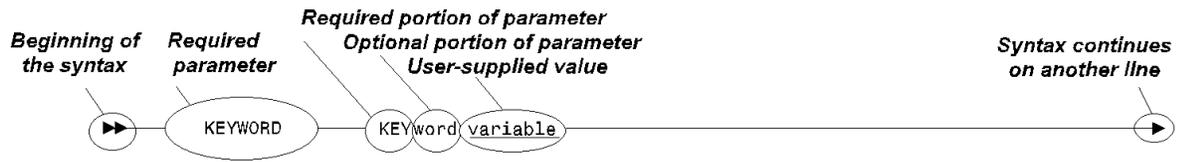
- *CA-IDMS DML Reference — Assembler*
- *CA-IDMS DML Reference — COBOL*
- *CA-IDMS DML Reference — PL/I*
- *CA-IDMS Online Debugger*

Understanding syntax diagrams

Look at the list of notation conventions below to see how syntax is presented in this manual. The example following the list shows how the conventions are used.

| | |
|---------------------------------------|---|
| UPPERCASE OR SPECIAL CHARACTERS | Represents a required keyword, partial keyword, character, or symbol that must be entered completely as shown. |
| lowercase | Represents an optional keyword or partial keyword that, if used, must be entered completely as shown. |
| <u>underlined lowercase</u> | Represents a value that you supply. |
| ◀ | Points to the default in a list of choices. |
| lowercase bold | Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document. |
| ▶▶ | Shows the beginning of a complete piece of syntax. |
| ◀◀ | Shows the end of a complete piece of syntax. |
| ▶ | Shows that the syntax continues on the next line. |
| ▶ | Shows that the syntax continues on this line. |
| ▶ | Shows that the parameter continues on the next line. |
| ▶ | Shows that a parameter continues on this line. |
| ▶ parameter ▶ | Shows a required parameter. |
| ▶ parameter parameter ▶ | Shows a choice of required parameters. You must select one. |
| ▶ parameter ▶ | Shows an optional parameter. |
| ▶ parameter parameter ▶ | Shows a choice of optional parameters. Select one or none. |
| ▶ parameter ▶ | Shows that you can repeat the parameter or specify more than one parameter. |
| ▶ parameter , parameter ▶ | Shows that you must enter a comma between repetitions of the parameter. |

Sample syntax diagram



Chapter 1. COBOL syntax

1.1 COBOL Precompiler options

Dictionary ready override

Begin in column 7.

▶▶ [*RETRIEVAL _____ ▶▶
 └ *PROTECTED-UPDATE ┘

Comment generation

▶▶ *SCHEMA-COMMENTS _____ ▶▶

List generation

▶▶ [*DMLIST _____ ▶▶
 └ *NODMLIST ◀ ┘

Log suppression

▶▶ *NO-ACTIVITY-LOG _____ ▶▶

ON statement

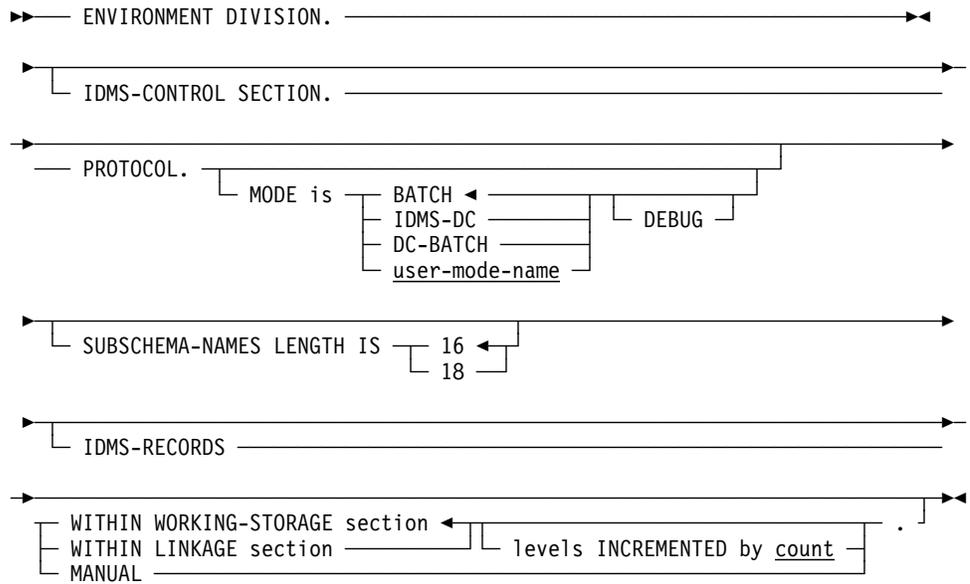
▶▶ ON condition-name imperative-statement . _____ ▶▶

1.2 COBOL Precompiler-Directive statements

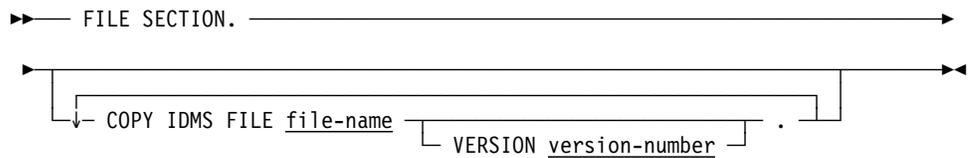
IDENTIFICATION DIVISION

▶— IDENTIFICATION DIVISION. —————▶
▶— PROGRAM-ID. program-name —————▶
 └ VERSION version-number ─┘

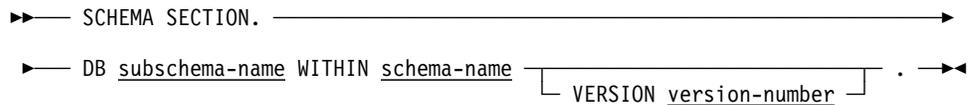
ENVIRONMENT DIVISION



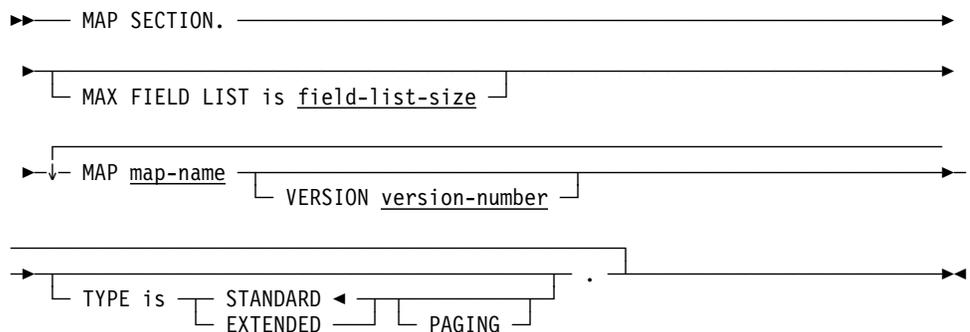
DATA DIVISION



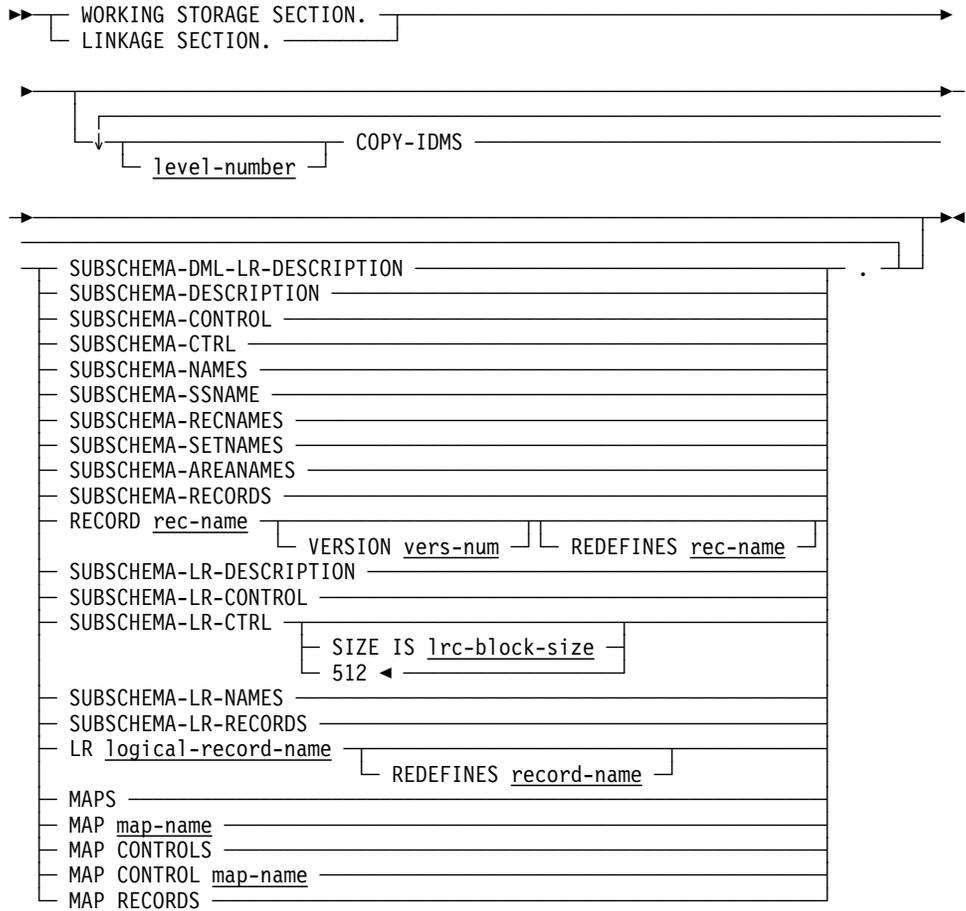
SCHEMA SECTION



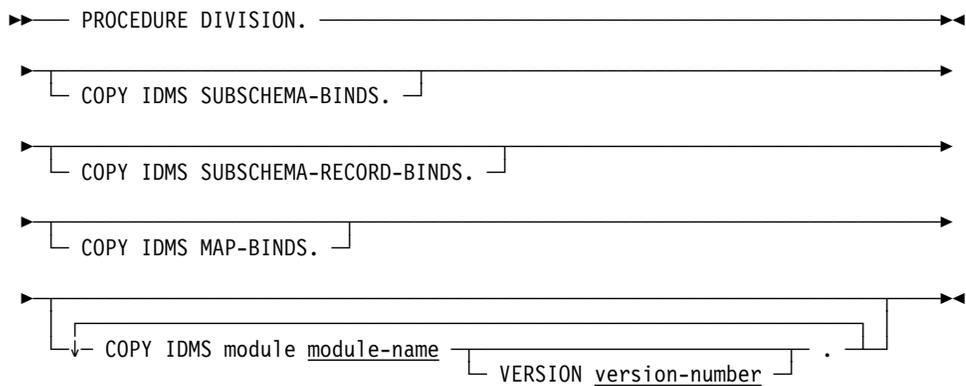
MAP SECTION



WORKING STORAGE SECTION



PROCEDURE DIVISION



1.3 COBOL DML statements

ABEND

```

▶▶— ABEND CODE abend-code [ DUMP ] [ NODUMP ] [ EXITS ] [ INVOKED ] [ IGNORED ] .

```

ACCEPT

```

▶▶— ACCEPT [ TASK CODE ] [ TASK ID ] [ LTERM ID ] [ PTERM ID ] [ SYSVERSION ] [ USER ID ] [ SCREENSIZE ] INTO return-location .

```

ACCEPT BIND ADDRESS

```

▶▶— ACCEPT bind-address FROM record-name BIND .

```

ACCEPT DATABASE STATISTICS

```

▶▶— ACCEPT db-statistics FROM IDMS-STATISTICS .

```

ACCEPT DB-KEY FROM CURRENCY

```

▶▶— ACCEPT db-key-location FROM [ record-name ] [ set-name ] [ area-name ] CURRENCY .

```

ACCEPT DB-KEY RELATIVE TO CURRENCY

```

▶▶— ACCEPT db-key-location FROM set-name [ NEXT ] [ PRIOR ] [ OWNER ] CURRENCY .

```

ACCEPT page-info-location

```

▶▶— ACCEPT page-info-location FOR record-name .

```

ACCEPT PROCEDURE CONTROL LOCATION

```

▶▶— ACCEPT procedure-control-location FROM procedure-name PROCEDURE .

```

ACCEPT TRANSACTION STATISTICS

```

▶▶— ACCEPT TRANSACTION STATISTICS [ WRITE ] [ NOWRITE ] INTO return-stat-data-location .

```

ATTACH

```

▶▶— ATTACH TASK CODE 'task-code' [ PRIORITY priority ] [ WAIT ] [ NOWAIT ] .

```

BIND MAP

```

▶▶ BIND MAP map-name [ RECORD rec-name ] TO [ NULL ] rec-location .

```

BIND PROCEDURE

```

▶▶ BIND PROCEDURE FOR procedure-name TO procedure-control-location .

```

BIND RECORD

```

▶▶ BIND [ record-name ] TO record-location [ WITH record-name ] .

```

BIND RUN-UNIT

```

▶▶ BIND RUN-UNIT [ FOR subschema-name ]
[ DBNODE nodename ] [ DBNAME database-name ]
[ DICTNODE nodename ] [ DICTNAME dictionary-name ] .

```

BIND TASK

```

▶▶ BIND TASK [ NODENAME node-name ] .

```

BIND TRANSACTION STATISTICS

```

▶▶ BIND TRANSACTION STATISTICS .

```

CHANGE PRIORITY

```

▶▶ CHANGE PRIORITY TO priority .

```

CHECK TERMINAL

```

▶▶ CHECK TERMINAL [ GET STORAGE ]
[ INTO input-data-location ] [ TO end-input-data-location ]
[ MAX LENGTH input-data-max-length ]
[ RETURN LENGTH INTO input-data-actual-length ] .

```

COMMIT

```

▶▶ COMMIT [ TASK ] [ ALL ] .

```

CONNECT

▶▶ CONNECT record-name TO set-name .

DC RETURN

▶▶ DC RETURN

┌ NEXT TASK CODE next-task-code ┐

┌ NORMAL ◀ ┐
┌ ABORT ─ ┐
┌ CONTINUE ─ ┐
┌ IMMEDIATE ─ ┐

┌ TIMEOUT ─ ┐
┌ INTERVAL timeout-interval ┐
┌ PROGRAM timeout-program ┐

┌ NEXT TASK INTERVAL start-interval EVENT TYPE ┐
┌ INTERNAL ─ ┐
┌ EXTERNAL ─ ┐

┌ EVENT ecb ─ ┐
┌ EVENT NAME ecb-id ┐

DELETE QUEUE

▶▶ DELETE QUEUE ┌ ID queue-id ┐ ┌ CURRENT ◀ ┐ .

┌ ALL ─ ┐

DELETE SCRATCH

▶▶ DELETE SCRATCH

┌ AREA ID scratch-area-id ┐

┌ CURRENT ◀ ┐
┌ FIRST ─ ┐
┌ LAST ─ ┐
┌ NEXT ─ ┐
┌ PRIOR ─ ┐
┌ ALL ─ ┐
┌ RECORD ID scratch-record-id ┐

┌ RETURN RECORD ID into return-scratch-record ┐ .

DELETE TABLE

```

▶▶— DELETE TABLE from 01-level-program-location .
    [ DICTNODE nodename ] [ DICTNAME dictionary-name ]
▶▶ [ LOADLIB library-name ] .

```

DEQUEUE

```

▶▶— DEQUEUE [ ALL ] [ NAME resource-id LENGTH resource-id-length ] .

```

DISCONNECT

```

▶▶— DISCONNECT record-name FROM set-name .

```

END LINE TERMINAL SESSION

```

▶▶— END LINE TERMINAL session .

```

END TRANSACTION STATISTICS

```

▶▶— END TRANSACTION STATISTICS
    [ WRITE ] [ NOWRITE ] [ INTO return-stat-data-location ] .

```

ENDPAGE

```

▶▶— ENDPAGE session .

```

ENQUEUE

```

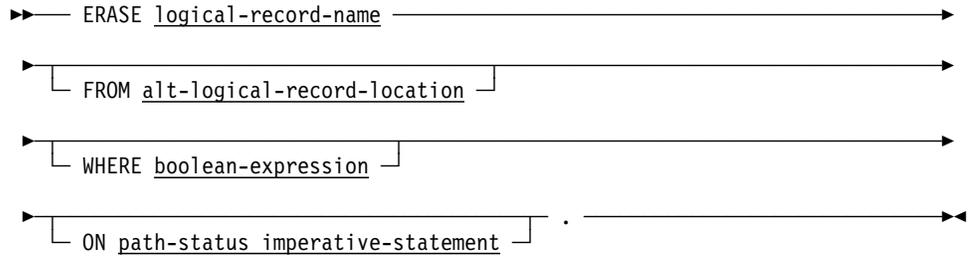
▶▶— ENQUEUE [ WAIT ] [ NOWAIT ] [ TEST ]
    [ NAME resource-id LENGTH resource-length ] [ EXCLUSIVE ] [ SHARED ] .

```

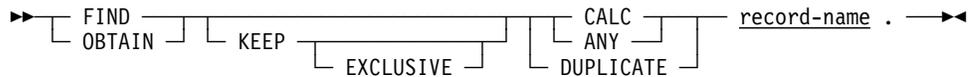
ERASE



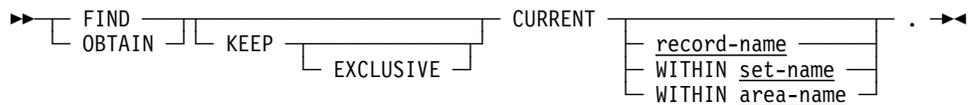
ERASE (LRF)



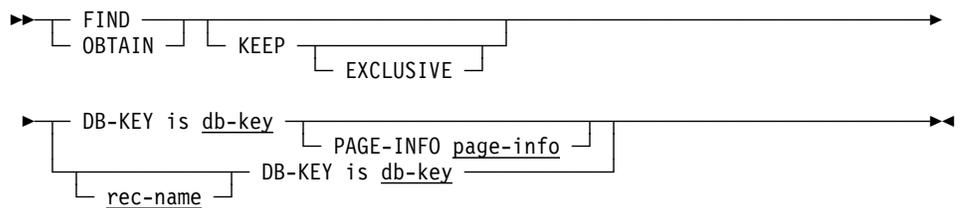
FIND/OBTAIN CALC/DUPLICATE



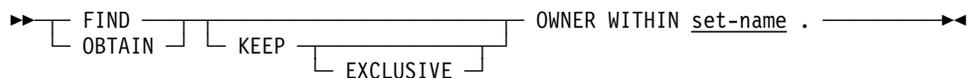
FIND/OBTAIN CURRENT



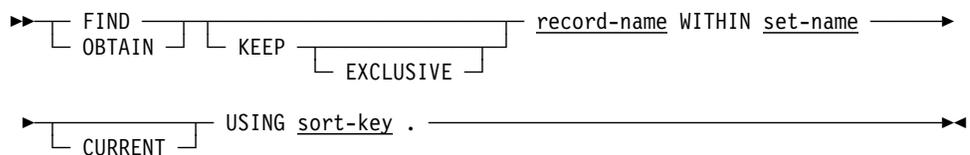
FIND/OBTAIN DB-KEY

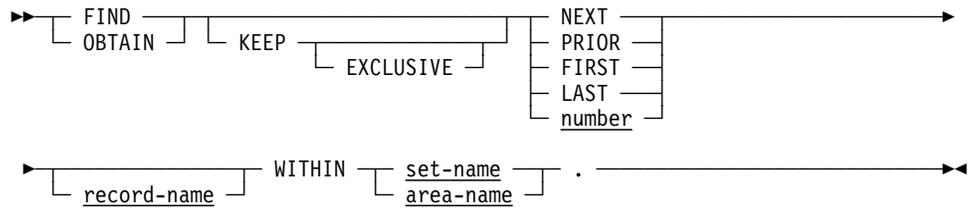


FIND/OBTAIN OWNER



FIND/OBTAIN WITHIN SET USING SORT KEY

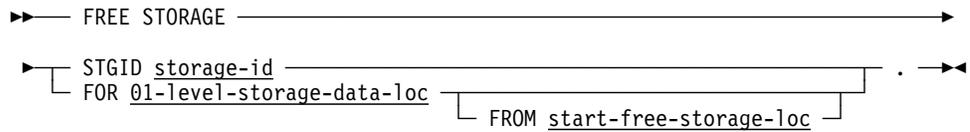


FIND/OBTAIN WITHIN SET/AREA

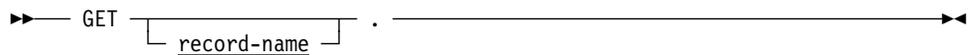
FINISH



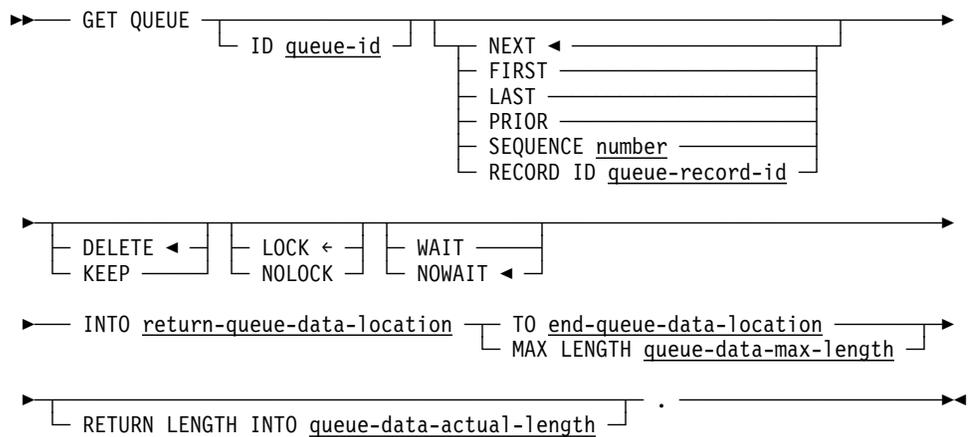
FREE STORAGE



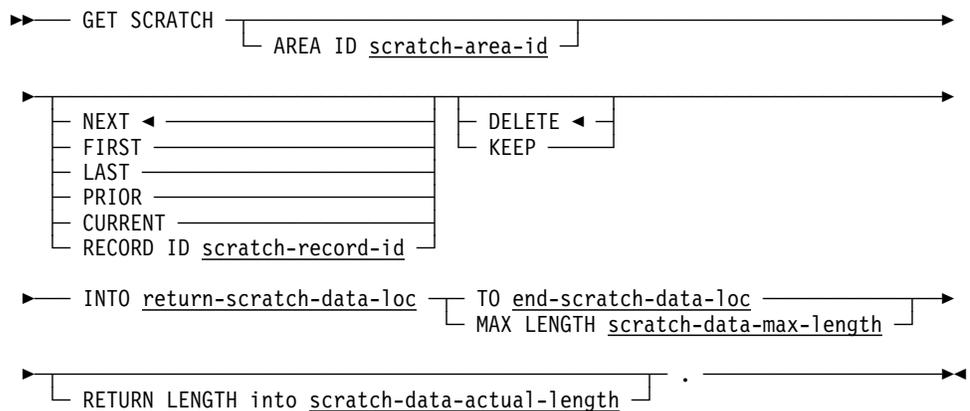
GET

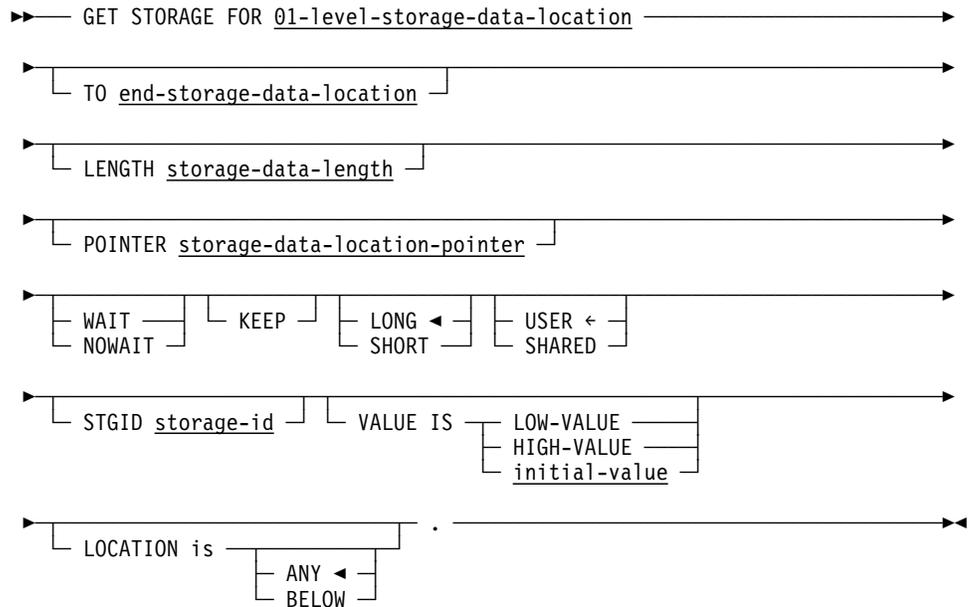
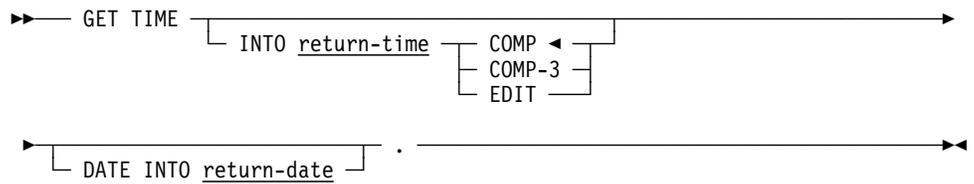


GET QUEUE

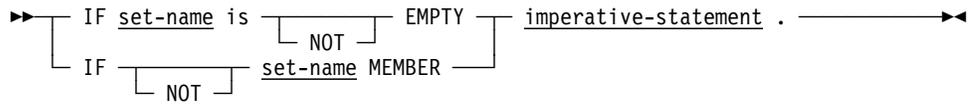


GET SCRATCH

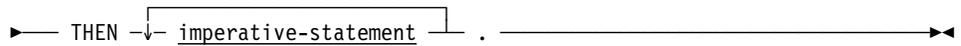
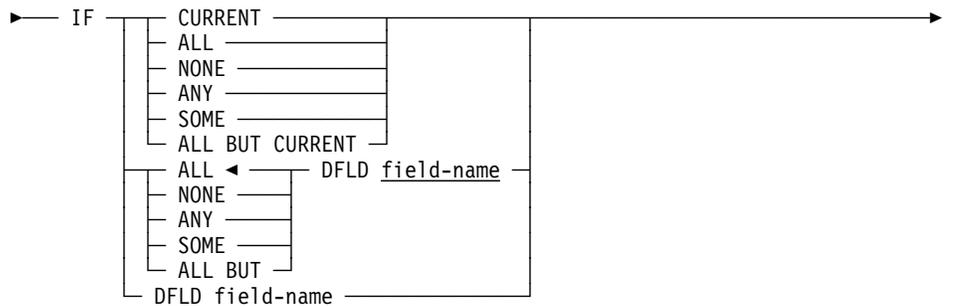
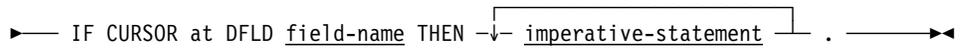
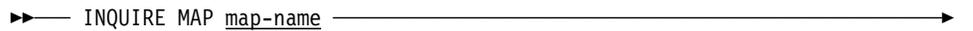
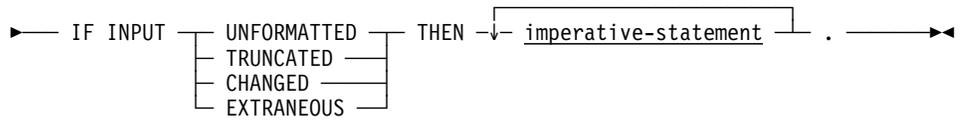
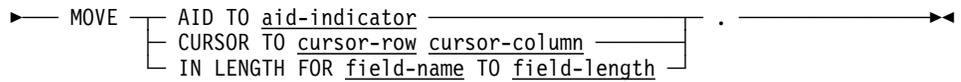


GET STORAGE**GET TIME**

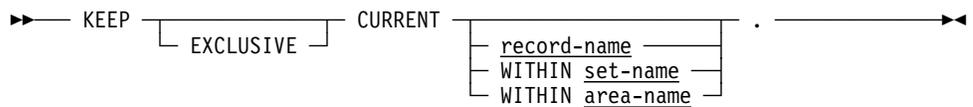
IF



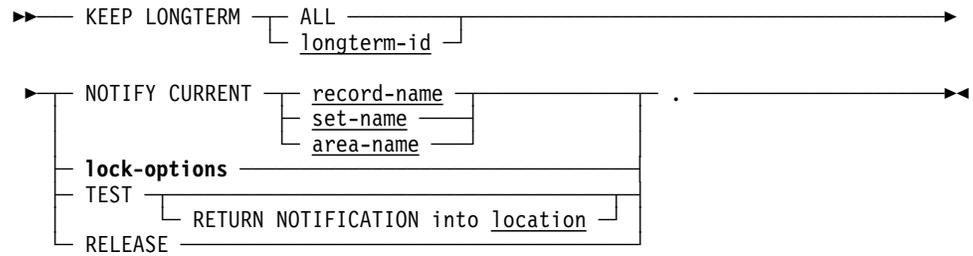
INQUIRE MAP



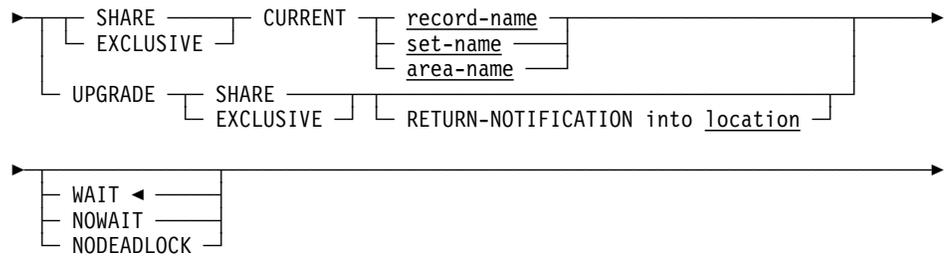
KEEP CURRENT



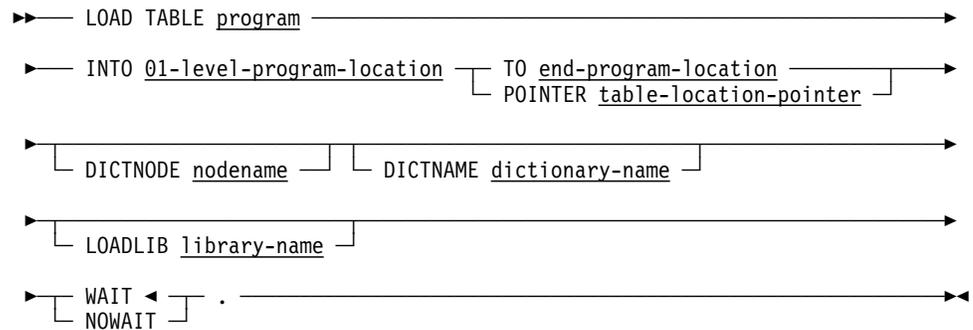
KEEP LONGTERM



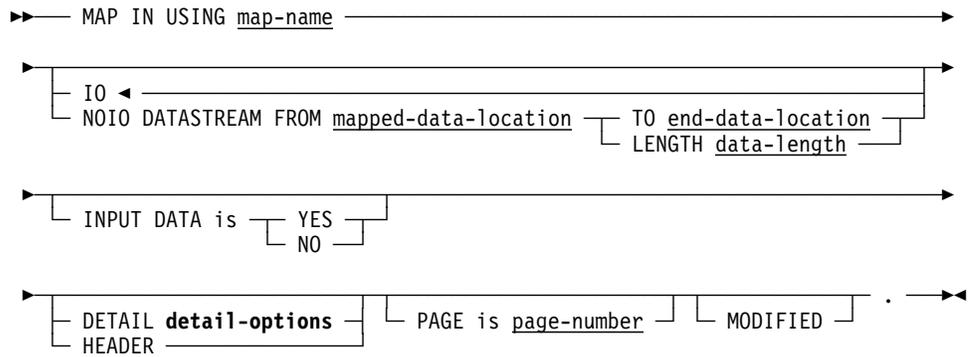
Expansion of lock-options



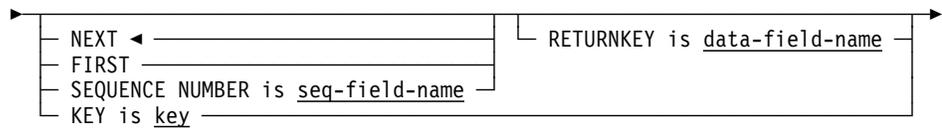
LOAD TABLE



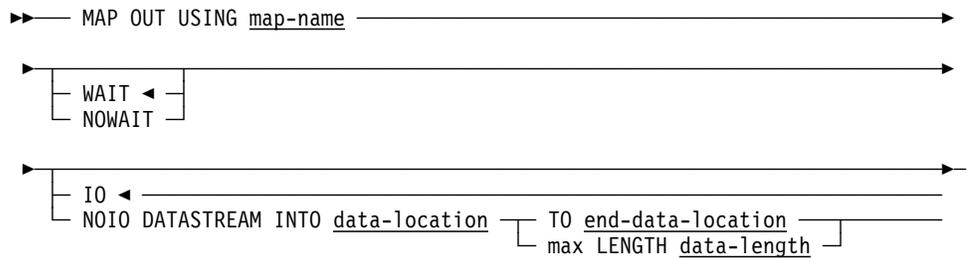
MAP IN

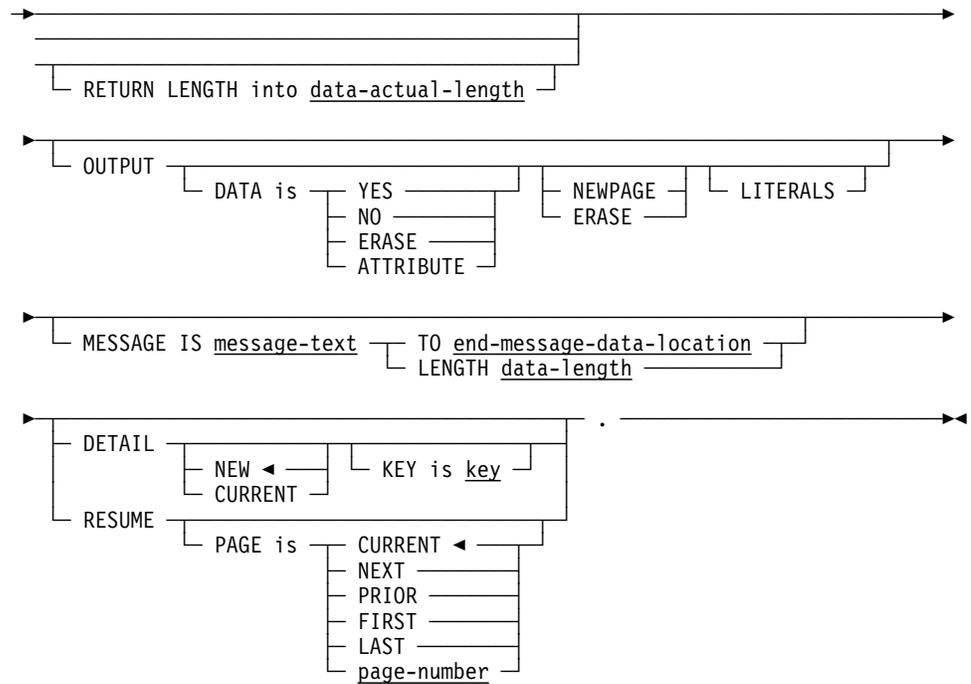
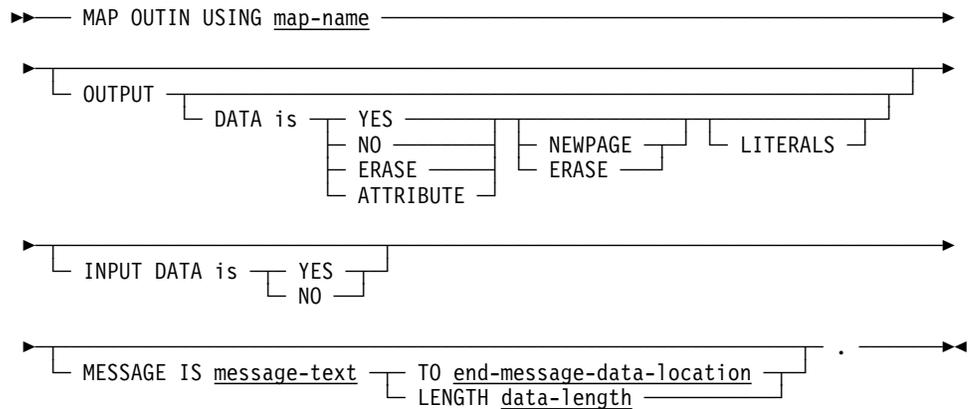
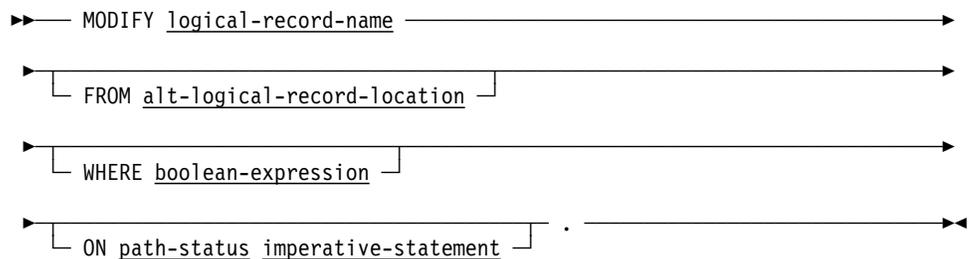


Expansion of detail-options

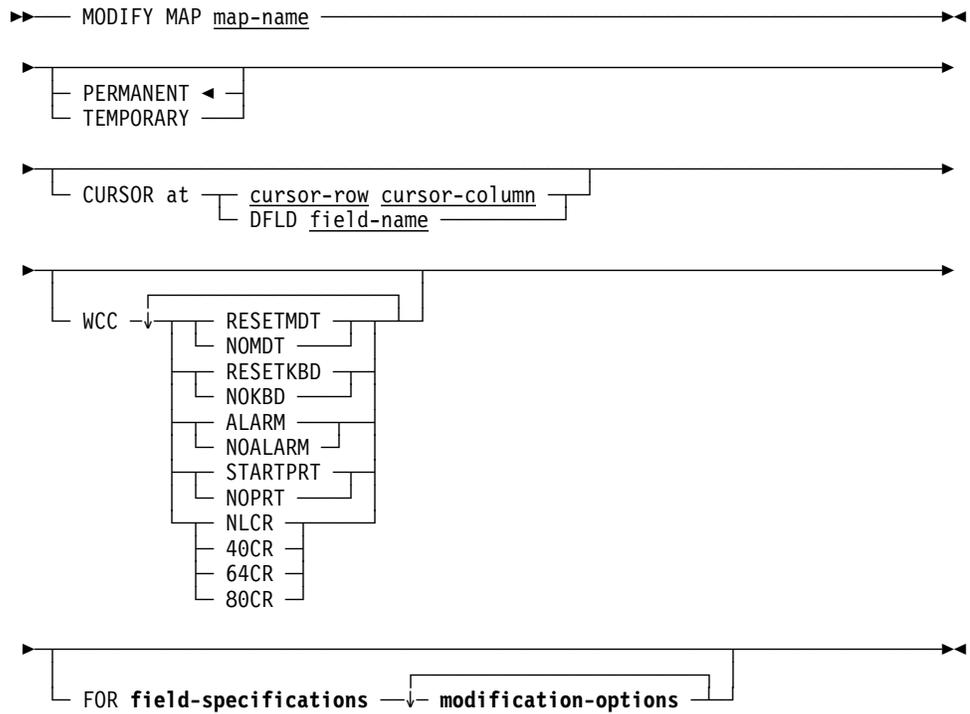


MAP OUT

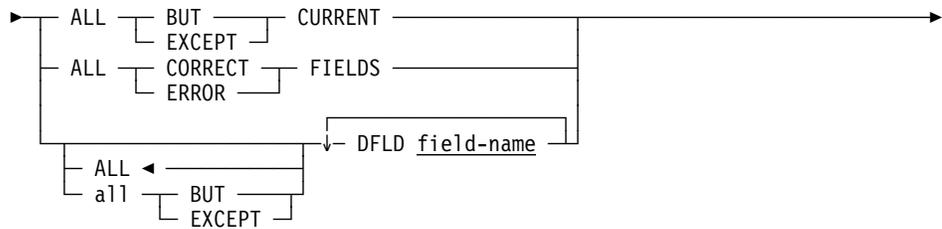


**MAP OUTIN****MODIFY****MODIFY (LRF)**

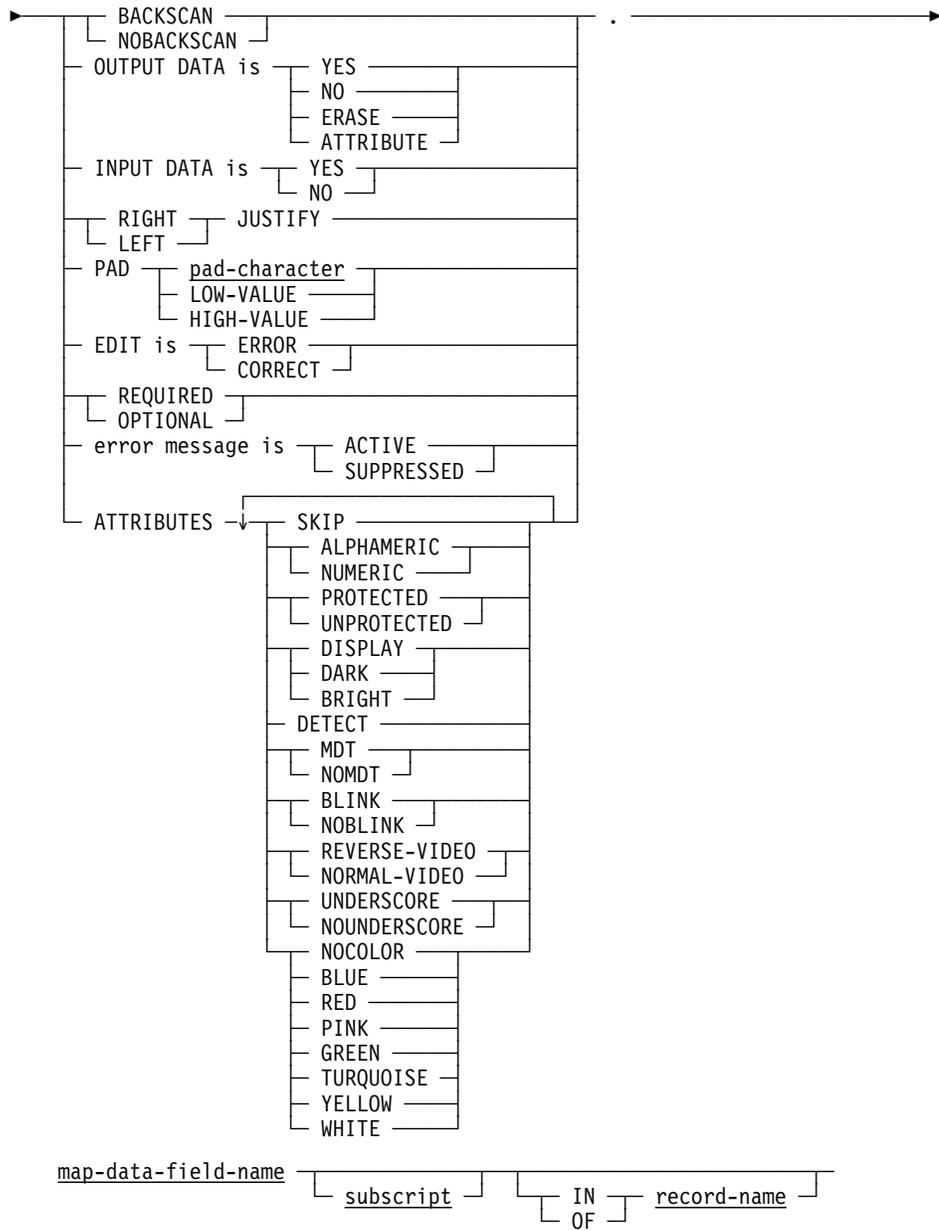
MODIFY MAP



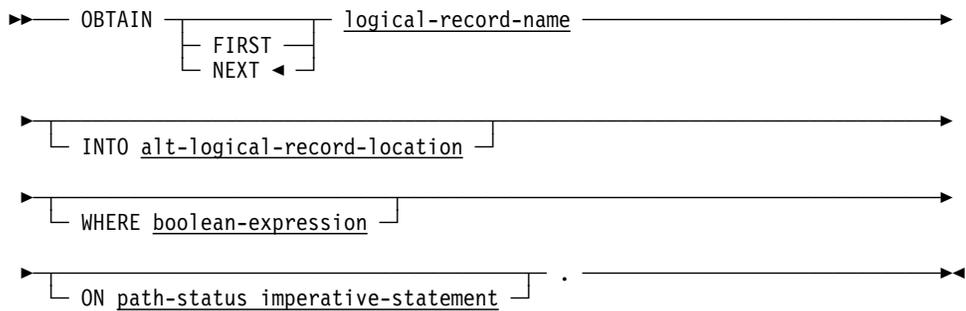
Expansion of field-specifications



Expansion of modification-options



OBTAIN (LRF)



POST

```

▶▶ POST [ EVENT ecb ] [ EVENT NAME ecb-id ] [ CLEAR ] .

```

PUT QUEUE

```

▶▶ PUT QUEUE [ ID queue-id ] [ FIRST ] [ LAST ]
▶ FROM queue-data-location [ TO end-queue-data-location ] [ LENGTH queue-data-length ]
▶ [ RETURN RECORD ID INTO return-queue-record-id ]
▶ [ RETENTION queue-retention-period ] .

```

PUT SCRATCH

```

▶▶ PUT SCRATCH [ AREA ID scratch-area-id ]
▶ FROM scratch-data-location [ TO end-scratch-data-location ] [ LENGTH scratch-data-length ]
▶ [ RECORD ID scratch-record-id ] [ REPLACE ]
▶ [ RETURN RECORD ID into return-scratch-record-id ] .

```

READ LINE FROM TERMINAL

```

▶▶ READ LINE from TERMINAL [ ECHO ] [ NOBACKPAGE ]
▶ INTO input-data-location [ TO end-input-data-location ] [ MAX LENGTH input-data-max-length ]
▶ [ RETURN LENGTH into input-data-actual-length ] .

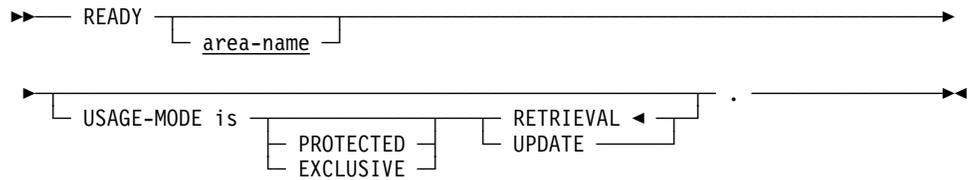
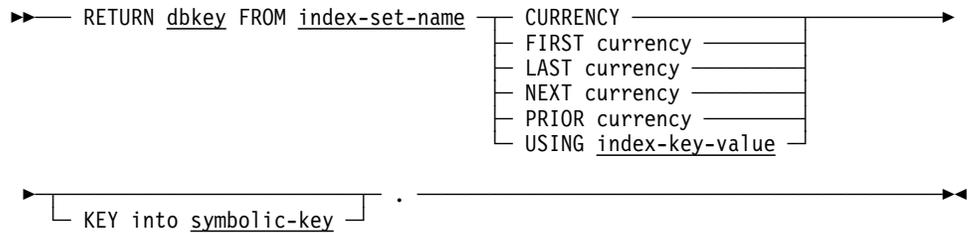
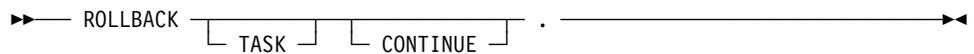
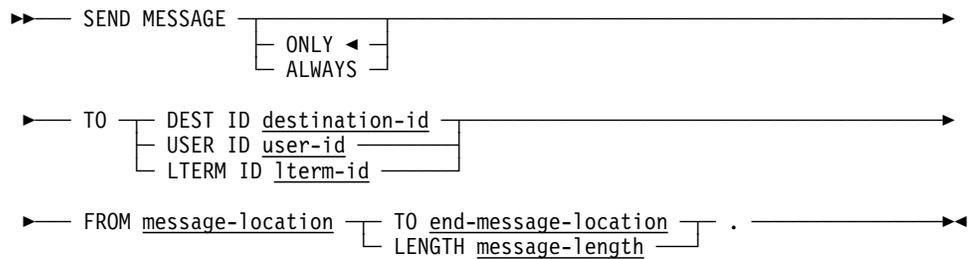
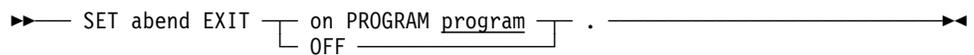
```

READ TERMINAL

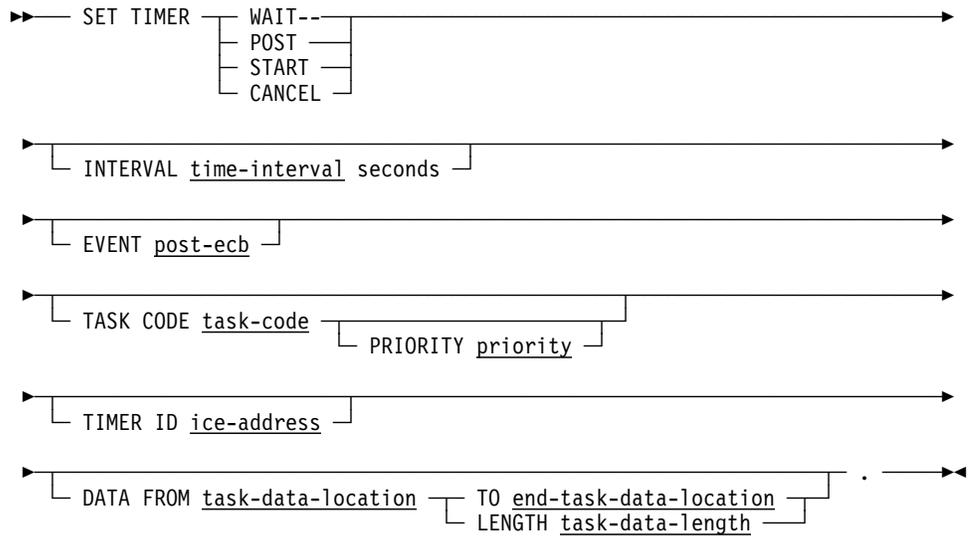
```

▶▶ READ TERMINAL
▶ [ WAIT ] [ NOWAIT ] [ BUFFER ] [ MODIFIED FROM POSITION screen-position ]
▶ [ GET STORAGE ]
▶ INTO input-data-location [ TO end-input-data-location ] [ MAX LENGTH input-data-max-length ]
▶ [ RETURN LENGTH into input-data-actual-length ] .

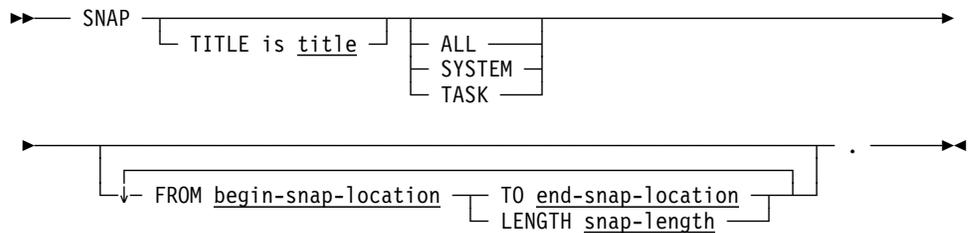
```

READY**RETURN****ROLLBACK****SEND MESSAGE****SET ABEND EXIT**

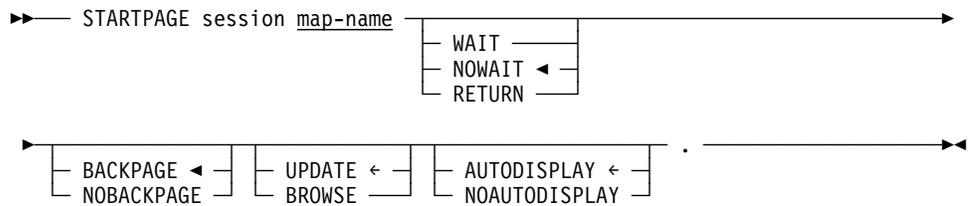
SET TIMER



SNAP



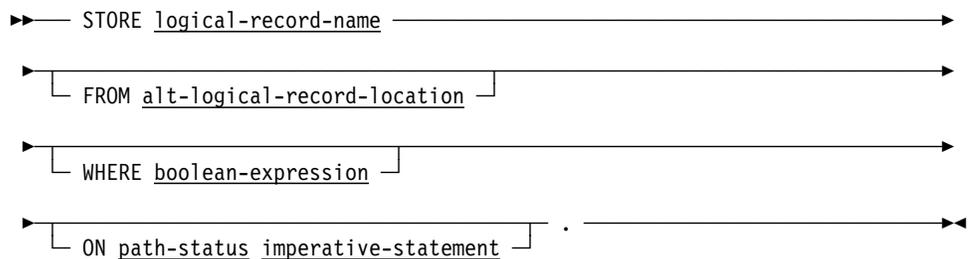
STARTPAGE

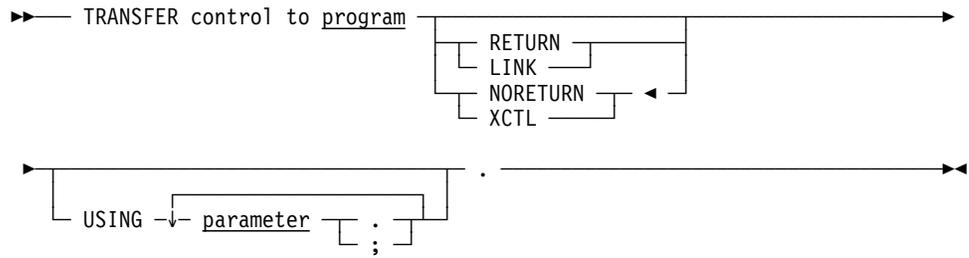
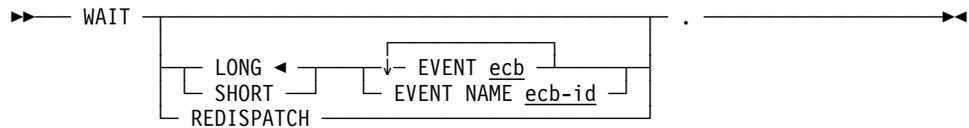
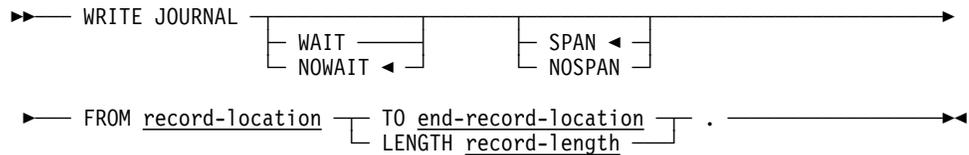
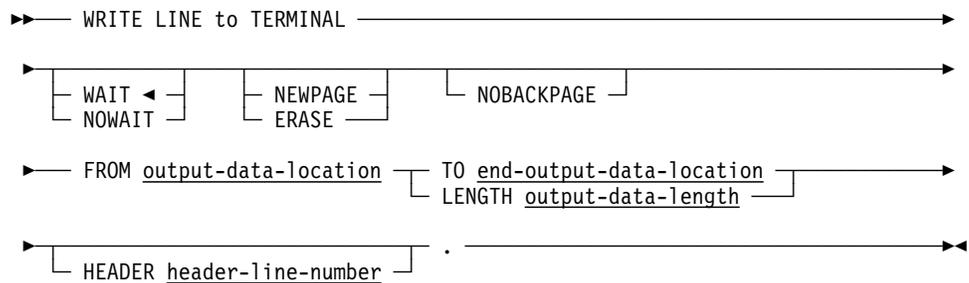


STORE

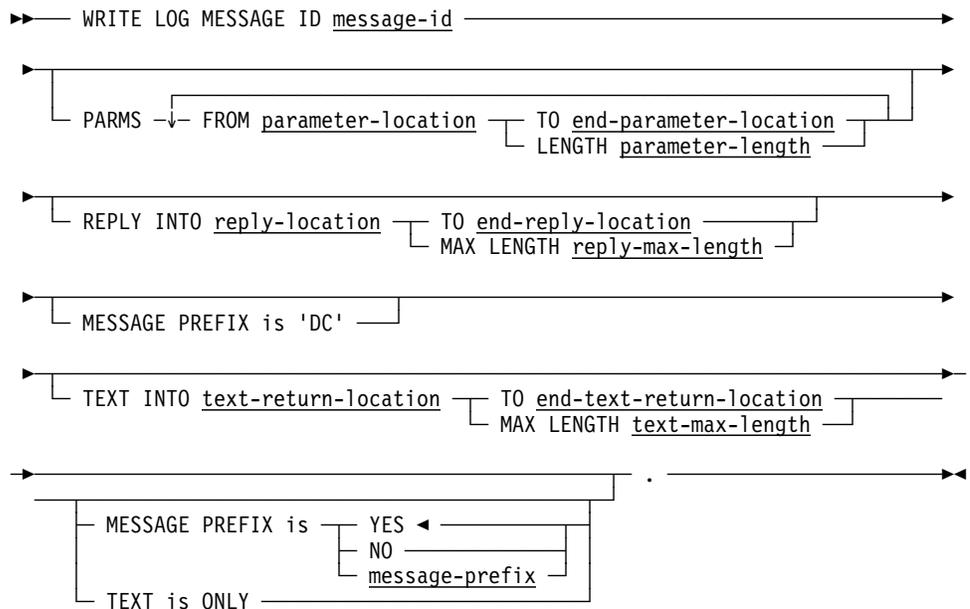


STORE (LRF)

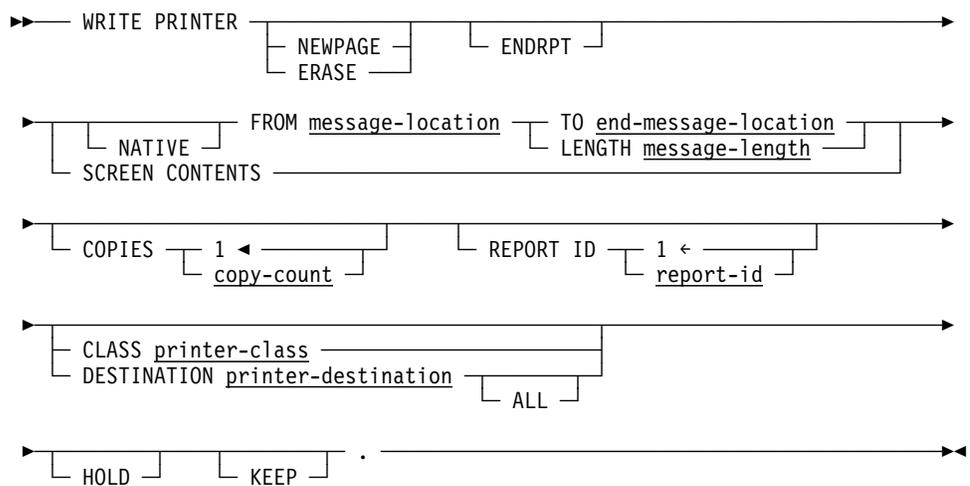


TRANSFER CONTROL**WAIT****WRITE JOURNAL****WRITE LINE TO TERMINAL**

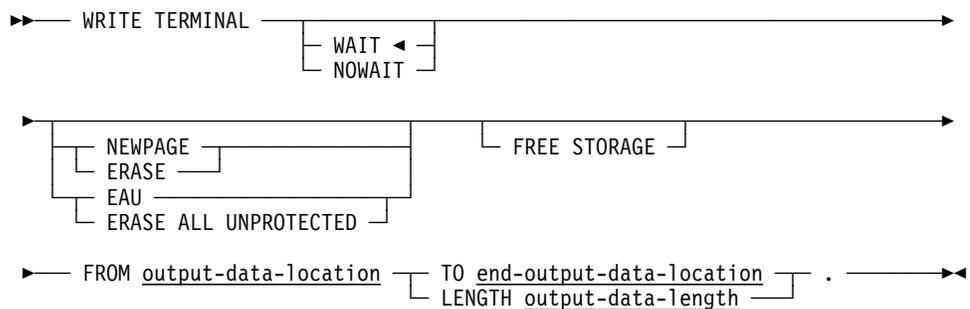
WRITE LOG



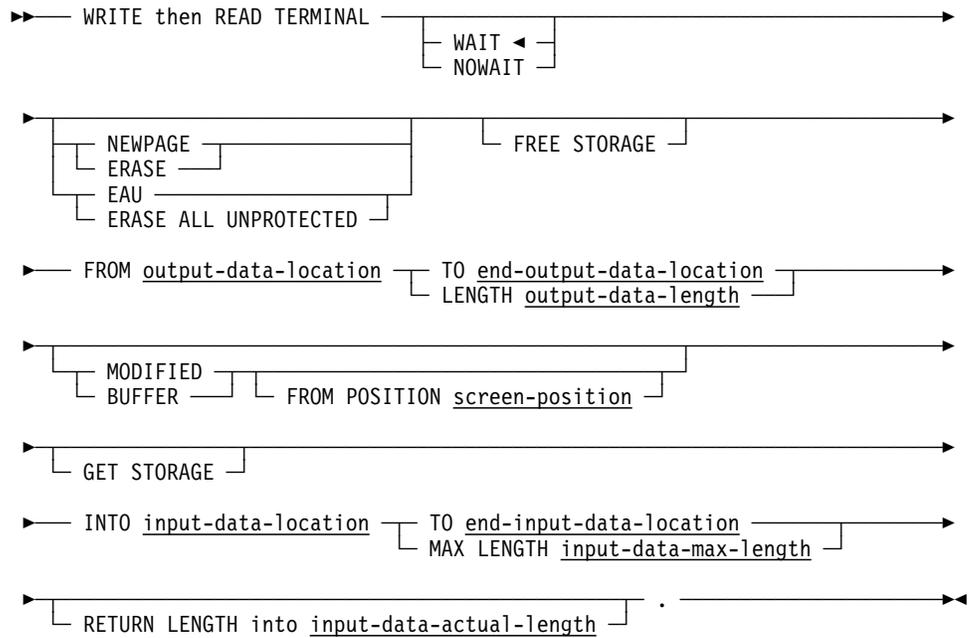
WRITE PRINTER



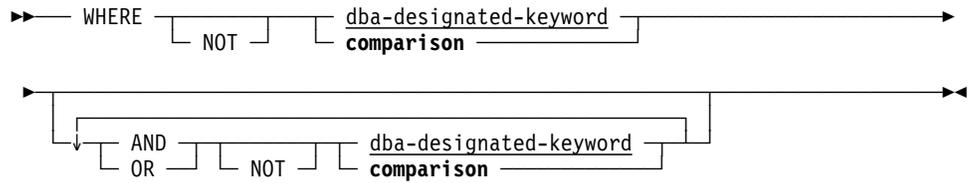
WRITE TERMINAL



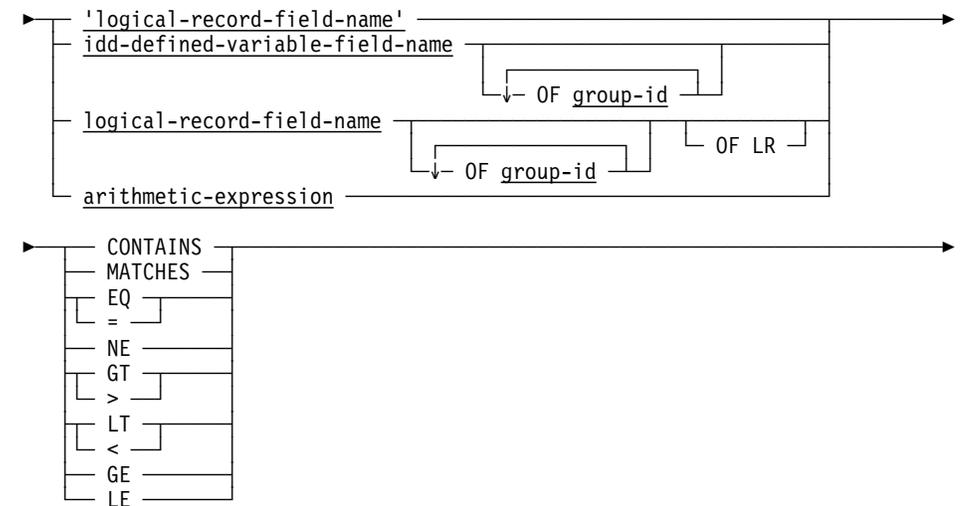
WRITE THEN READ TERMINAL



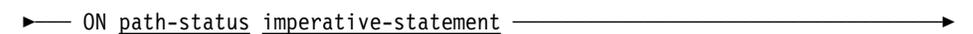
Logical-record clauses (WHERE)



Expansion of comparison

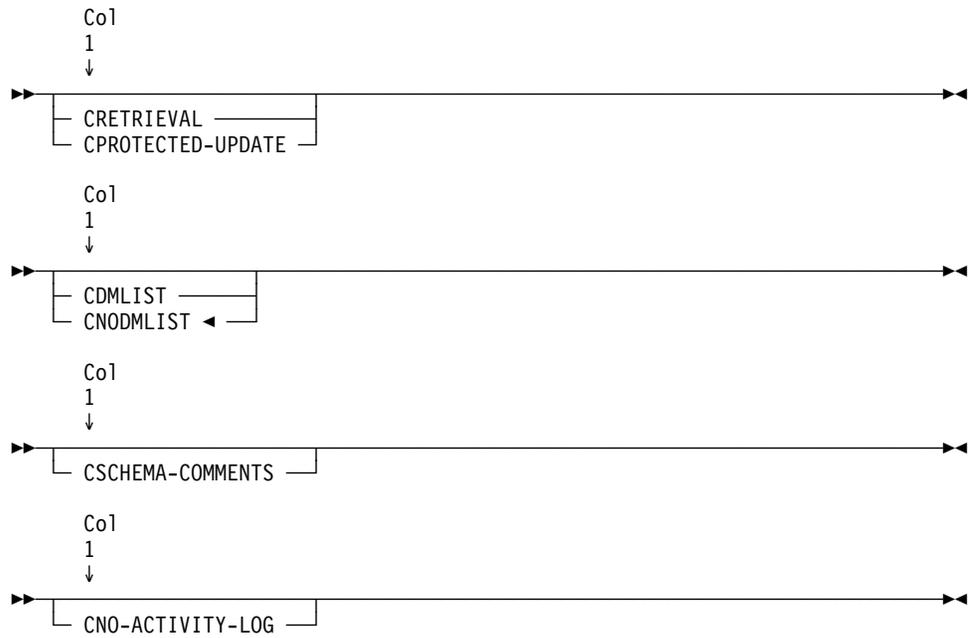


Logical-record clauses (ON)

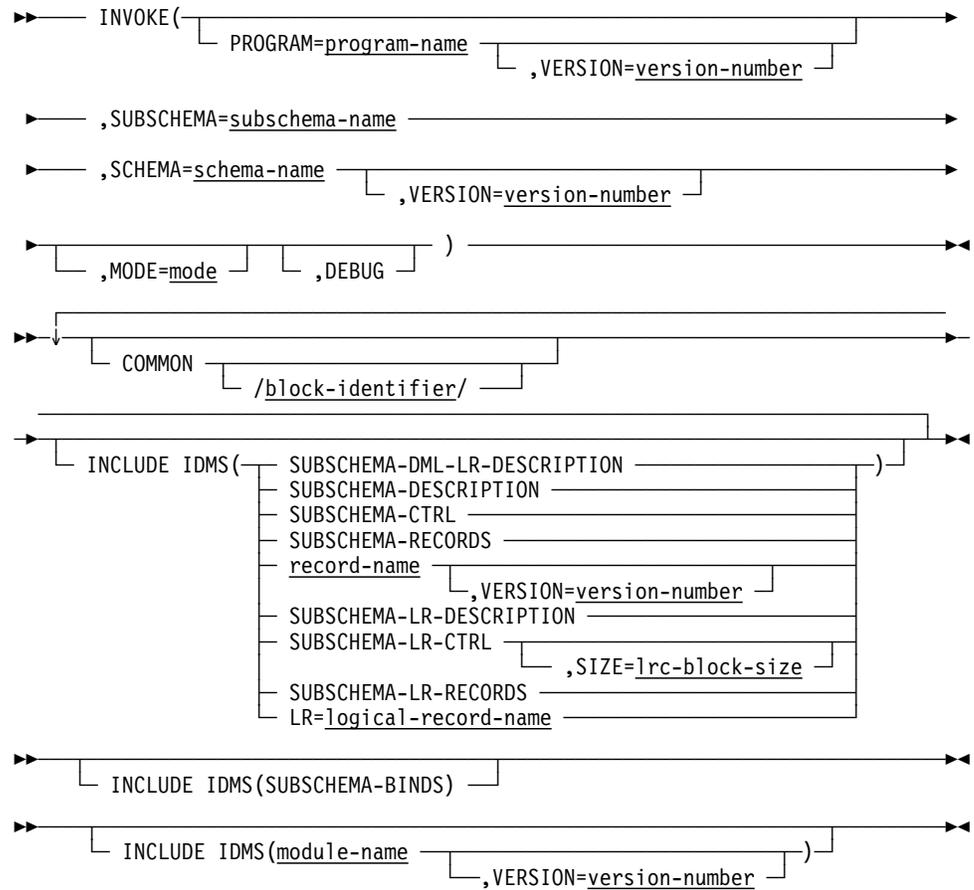


Chapter 2. FORTRAN

2.1 Compiler options statements



2.2 Compiler-directive statements



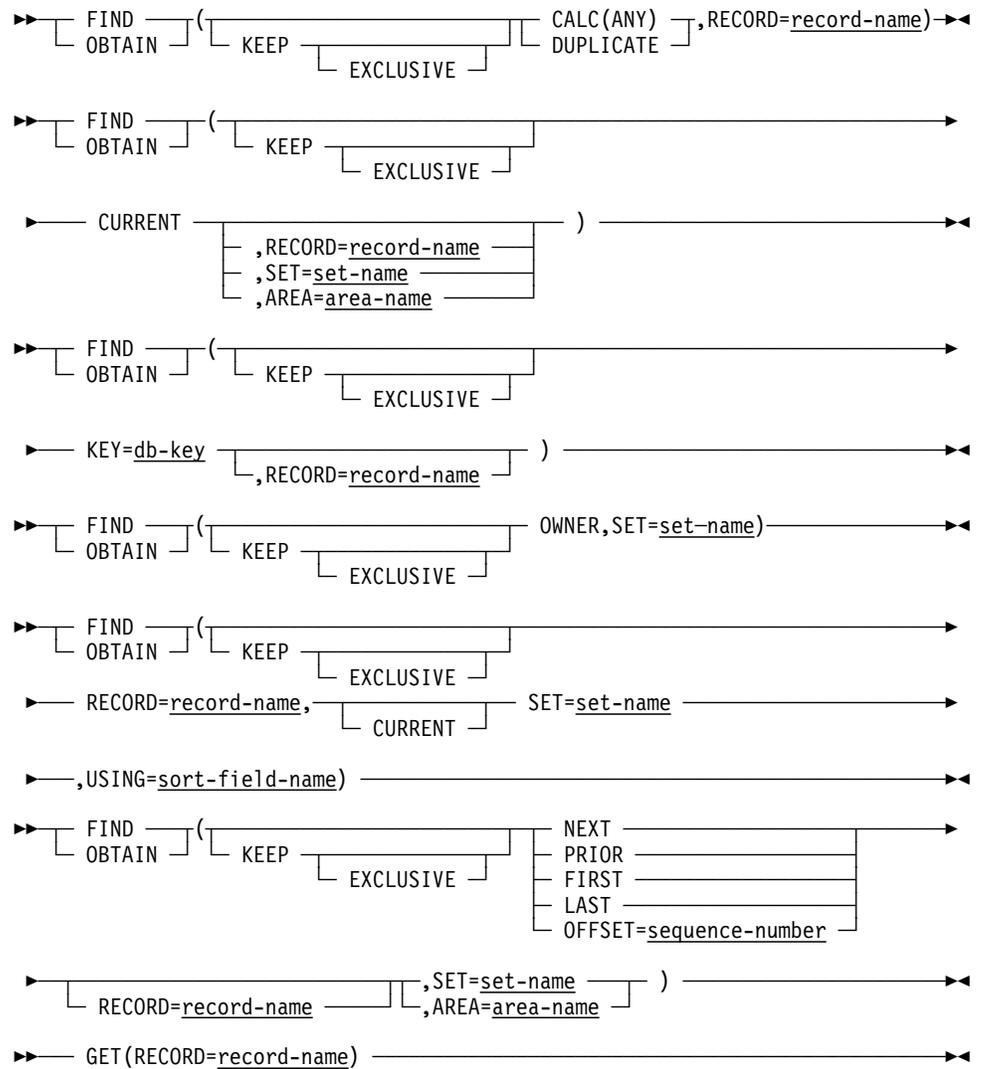
2.3 Control statements

```

▶▶ BIND(PROCEDURE=procedure-name,procedure-control-location)
▶▶ BIND(RECORD=record-name [,record-location ])
▶▶ BIND(RUN-UNIT)
▶▶ [ ,SUBSCHEMA=subschema-name [,SUBSCHEMA-CTRL=subschema-cntrl-loc ] ]
    [ ,DBNODE=nodename ]
    [ ,DBNAME=database-name ]
    [ ,DICTNODE=nodename ]
    [ ,DICTNAME=dictionary-name ]
▶▶ IF( [.NOT. ] EMPTY(set-name) ) imperative-statement
▶▶ IF( [.NOT. ] MEMBER(set-name) ) imperative-statement
▶▶ KEEP( [EXCLUSIVE ] CURRENT [,RECORD=record-name
    [,SET=set-name
    [,AREA=area-name ] )
▶▶ COMMIT [(ALL) ]
▶▶ FINISH [(ALL) ]
▶▶ READY( [ALL ] [AREA=area-name ] [PROTECTED
    [EXCLUSIVE ] [RETRIEVAL
    [UPDATE ] )
▶▶ ROLLBACK [(CONTINUE) ]

```

2.4 Retrieval statements



2.5 Modification statements

▶▶— CONNECT (RECORD=record-name,SET=set-name) —————▶▶

▶▶— DISCONNECT (RECORD=record-name,SET=set-name) —————▶▶

▶▶— MODIFY (RECORD=record-name) —————▶▶

▶▶— ERASE (RECORD=record-name

) —————▶▶

▶▶— STORE (RECORD=record-name) —————▶▶

2.6 Save statements

```

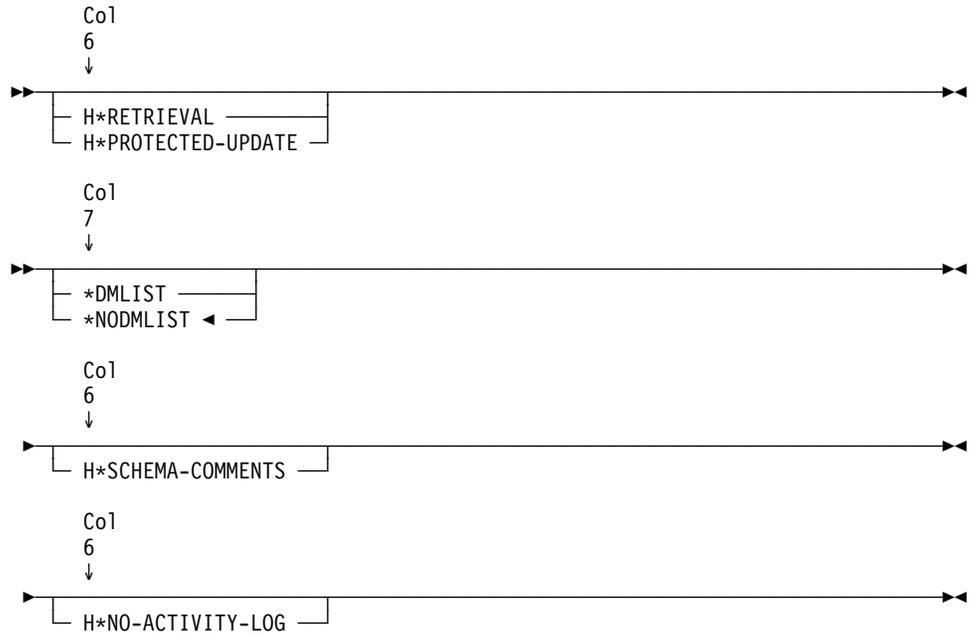
▶▶ ACCEPT (BIND=bind-address, RECORD=record-name) ▶▶
▶▶ ACCEPT (IDMS-STATISTICS=db-statistics) ▶▶
▶▶ ACCEPT (CURRENCY=db-key ) ▶▶
▶▶ (
    , RECORD=record-name
    , SET=set-name
    , AREA=area-name
) ▶▶
▶▶ ACCEPT (CURRENCY=db-key, SET=set-name
    , NEXT
    , PRIOR
    , OWNER
) ▶▶
▶▶ ACCEPT (PROCEDURE=procedure-name, procedure-control-location) ▶▶
▶▶ RETURN (CURRENCY=db-key, SET=index-set-name
    , FIRST
    , LAST
    , NEXT
    , PRIOR
) ▶▶
▶▶ RETURN (CURRENCY=db-key, SET=index-set-name, USING=index-key-value) ▶▶

```

2.7 Logical Record Facility statements



2.8 The DMLR Precompiler



Chapter 3. PL/1 syntax

3.1 PL/1 Precompiler options

Ready DDL/DML area

▶▶ [/*RETRIEVAL*/ —————▶▶
 [/*PROTECTED_UPDATE*/]

Begin this syntax in column 2.

PL/1 character set usage

▶▶ * PROCESS CHARSET (48); —————▶▶

Begin this syntax in column 2.

Comment generation

▶▶ /*SCHEMA_COMMENTS*/ —————▶▶

Begin this syntax in column 2.

List generation

▶▶ [/*NODMLIST*/ ◀ —————▶▶
 [/*DMLIST*/ —————▶▶

Begin this syntax in column 2.

Log suppression

▶▶ /*NO_ACTIVITY_LOG*/ —————▶▶

3.2 PL/1 Precompiler-Directive statements

Begin this syntax in column 2.

DECLARE IDMS

```
▶▶ [ DECLARE ] IDMS ENTRY OPTIONS (INTER, ASSEMBLER); ▶▶  
   [ DCL ]
```

DECLARE IDMSPLI

```
▶▶ [ DECLARE ] IDMSPLI ENTRY OPTIONS (INTER, ASSEMBLER); ▶▶  
   [ DCL ]
```

DECLARE IDMSDCP

```
▶▶ [ DECLARE ] IDMSDCP ENTRY OPTIONS (INTER, ASSEMBLER); ▶▶  
   [ DCL ]
```

DECLARE SQLXQ1

```
▶▶ [ DECLARE ] SQLXQ1 ENTRY OPTIONS (INTER, ASSEMBLER); ▶▶  
   [ DCL ]
```

DECLARE ADDR BUILTIN

```
▶▶ [ DECLARE ] ADDR BUILTIN; ▶▶  
   [ DCL ]
```

DECLARE ABORT

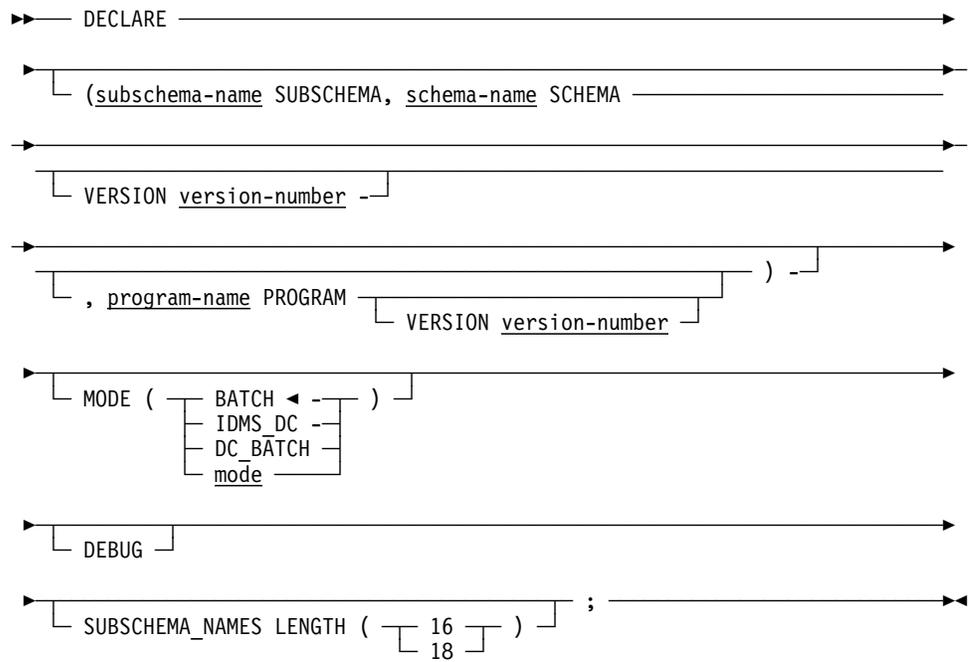
```
▶▶ [ DECLARE ] ABORT ENTRY OPTIONS (INTER, ASSEMBLER); ▶▶  
   [ DCL ]
```

DECLARE IDMSP

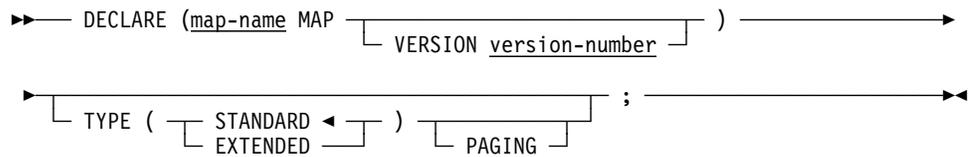
```
▶▶ [ DECLARE ] IDMSP ENTRY; ▶▶  
   [ DCL ]
```

DECLARE SUBSCHEMA

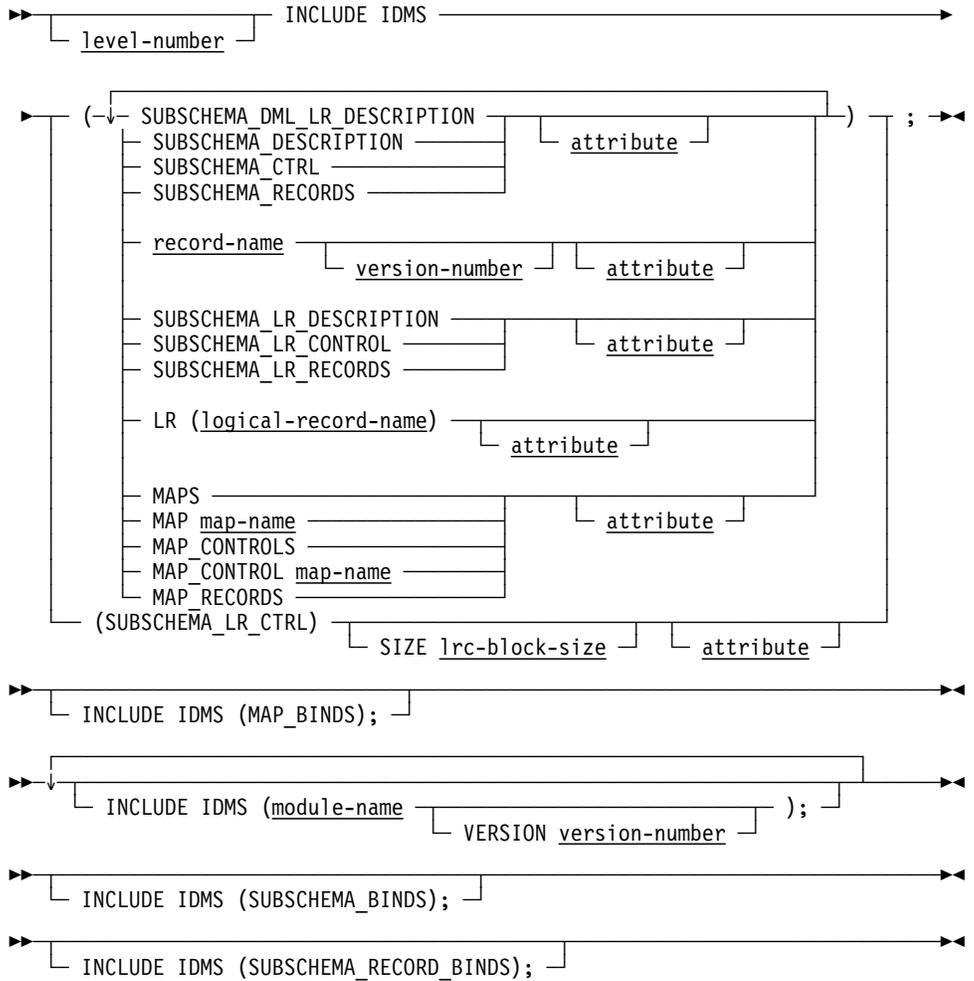
3.2 PL/1 Precompiler-Directive statements



DECLARE MAP



INCLUDE IDMS statements



3.3 PL/1 DML statements

ABEND (DC/UCF)

```

▶▶ ABEND CODE (abend-code) [ NODUMP | DUMP ] ;

```

ACCEPT (DC/UCF)

```

▶▶ ACCEPT [ TASK CODE | TASK ID | LTERM ID | PTERM ID |
  SYSVERSION | USER ID | SCREENSIZE ] INTO (return-location);

```

ACCEPT BIND RECORD

```

▶▶ ACCEPT BIND RECORD (record-name) INTO (bind-address);

```

ACCEPT DBKEY FROM CURRENCY

```

▶▶ ACCEPT CURRENCY [ RECORD (record-name) | SET (set name) |
  AREA (area-name) ] INTO (db-key-field);

```

ACCEPT DBKEY RELATIVE TO CURRENCY

```

▶▶ ACCEPT CURRENCY SET (set name) [ NEXT | PRIOR | OWNER ] INTO (db-key-field);

```

ACCEPT IDMS STATISTICS

```

▶▶ ACCEPT IDMS_STATISTICS INTO (db-statistics-field);

```

ACCEPT PAGE_INFO

```

▶▶ ACCEPT PAGE_INFO RECORD (record-name) INTO (page-info-location);

```

ACCEPT PROCEDURE CONTROL LOCATION

```

▶▶ ACCEPT PROCEDURE (procedure-name) INTO (procedure-control-location);

```

ACCEPT TRANSACTION STATISTICS (DC/UCF)

```

▶▶ ACCEPT TRANSACTION STATISTICS [ WRITE | NOWRITE ]
  INTO (return-stat-data-location);

```

ATTACH (DC/UCF)

```

▶▶ ATTACH TASK CODE (task-code) [ PRIORITY (priority) ] [ WAIT | NOWAIT ] ;

```

BIND MAP (DC/UCF)

```

▶— BIND MAP (map-name) ┌ RECORD (record-name) ───────────────────▶
                          └──────────────────────────────────────────┘
▶──────────────────────────────────────────┘ ; ───────────────────▶
┌ TO ┌ NULL ───────────────────┘
└───┘ (record-location) ───┘

```

BIND PROCEDURE

```

▶— BIND PROCEDURE (procedure-name) TO (procedure-control-location); ───▶

```

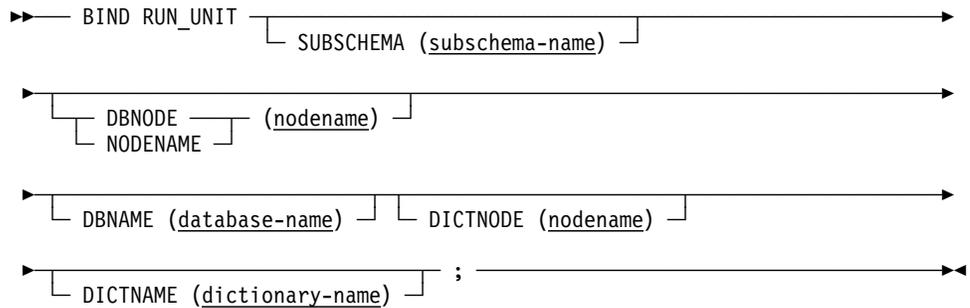
BIND RECORD

```

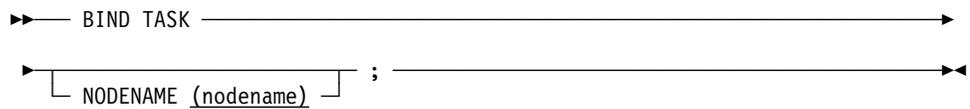
▶— BIND RECORD (record-name) ┌ TO (record-location) ───┘ ; ───▶

```

BIND RUN_UNIT



BIND TASK (DC/UCF)



BIND TRANSACTION STATISTICS (DC/UCF)



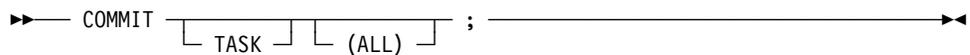
CHANGE PRIORITY (DC/UCF)



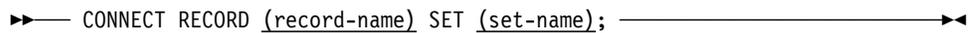
CHECK TERMINAL (DC/UCF)



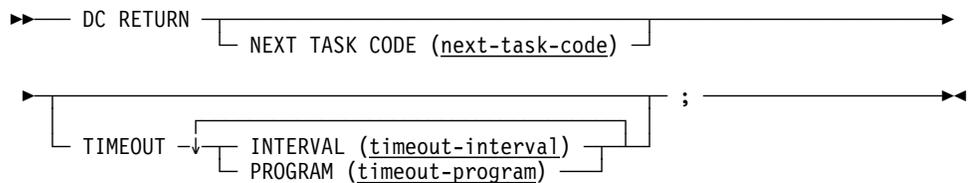
COMMIT



CONNECT



DC RETURN (DC/UCF)



DELETE QUEUE (DC/UCF)



DELETE SCRATCH (DC/UCF)

```

▶▶ DELETE SCRATCH- [ AREA ID (scratch-area-id ) ]
|
| CURRENT
| FIRST
| LAST
| NEXT
| PRIOR
| ALL
| RECORD ID (scratch-record-id)
|
▶▶ [ RETURN RECORD ID INTO (return-scratch-record-id) ] ;

```

DELETE TABLE (DC/UCF)

```

▶▶ DELETE TABLE FROM (table-location-pointer);

```

DEQUEUE (DC/UCF)

```

▶▶ DEQUEUE [ ALL ] ;
|
| NAME (resource-id) LENGTH (resource-id-length)
|

```

DISCONNECT

```

▶▶ DISCONNECT RECORD (record-name) SET (set-name);

```

END LINE TERMINAL SESSION (DC/UCF)

```

▶▶ END LINE TERMINAL session ;

```

END TRANSACTION STATISTICS (DC/UCF)

```

▶▶ END TRANSACTION STATISTICS [ WRITE ]
| [ NOWRITE ]
|
▶▶ [ INTO (return-stat-data-location) ] ;

```

ENDPAGE (DC/UCF)

```

▶▶ ENDPAGE session ;

```

ENQUEUE (DC/UCF)

```

▶▶ ENQUEUE [ WAIT ]
| [ NOWAIT ]
| [ TEST ]
|
▶▶ NAME (resource-id) LENGTH (resource-id-length) [ EXCLUSIVE ]
| [ SHARED ] ;

```

ERASE

►► ERASE RECORD (record-name)

| |
|-----------|
| PERMANENT |
| SELECTIVE |
| ALL |

 ;

ERASE (LRF)

►► ERASE RECORD (logical-record-name)
 FROM (alt-logical-record) WHERE (boolean-expression)
 ON LR_STATUS (path-status) imperative-statement ;

FIND/OBTAIN

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

| |
|-----------|
| CALC |
| ANY |
| DUPLICATE |

 RECORD (record-name);

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

 CURRENT RECORD (record-name);

►►

| |
|-------------------------------|
| RECORD (<u>record-name</u>) |
| SET (<u>set-name</u>) |
| AREA (<u>area-name</u>) |

 ;

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

►►

| |
|----------------------------------|
| DBKEY (<u>db-key-v</u>) |
| PAGE_INFO (<u>page-info-v</u>) |
| DBKEY (<u>db-key-v</u>) |

 RECORD (record-name);

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

 OWNER SET (set-name);

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

 RECORD (record-name);

► SET (set-name)

| |
|---------|
| CURRENT |
|---------|

 USING (sort-field-name);

►►

| |
|--------|
| FIND |
| OBTAIN |

 KEEP

| |
|-----------|
| EXCLUSIVE |
|-----------|

| |
|--------------------------------|
| NEXT |
| FIRST |
| LAST |
| PRIOR |
| NTH (<u>sequence-number</u>) |

►

| |
|-------------------------------|
| RECORD (<u>record-name</u>) |
| SET (<u>set-name</u>) |
| AREA (<u>area-name</u>) |

 ;

FINISH

```

▶▶ FINISH [ TASK ] ;

```

FREE STORAGE (DC/UCF)

```

▶▶ FREE STORAGE
▶ STGID (storage-id)
  FOR (storage-location)
    FROM (start-free-storage-location) ;

```

GET

```

▶▶ GET [ RECORD (record-name) ] ;

```

GET QUEUE (DC/UCF)

```

▶▶ GET QUEUE [ ID (queue-id) ]
  NEXT ◀
  FIRST
  LAST
  PRIOR
  SEQUENCE (sequence-number)
  RECORD ID (queue-record-id)
  DELETE ◀
  KEEP
  LOCK ◀
  NOLOCK
  WAIT
  NOWAIT ◀
▶ INTO (return-queue-data-location)
▶ TO (end-queue-data-location)
  MAX LENGTH (queue-data-max-length)
▶ RETURN LENGTH INTO (queue-data-actual-length) ;

```

GET SCRATCH (DC/UCF)

```

▶▶ GET SCRATCH [ AREA ID (scratch-area-id) ]
  NEXT ◀
  FIRST
  LAST
  CURRENT
  PRIOR
  RECORD ID (scratch-record-id)
  DELETE ◀
  KEEP
▶ INTO (return-scratch-data-location)
▶ TO (end-scratch-data-location)
  MAX LENGTH (scratch-data-max-length)
▶ RETURN LENGTH INTO (scratch-data-actual-length) ;

```

GET STORAGE (DC/UCF)

```

▶▶ GET STORAGE FOR (storage-data-location)
  ▶ TO (end-storage-data-location)
    LENGTH (storage-data-length)
  ▶ POINTER (storage-data-location-pointer)
    [ WAIT ← ] [ KEEP ]
    [ NOWAIT ]
  [ LONG ← ] [ USER ← ] [ STGID (storage-id) ]
  [ SHORT - ] [ SHARED ]
  [ VALUE (initial-value) ]
  [ LOCATION IS [ ANY ← ] ] ;
    [ BELOW ]

```

GET TIME (DC/UCF)

```

▶▶ GET TIME [ BINARY ← ] INTO (return-time)
  [ DECIMAL - ]
  [ EDIT ]
  [ DATE INTO (return-date) ] ;

```

IF

```

▶▶ IF [ NOT ] SET (set-name) [ EMPTY ] THEN imperative-statement;--
  [ MEMBER ]

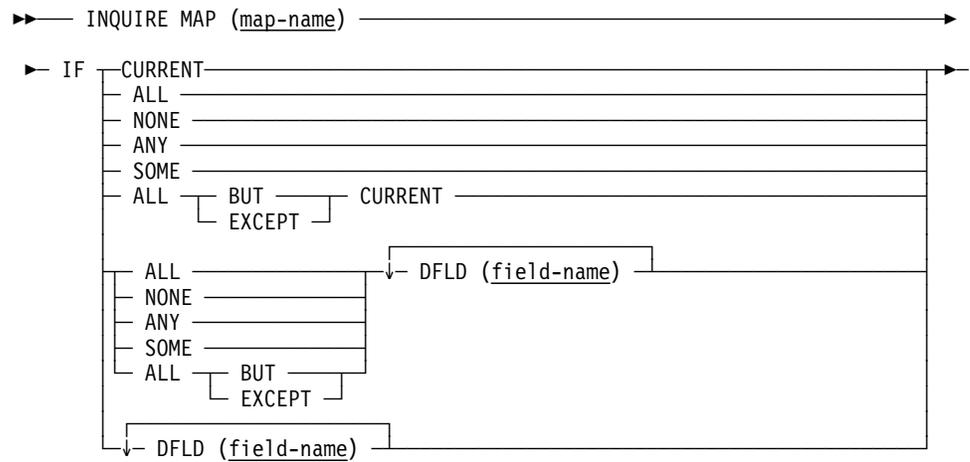
```

INQUIRE MAP (DC/UCF)

```

▶▶ INQUIRE MAP (map-name)
  ▶ MOVE [ AID TO (aid-indicator) ] ;
    [ CURSOR TO (cursor-row) (cursor-column) ]
    [ IN LENGTH FOR (field-name) TO (field-length) ]
▶▶ INQUIRE MAP (map-name)
  ▶ IF INPUT [ UNFORMATTED ] THEN imperative-statement;
    [ TRUNCATED ]
    [ CHANGED ]
    [ EXTRANEOUS - ]
▶▶ INQUIRE MAP (map-name)
  ▶ IF CURSOR AT DFLD (field-name) THEN imperative-statement;

```

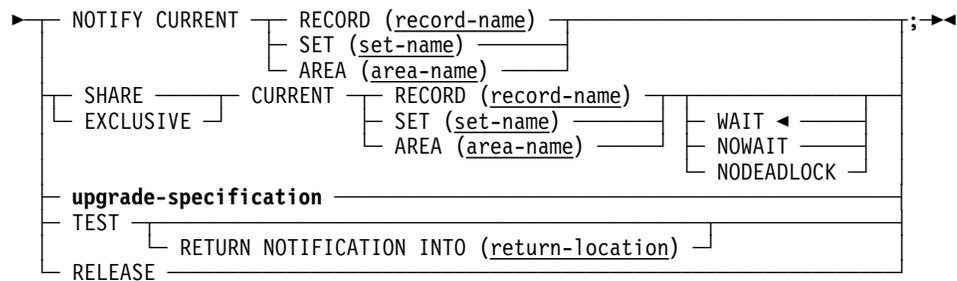
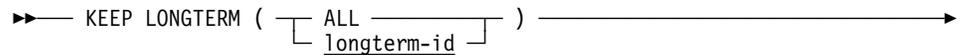


THEN imperative-statement;

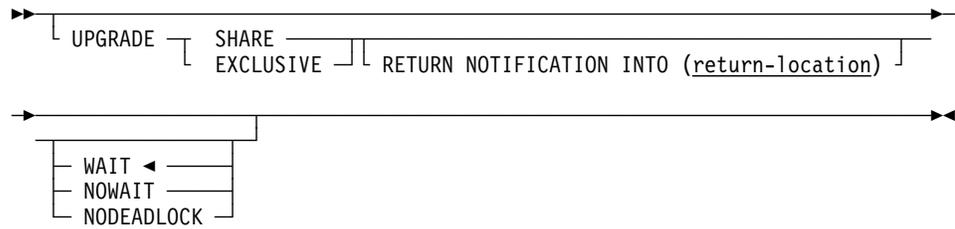
KEEP



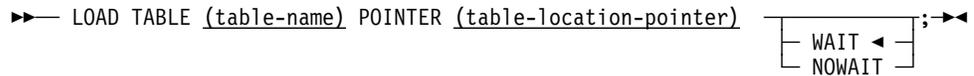
KEEP LONGTERM (DC/UCF)



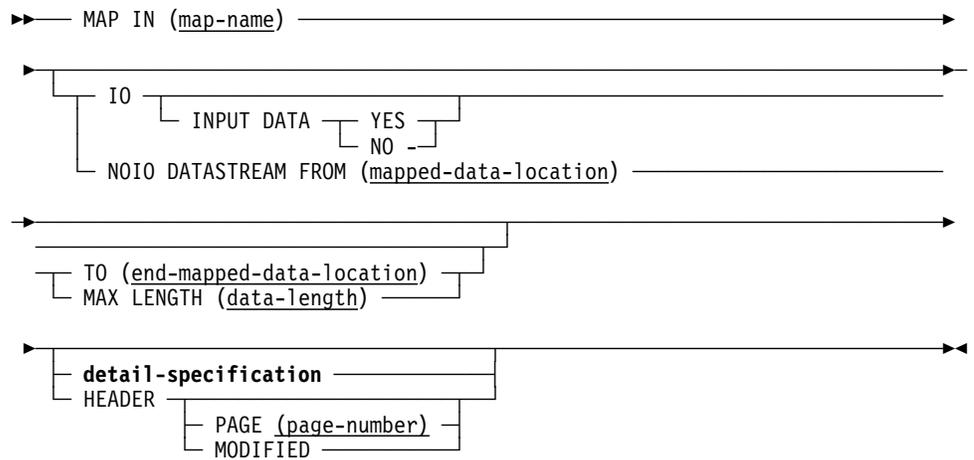
Expansion of upgrade-specification



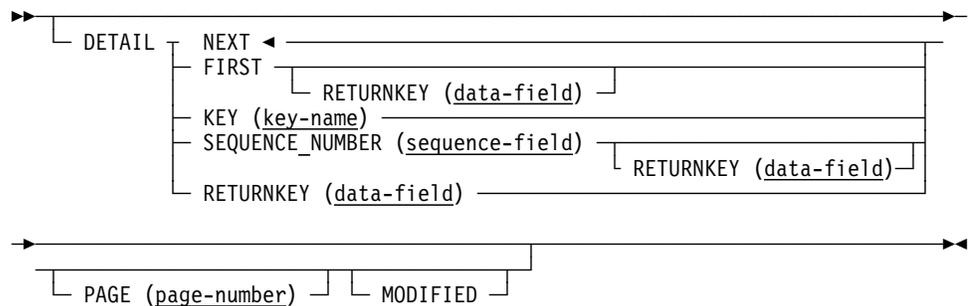
LOAD TABLE (DC/UCF)



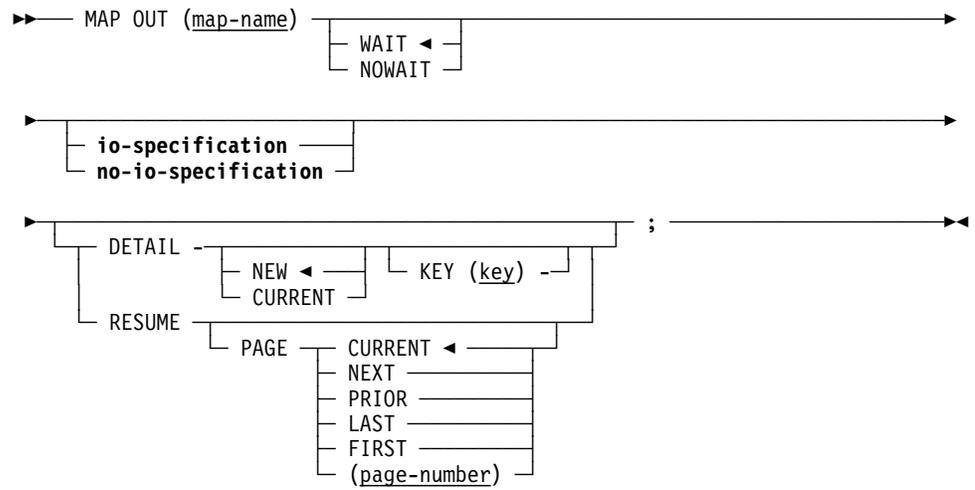
MAP IN (DC/UCF)



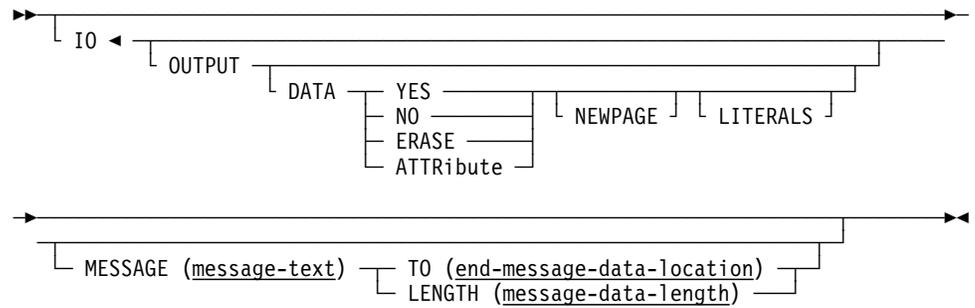
Expansion of detail-specification



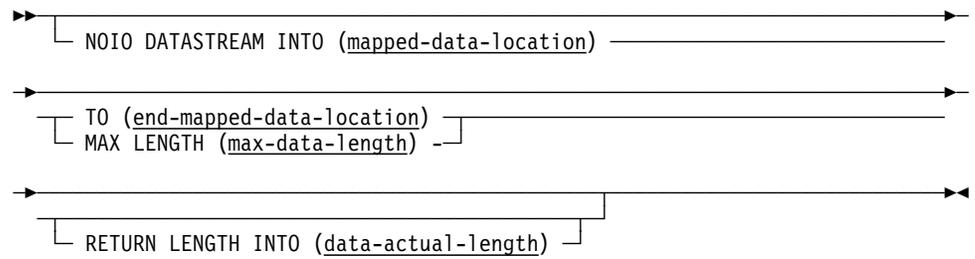
MAP OUT (DC/UCF)



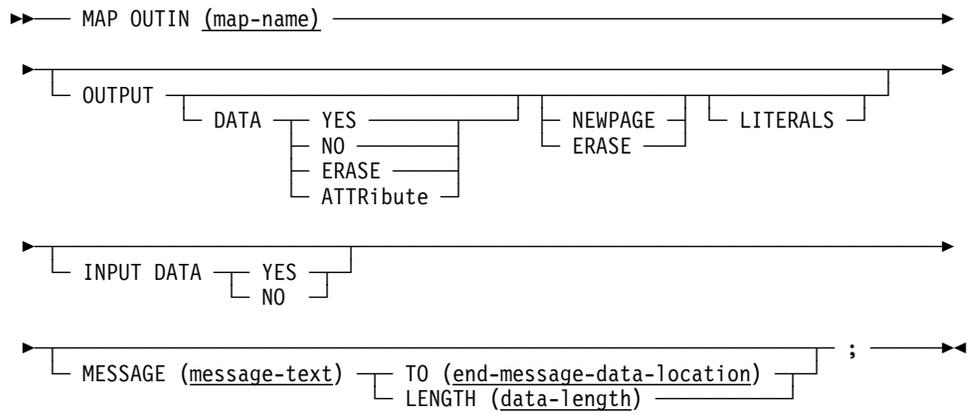
Expansion of io-specification

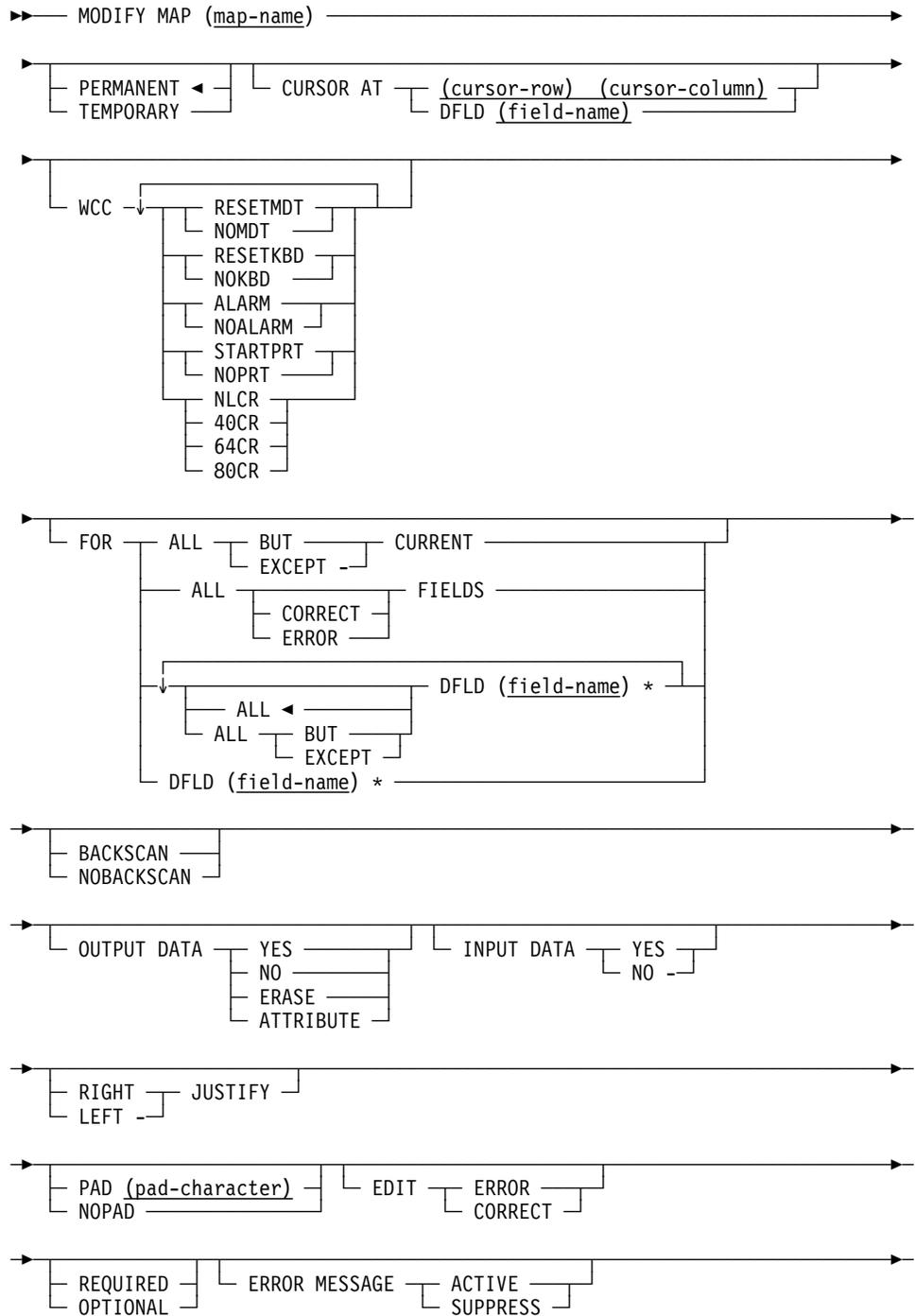


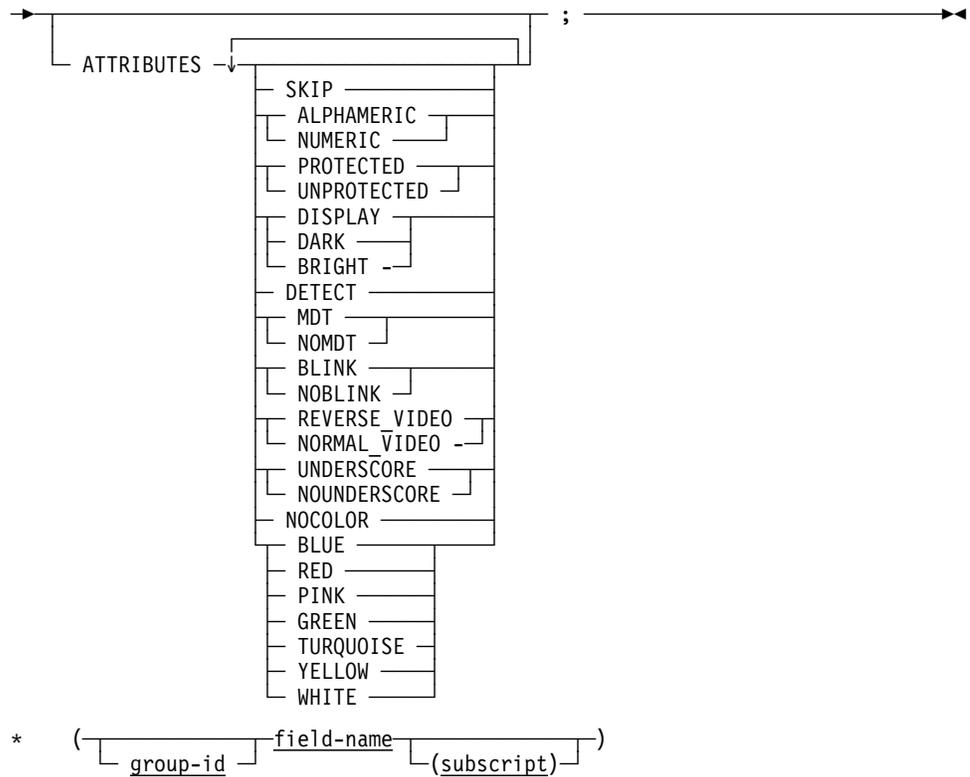
Expansion of no-io-specification



MAP OUTIN (DC/UCF)



MODIFY MAP (DC/UCF)



MODIFY RECORD

➤➤ MODIFY RECORD (record-name);

MODIFY RECORD (LRF)

➤➤ MODIFY RECORD (logical-record-name)

┌ FROM (alt-logical-record-location) ─┐ ┌ WHERE (boolean-expression) ─┐

└──────────────────────────────────┘ └──────────────────────────────────┘

┌ ON LR_STATUS (path-status) imperative-statement ─┐ ;

OBTAIN (LRF)

➤➤ OBTAIN ┌ FIRST ─┐ RECORD (logical-record-name)

└ NEXT ─┘

┌ INTO (alt-logical-record-location) ─┐ ┌ WHERE (boolean-expression) ─┐

└──────────────────────────────────┘ └──────────────────────────────────┘

┌ ON LR_STATUS (path-status) imperative-statement ─┐ ;

POST (DC/UCF)

➤➤ POST ┌ EVENT (ecb-name) ─┐ ;

└ EVENT NAME (ecb-id) ─┘ ┌ CLEAR ─┘

PUT QUEUE (DC/UCF)

```

▶▶ PUT QUEUE [ ID (queue-id) ] [ FIRST
              LAST ]
▶ FROM (queue-data-location) [ TO (end-queue-data-location)
                              LENGTH (queue-data-length) ]
▶ [ RETURN RECORD ID INTO (return-queue-record-id) ]
▶ [ RETENTION (queue-retention-period) ] ;

```

PUT SCRATCH (DC/UCF)

```

▶▶ PUT SCRATCH [ AREA ID (scratch-area-id) ]
▶ FROM (scratch-data-location) [ TO (end-scratch-data-location)
                                 LENGTH (scratch-data-location-length) ]
▶ [ RECORD ID (scratch-record-id) [ REPLACE ] ]
▶ [ RETURN RECORD ID INTO (scratch-record-id) ] ;

```

READ LINE FROM TERMINAL (DC/UCF)

```

▶▶ READ LINE FROM TERMINAL [ ECHO ] [ NOBACKPAGE ]
▶ INTO (input-data-location) [ TO (end-input-data-location)
                               MAX LENGTH (input-data-location-max-length) ]
▶ [ RETURN LENGTH INTO (input-data-actual-length) ] ;

```

READ TERMINAL (DC/UCF)

```

▶▶ READ TERMINAL [ WAIT
                 NOWAIT ]
▶ [ MODIFIED
  BUFFER ] FROM POSITION (screen-position)
▶ INTO (input-data-location) [ TO (end-input-data-location)
                               MAX LENGTH (input-data-max-length) ]
▶ [ RETURN LENGTH INTO (input-data-actual-length) ] ;

```

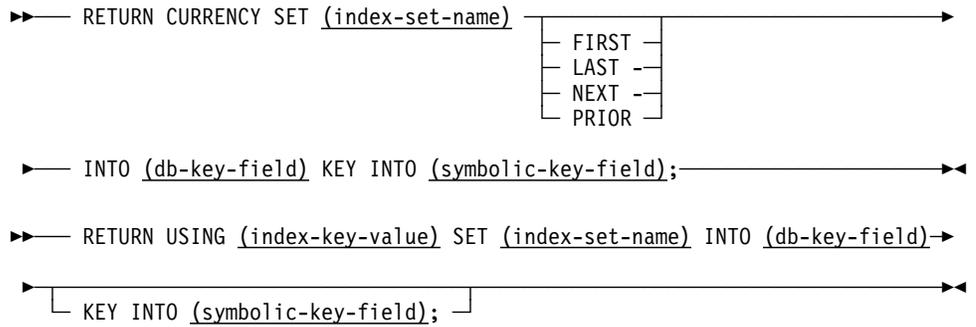
READY

```

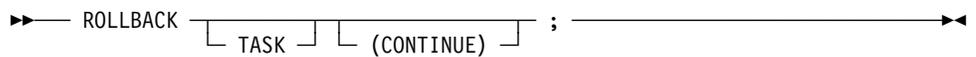
▶▶ READY [ AREA (area-name) ] [ PROTECTED
                                EXCLUSIVE ] [ RETRIEVAL
                                                UPDATE ] ;

```

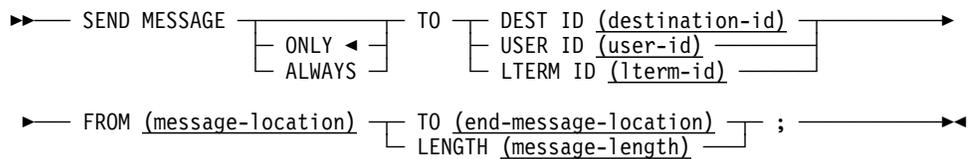
RETURN (DC/UCF)



ROLLBACK

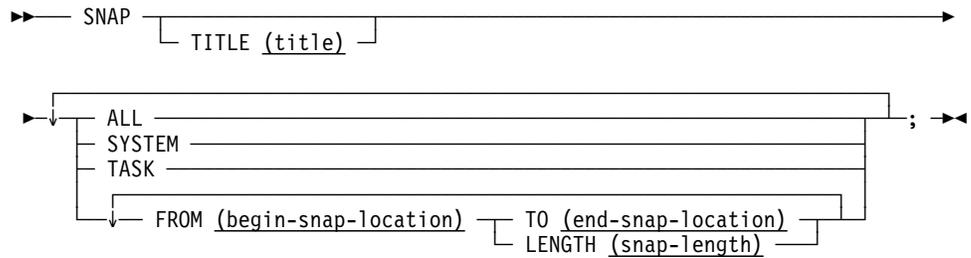
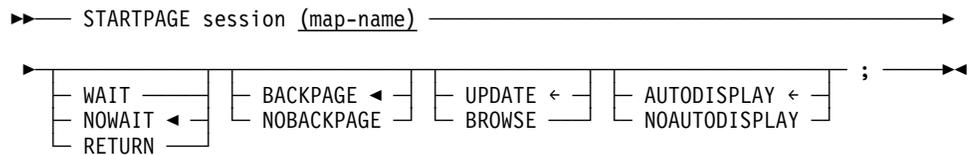
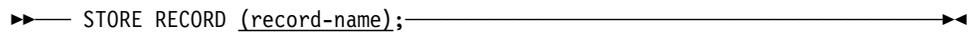
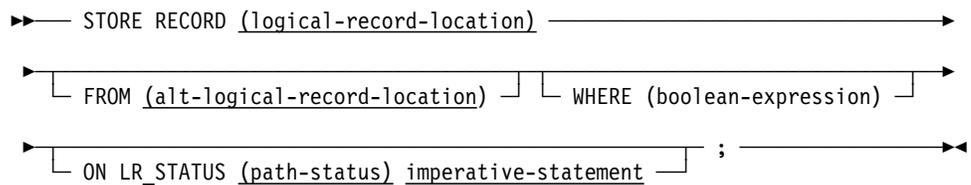
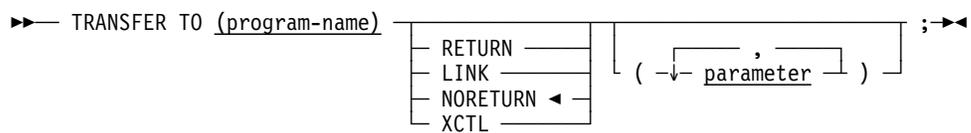
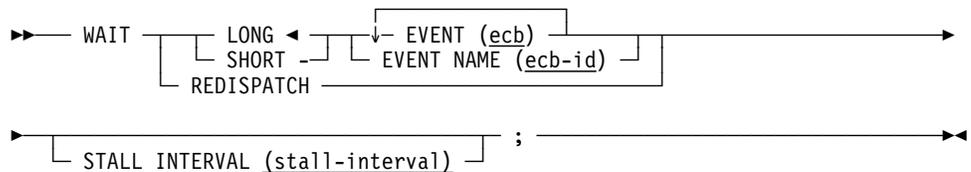
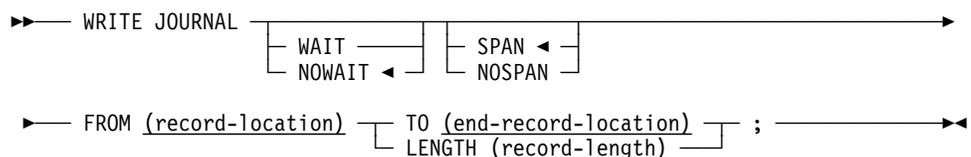


SEND MESSAGE (DC/UCF)

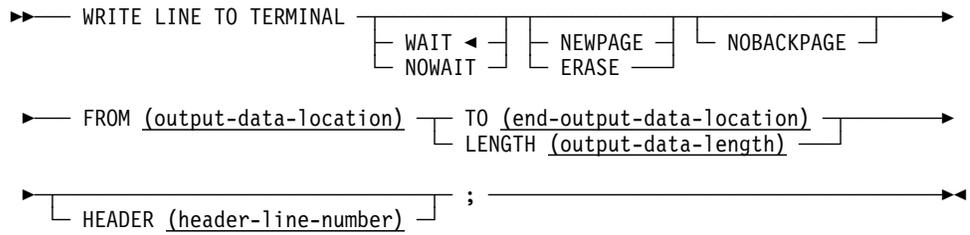


SET TIMER (DC/UCF)

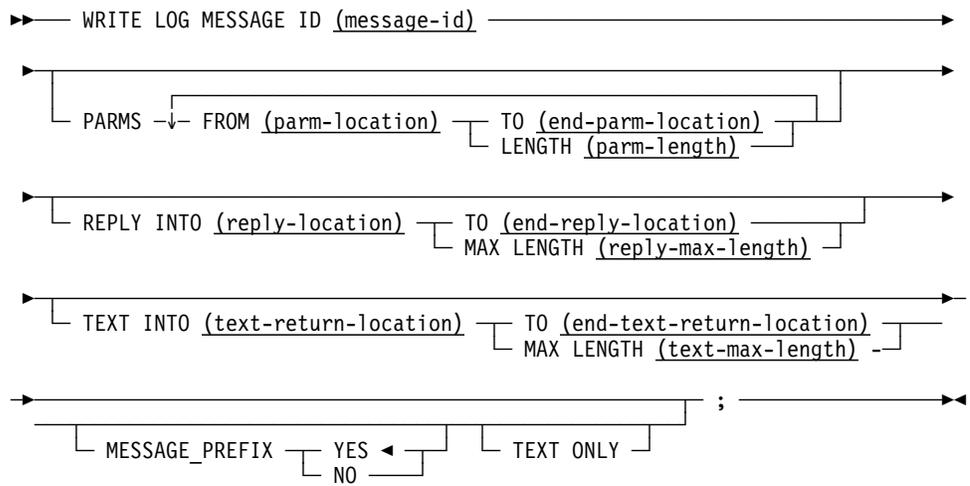


SNAP (DC/UCF)**STARTPAGE (DC/UCF)****STORE RECORD****STORE RECORD (LRF)****TRANSFER (DC/UCF)****WAIT (DC/UCF)****WRITE JOURNAL (DC/UCF)**

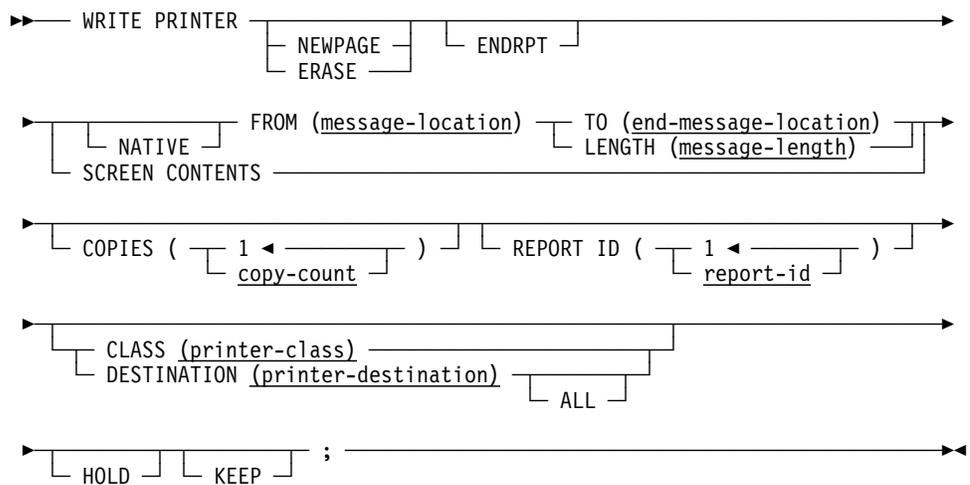
WRITE LINE TO TERMINAL (DC/UCF)

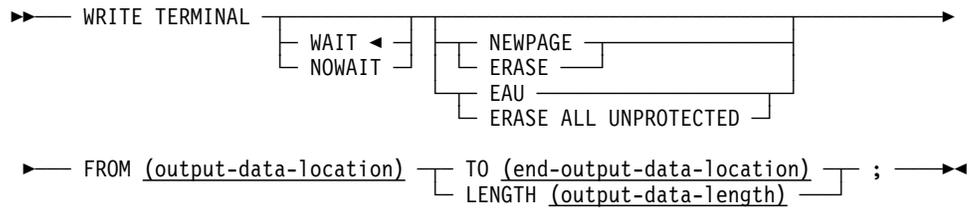
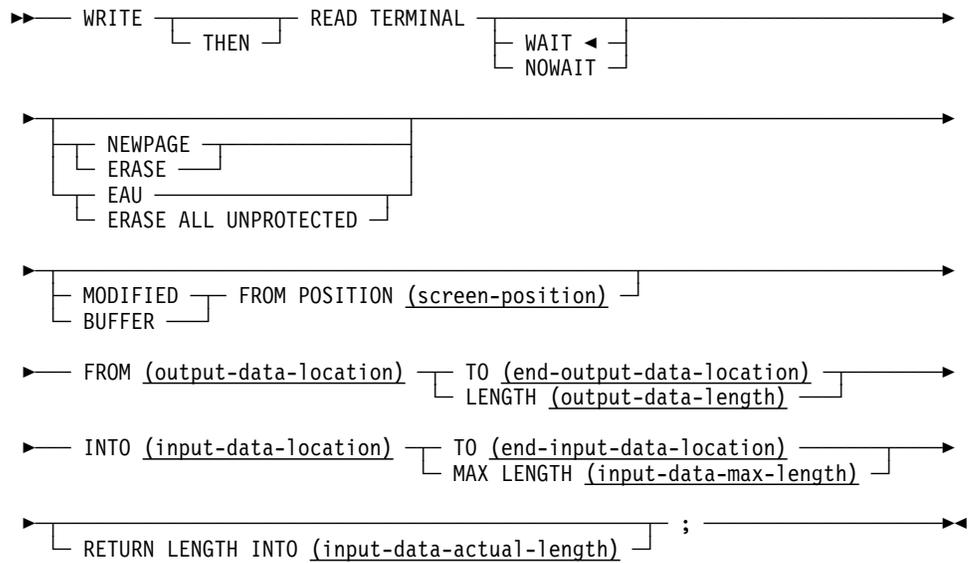
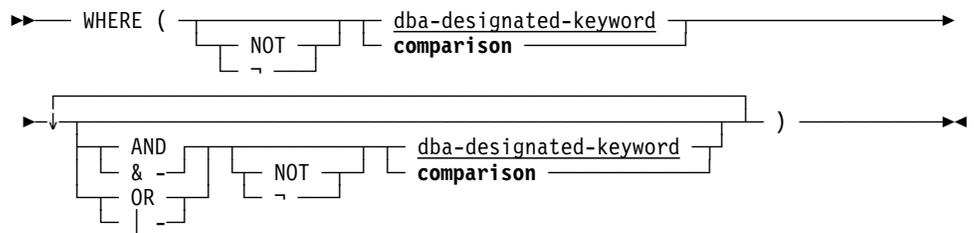


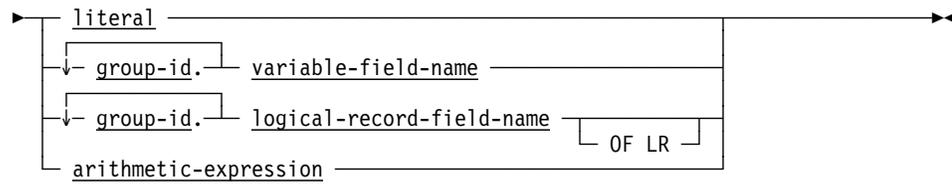
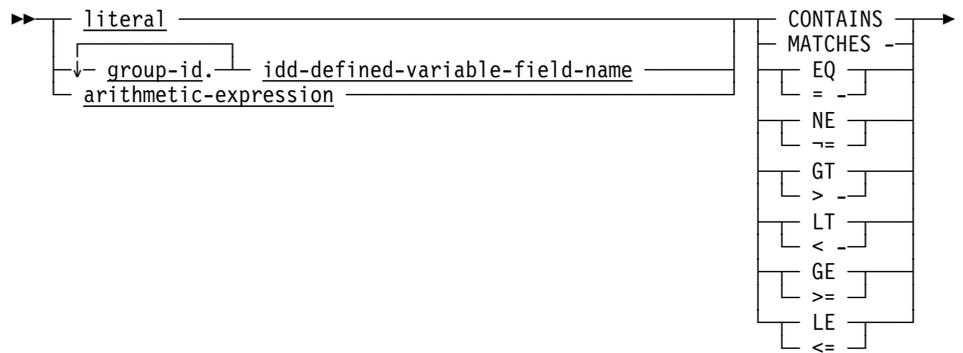
WRITE LOG (DC/UCF)



WRITE PRINTER (DC/UCF)



WRITE TERMINAL (DC/UCF)**WRITE THEN READ TERMINAL (DC/UCF)****Logical-record clauses (WHERE)****Expansion of comparison**



Logical-record clauses (ON)

▶▶ ON LR_STATUS (path-status) imperative-statement;

Chapter 4. Assembler Syntax

4.1 Assembler Precompiler options

Dictionary usage mode

▶ [*RETRIEVAL _____]
[*PROTECTED-UPDATE]

The asterisk (*) must be in column 1.

Comment generation

▶ *SCHEMA-COMMENTS _____

The asterisk (*) must be in column 1.

List generation

▶ [*NODMLIST ◀]
[*DMLIST]

The asterisk (*) must be in column 1.

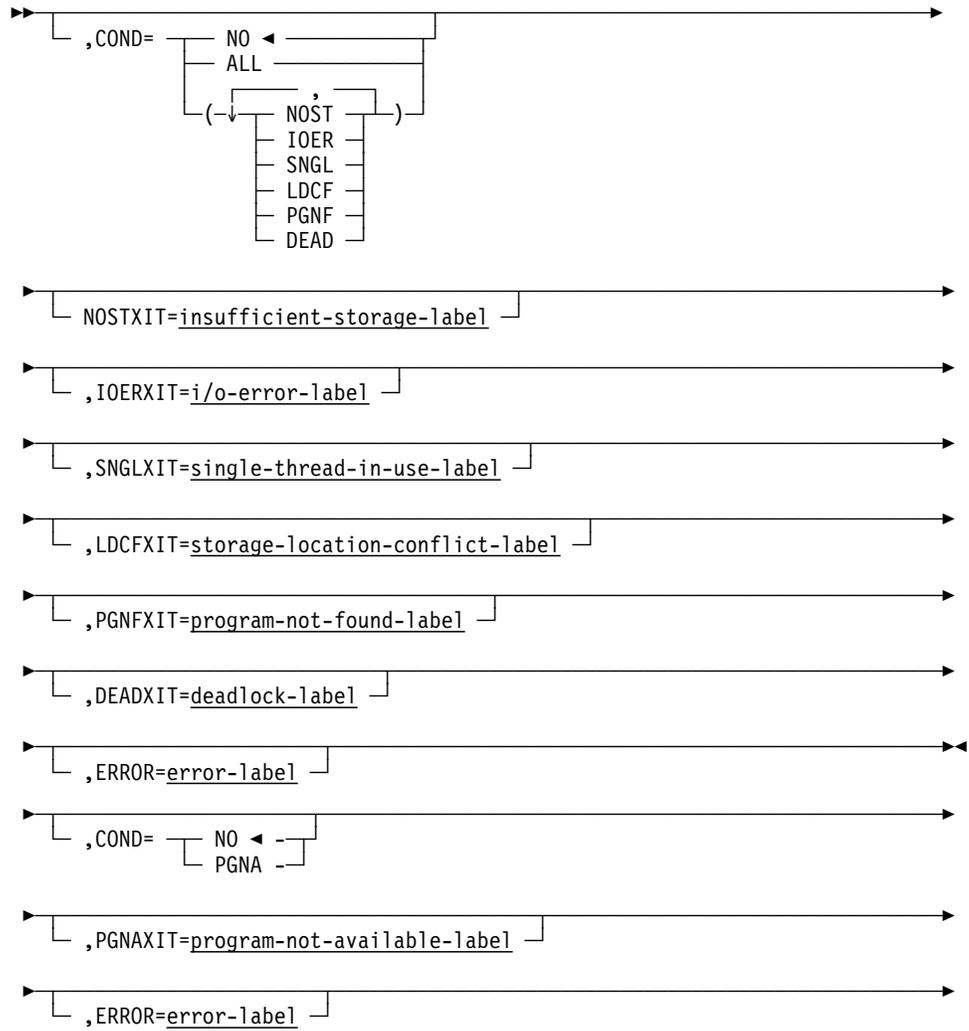
Log suppression

▶ *NO-ACTIVITY LOG _____

The asterisk (*) must be in column 1.

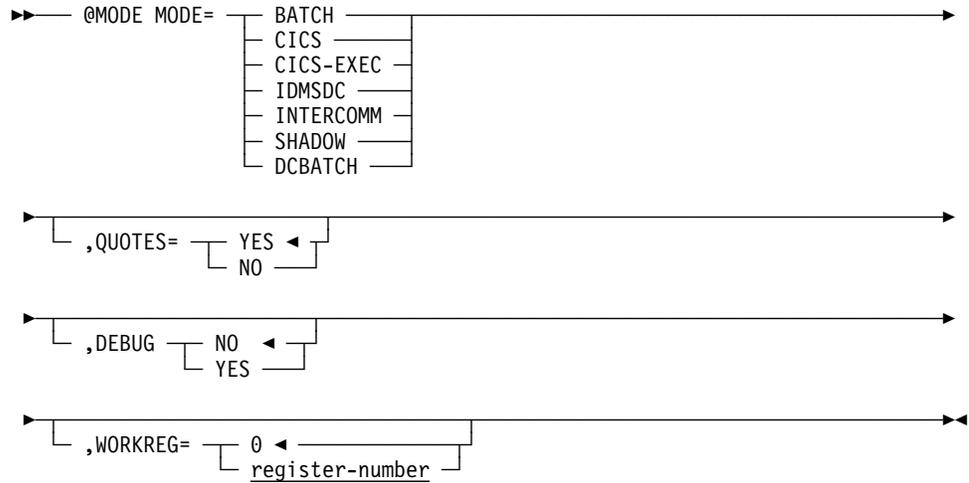
Logical-record request control (LRC) block

▶ [,ONLRSTS=path-status,GOTO=branch-location]

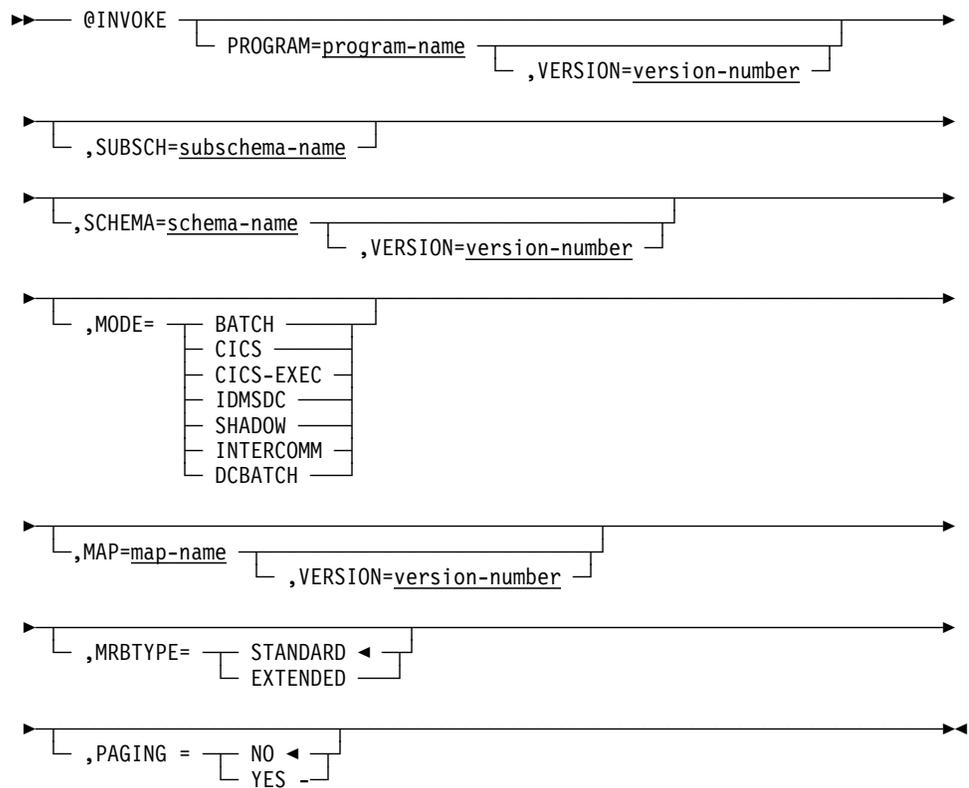
DC/UCF general registers

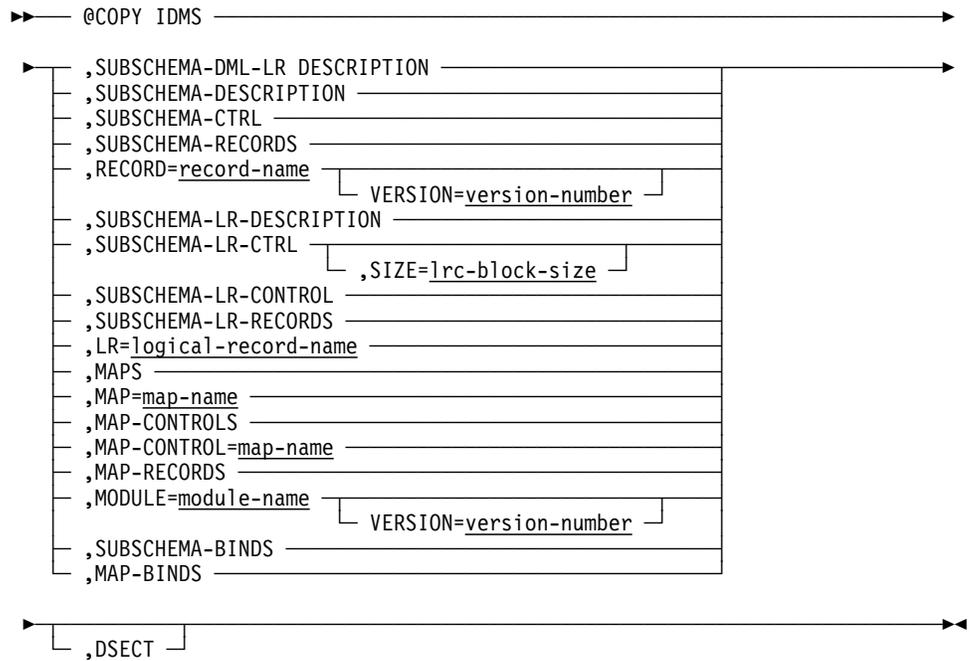
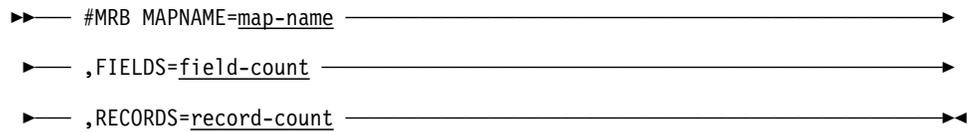
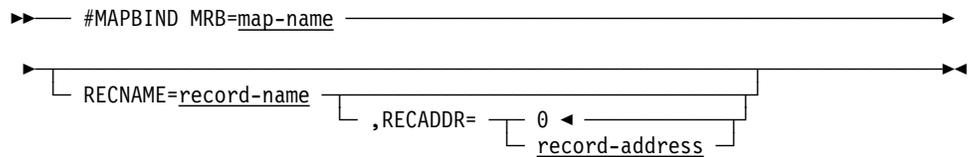
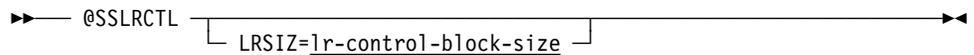
4.2 Assembler Precompiler-Directive statements

@MODE



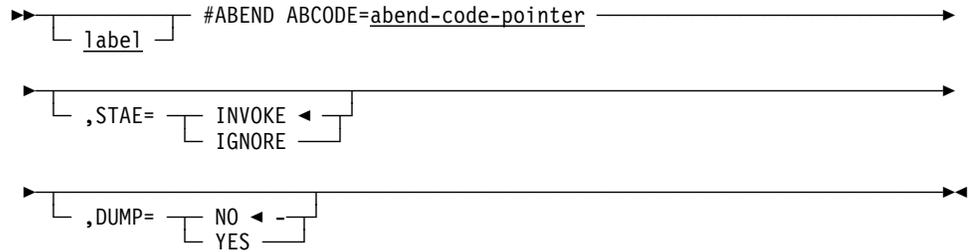
@INVOKE



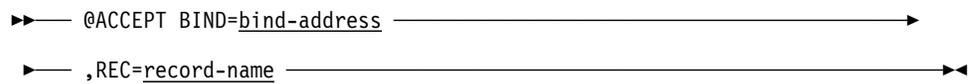
@COPY IDMS**#MRB****#MAPBIND****@SSCTRL****@SSLRCTL**

4.3 Assembler DML statements

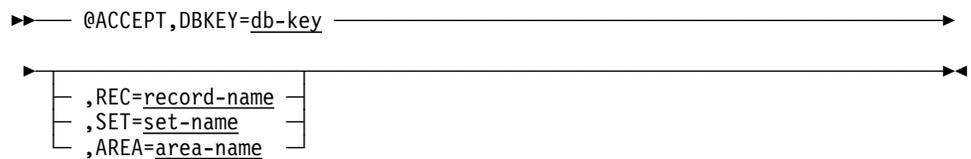
#ABEND



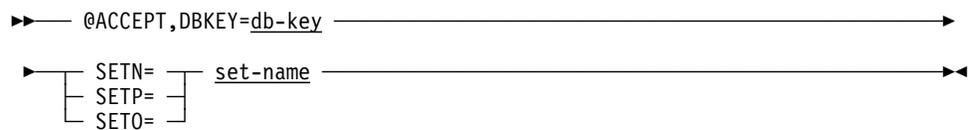
@ACCEPT BIND



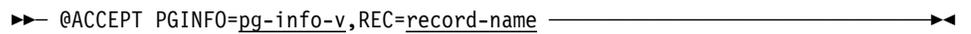
@ACCEPT DBKEY FROM CURRENCY



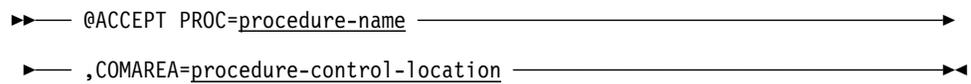
@ACCEPT DBKEY RELATIVE TO CURRENCY



@ACCEPT PGINFO

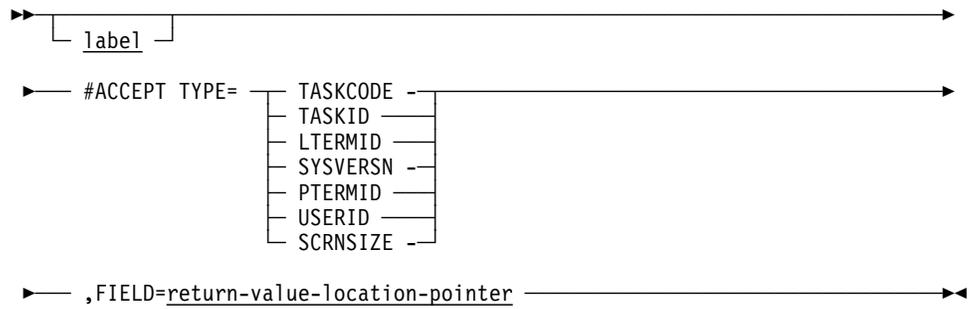
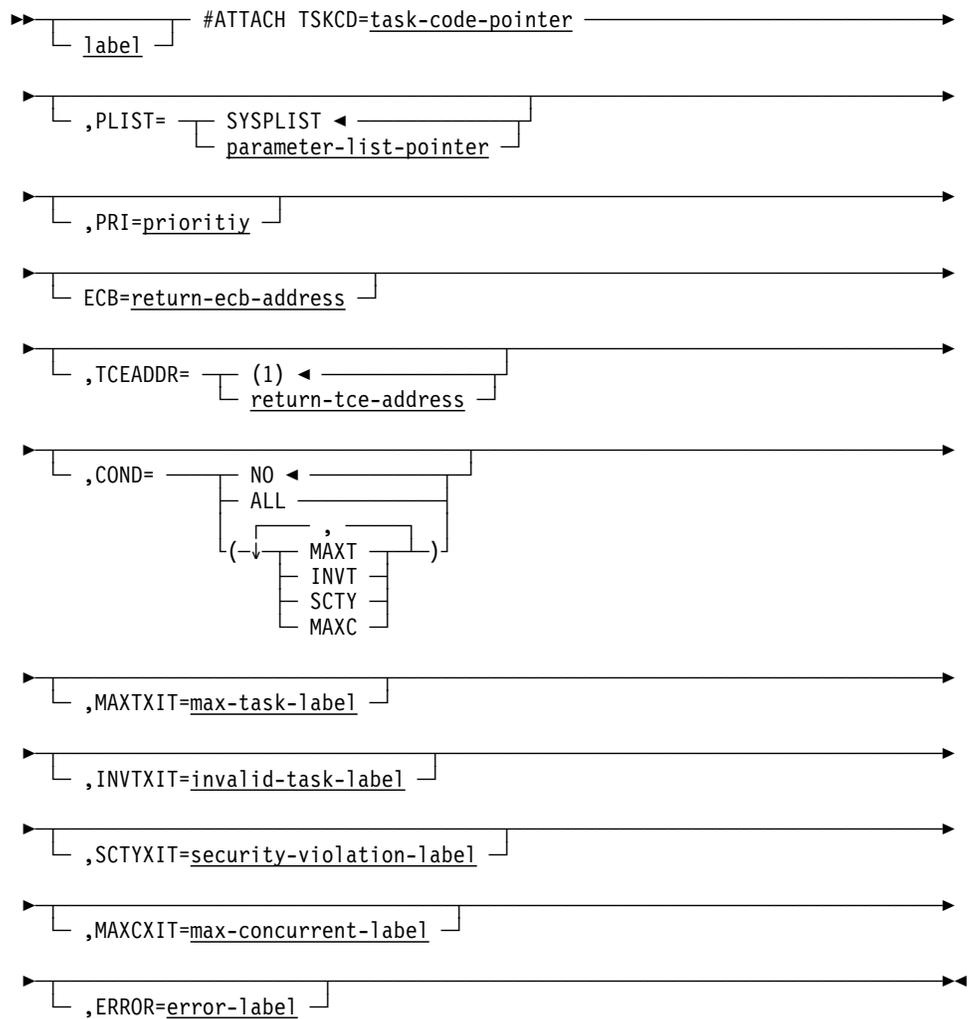
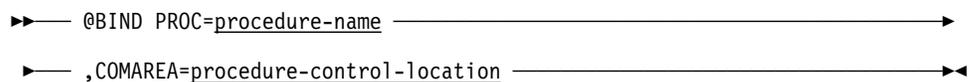


@ACCEPT PROC



@ACCEPT STATS



#ACCEPT**#ATTACH****@BIND PROC**

@BIND REC

▶— @BIND REC=record-name —————▶

▶— ,IOAREA=record-location —————▶

@BIND SUBSCH

```

▶▶ @BIND SUBSCH=subschema-name
└─┬─┘
  └─┬─┘ ,PGMNAME=program-name
    └─┬─┘ ,LRC=lr-control-block-location
      └─┬─┘ ,LRSIZ=lr-control-block-size
        └─┬─┘ ,DBNAME=database-name-pointer
          └─┬─┘ ,DBNODE=nodename-pointer
            └─┬─┘ ,DICTNAM=dictionary-name-pointer
              └─┬─┘ ,DICTNOD=dictionary-nodename-pointer
                ▶▶

```

#BIND TASK

```

▶▶ └─┬─┘ label #BIND TASK
    └─┬─┘ ,NODE=nodename
      ▶▶

```

#CHAP

```

▶▶ └─┬─┘ label #CHAP PRI=priority
    ▶▶

```

@COMMIT

```

▶▶ @COMMIT └─┬─┘ ,ALL
    ▶▶

```

#COMMIT

```

▶▶ └─┬─┘ label
    └─┬─┘ #COMMIT └─┬─┘ ,TASK └─┬─┘ ,ALL
      ▶▶

```

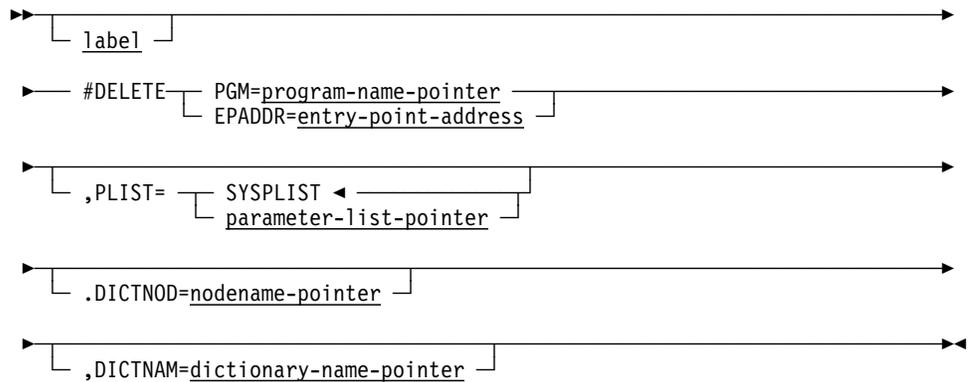
@CONNECT

```

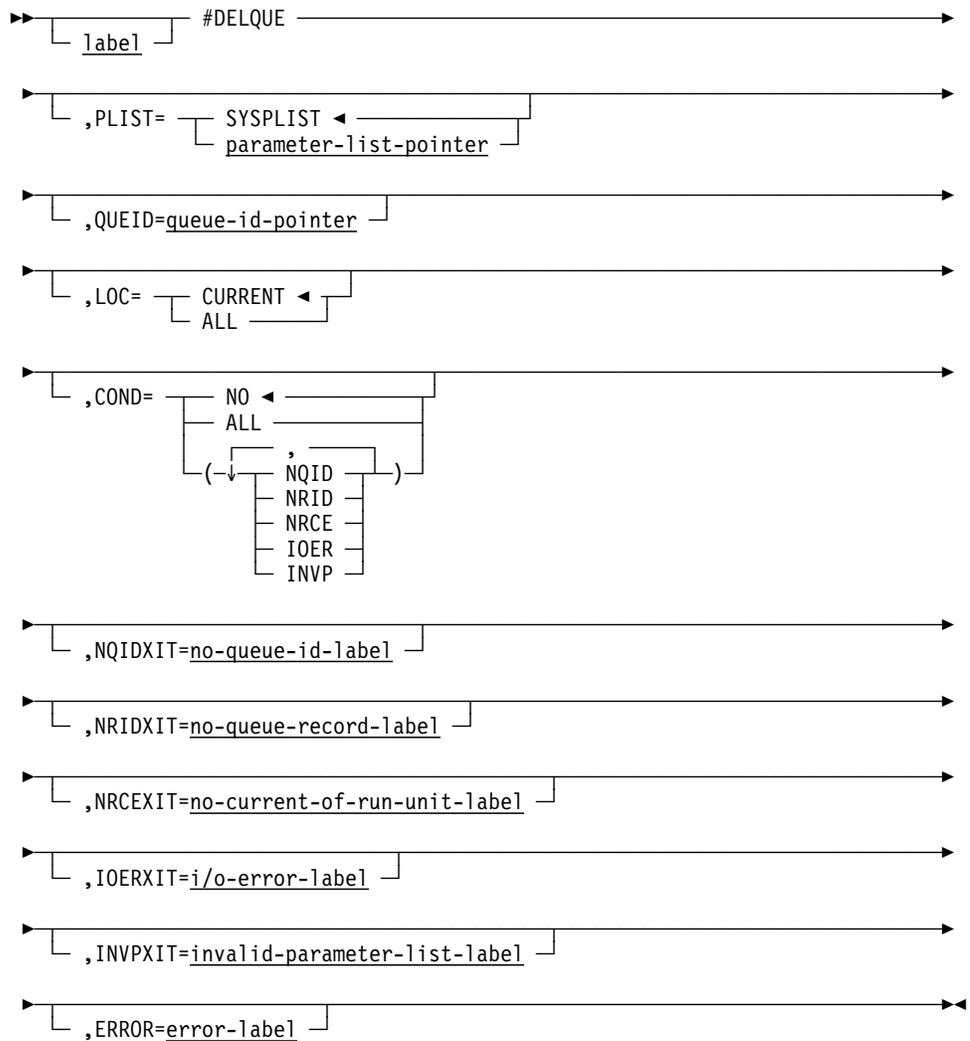
▶▶ @CONNECT REC=record-name
▶▶ ,SET=set-name
▶▶

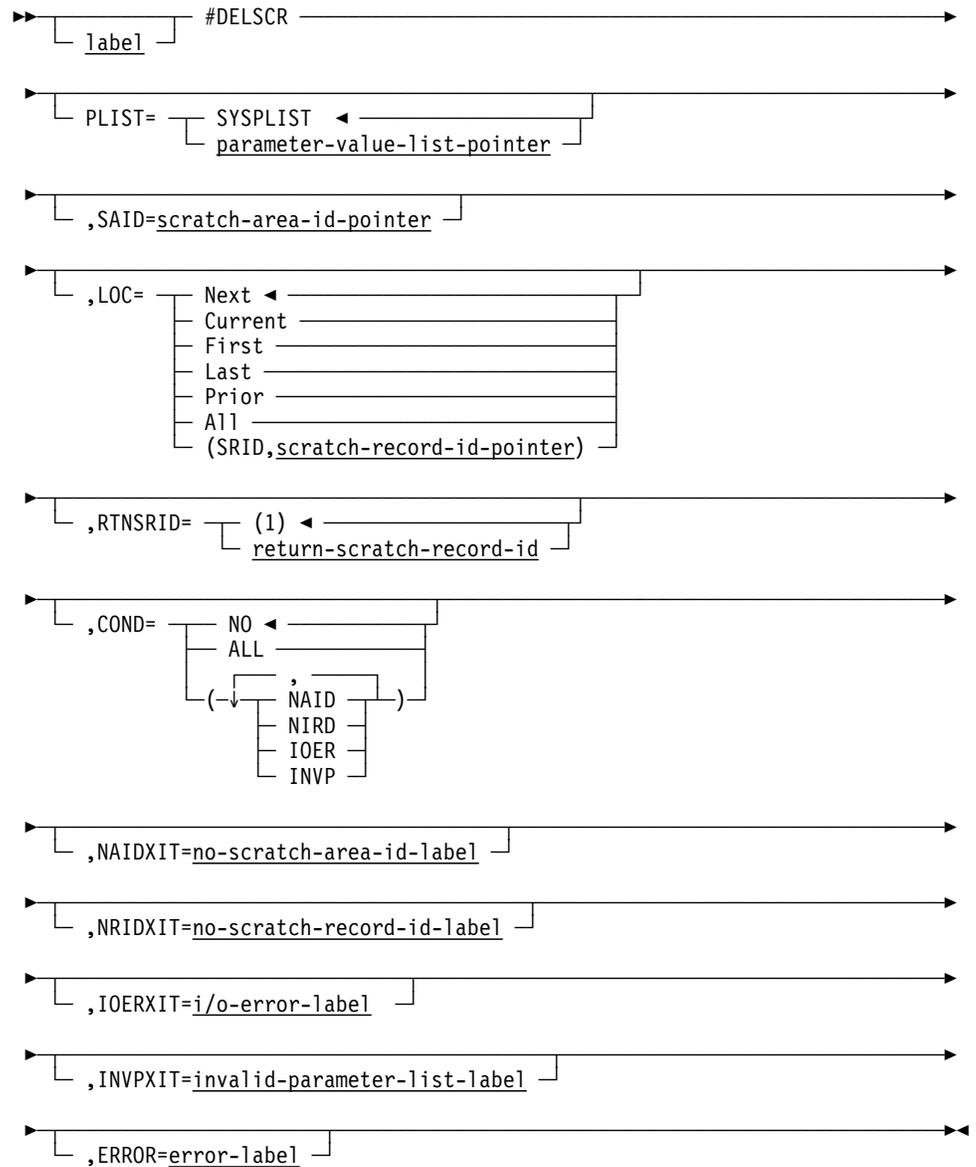
```

#DELETE

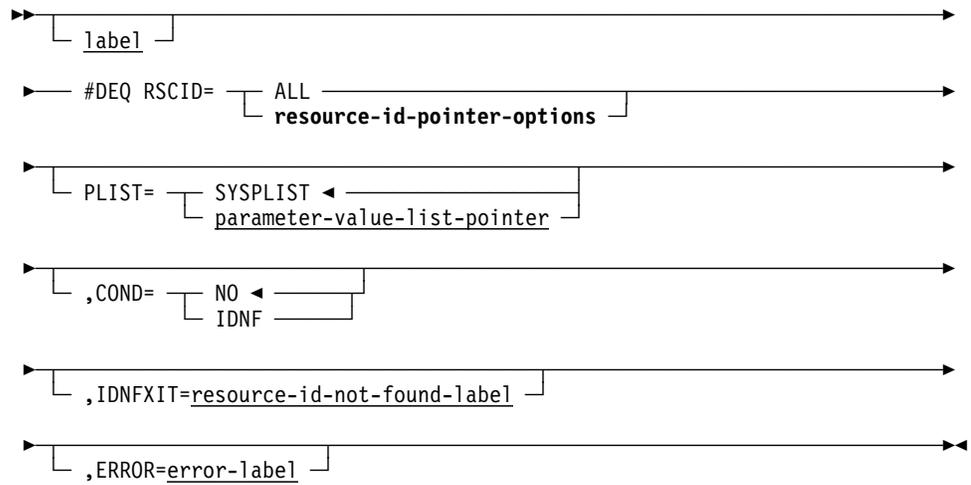


#DELQUE

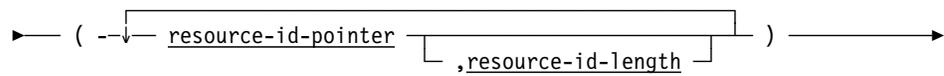


#DELSCR

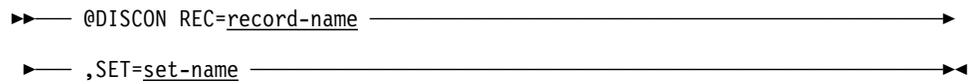
#DEQ



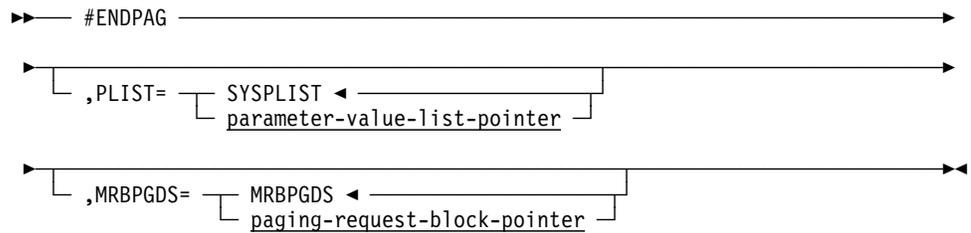
Expansion of resource-id-pointer-options

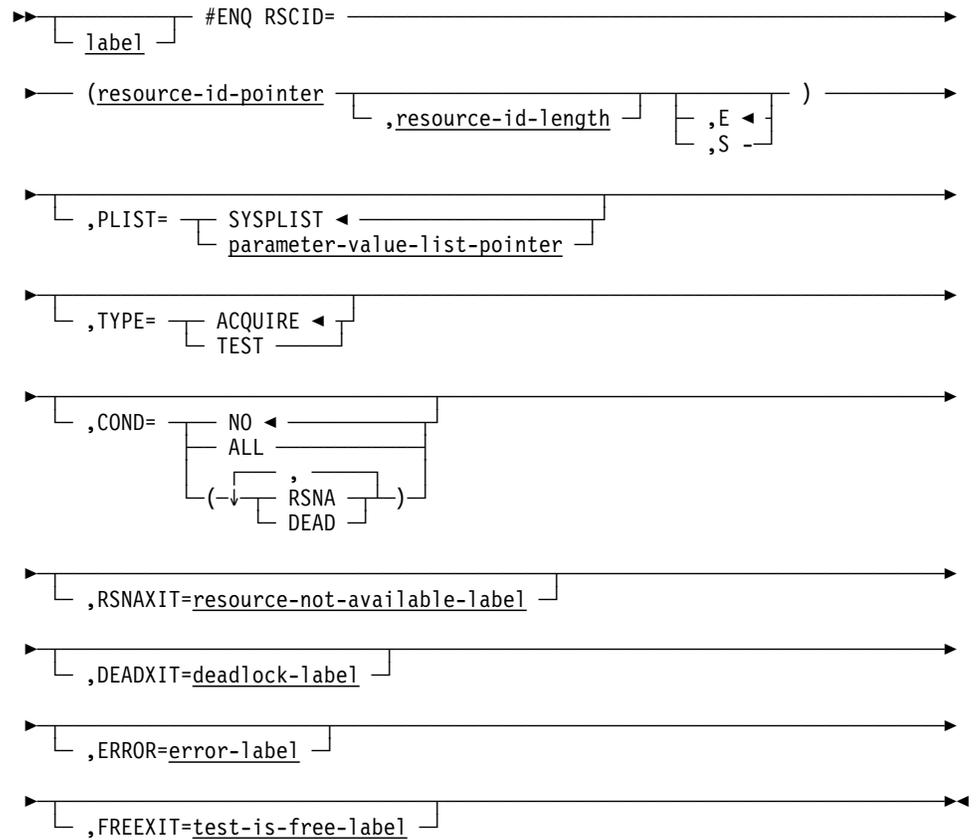
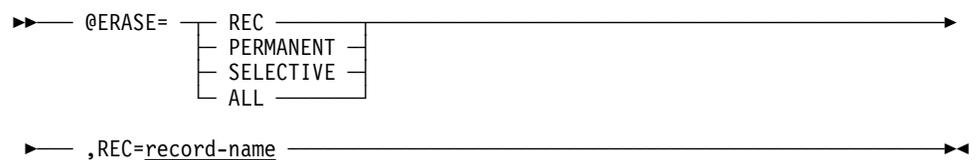
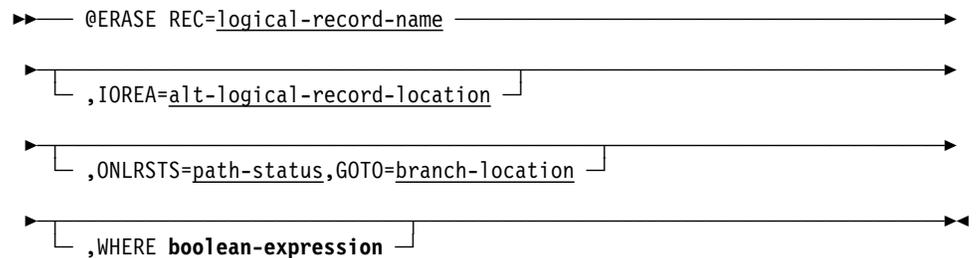


@DISCON

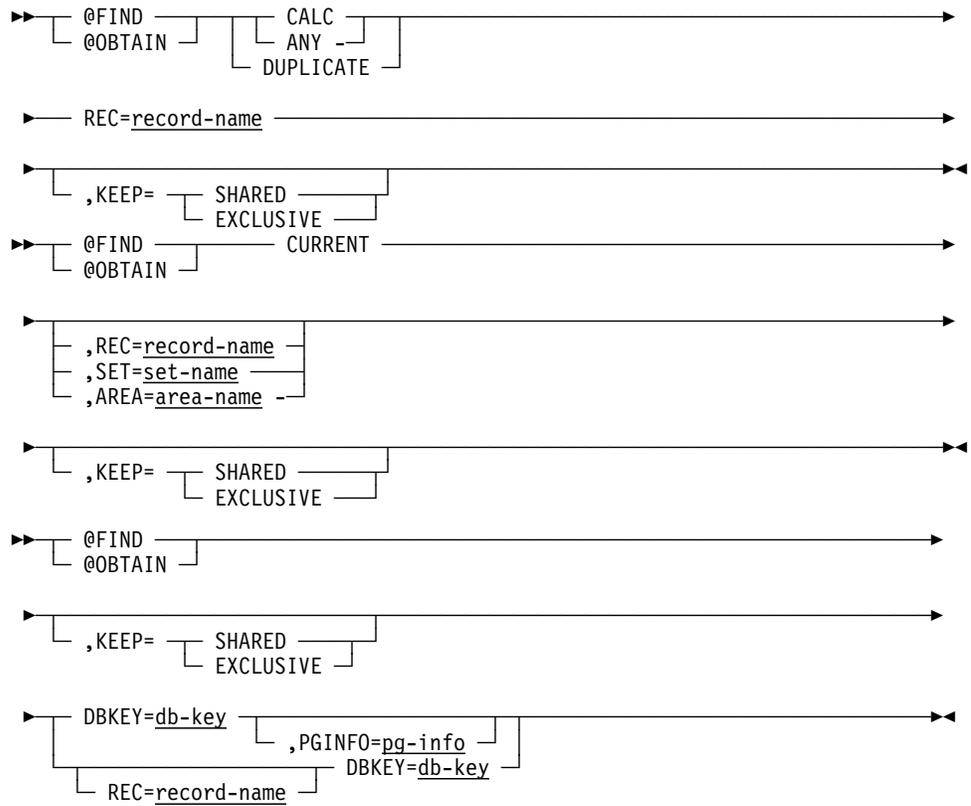


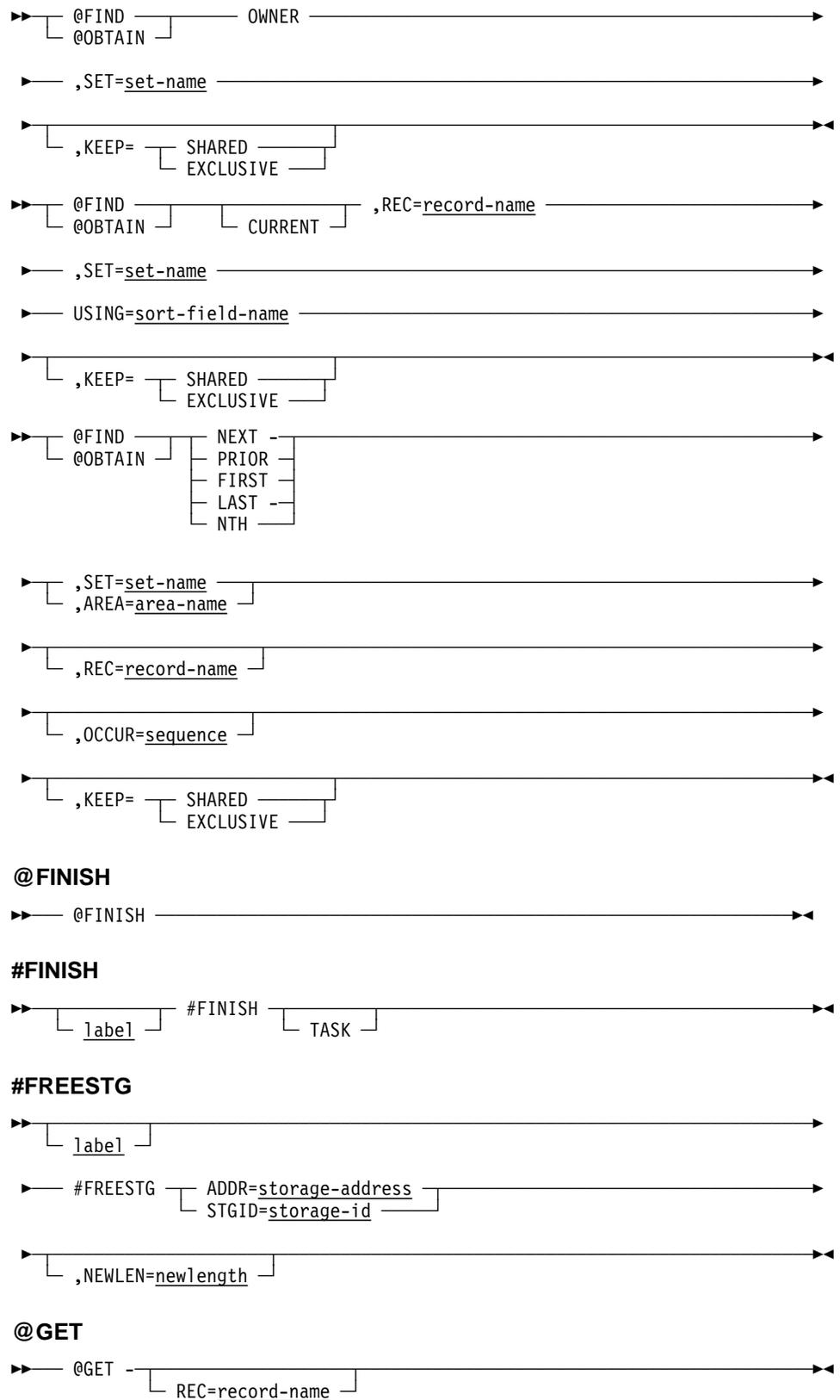
#ENDPAG



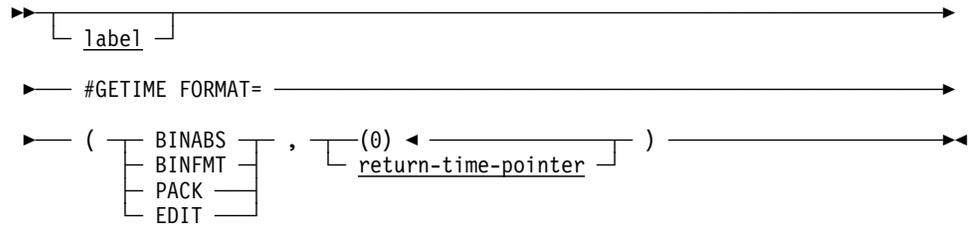
#ENQ**@ERASE****@ERASE (LRF)**

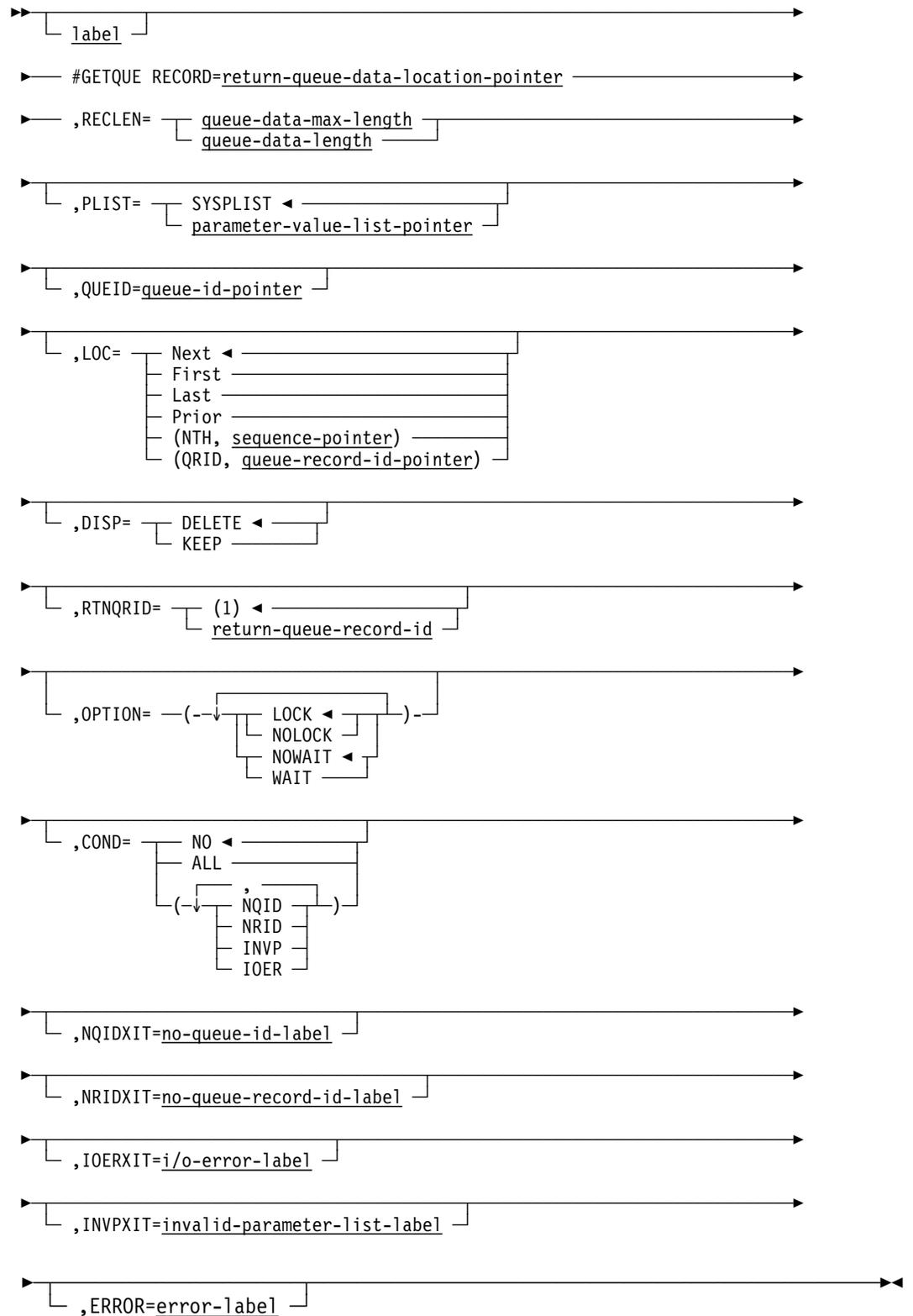
@FIND/@OBTAIN statements



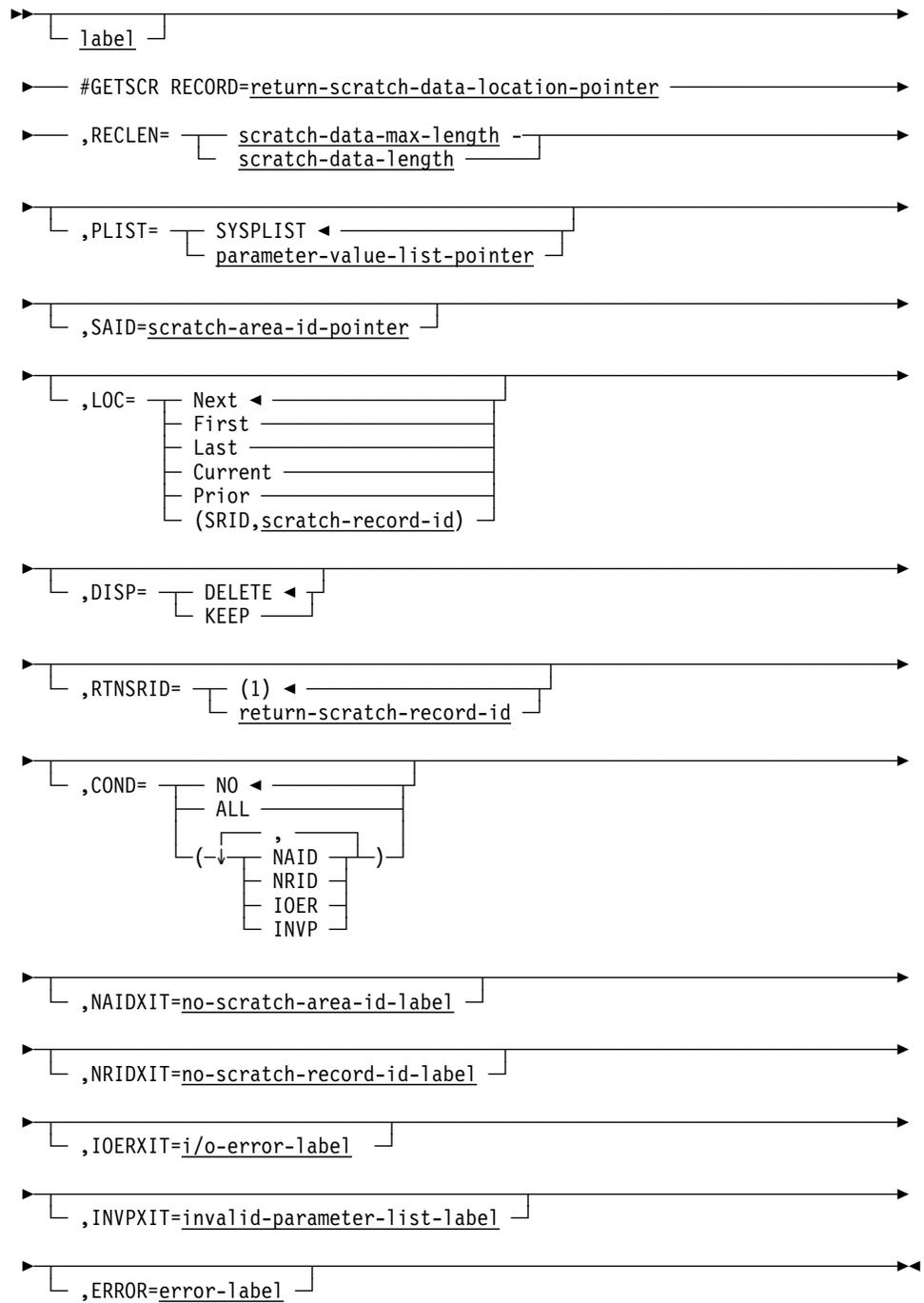


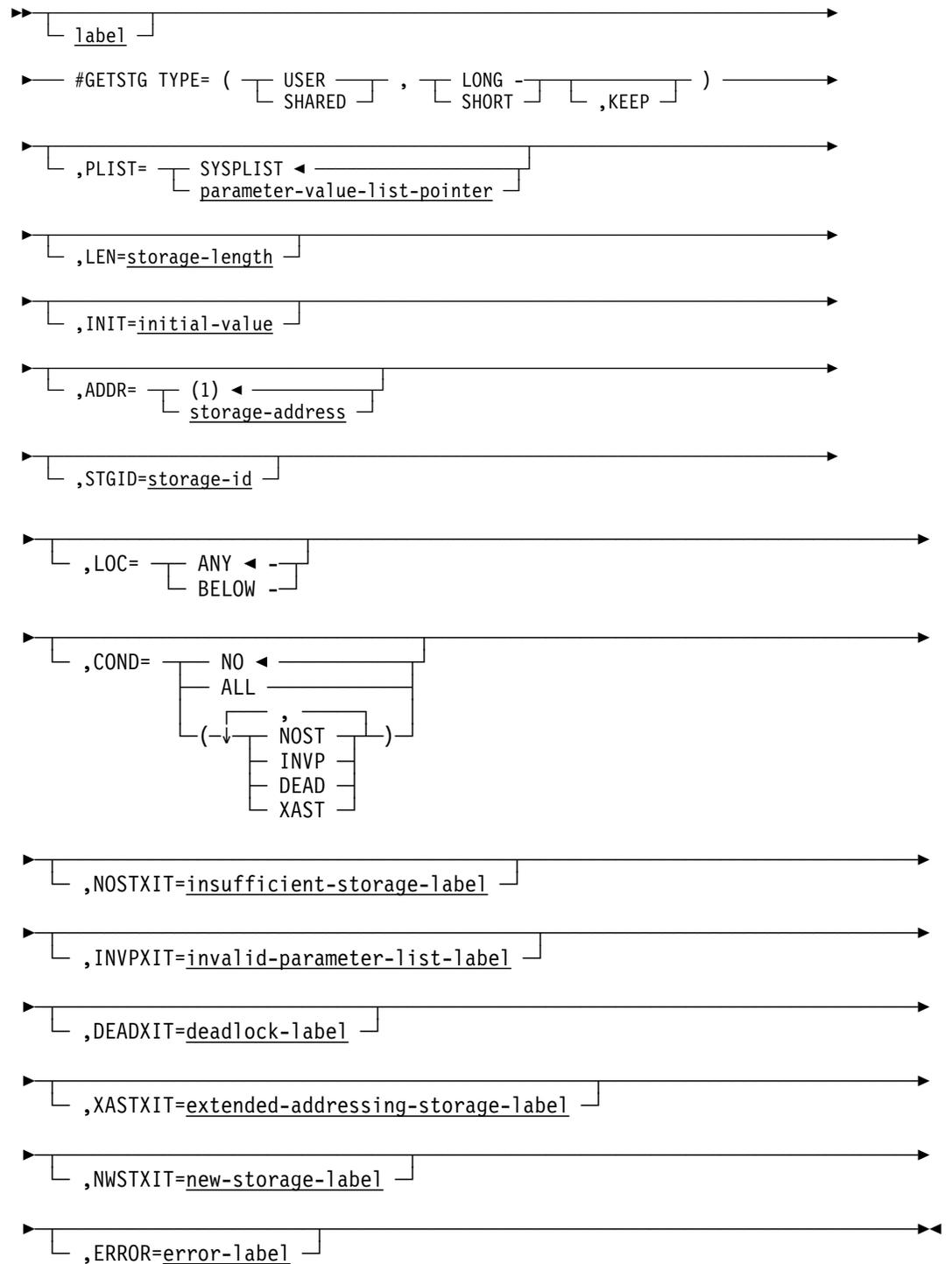
#GETIME



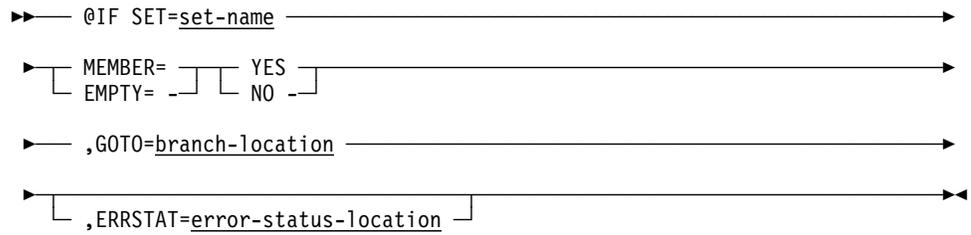
#GETQUE

#GETSCR

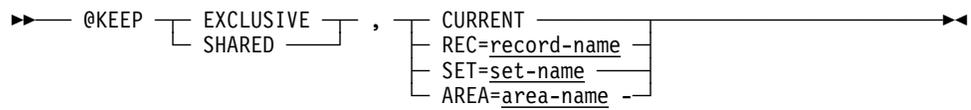


#GETSTG

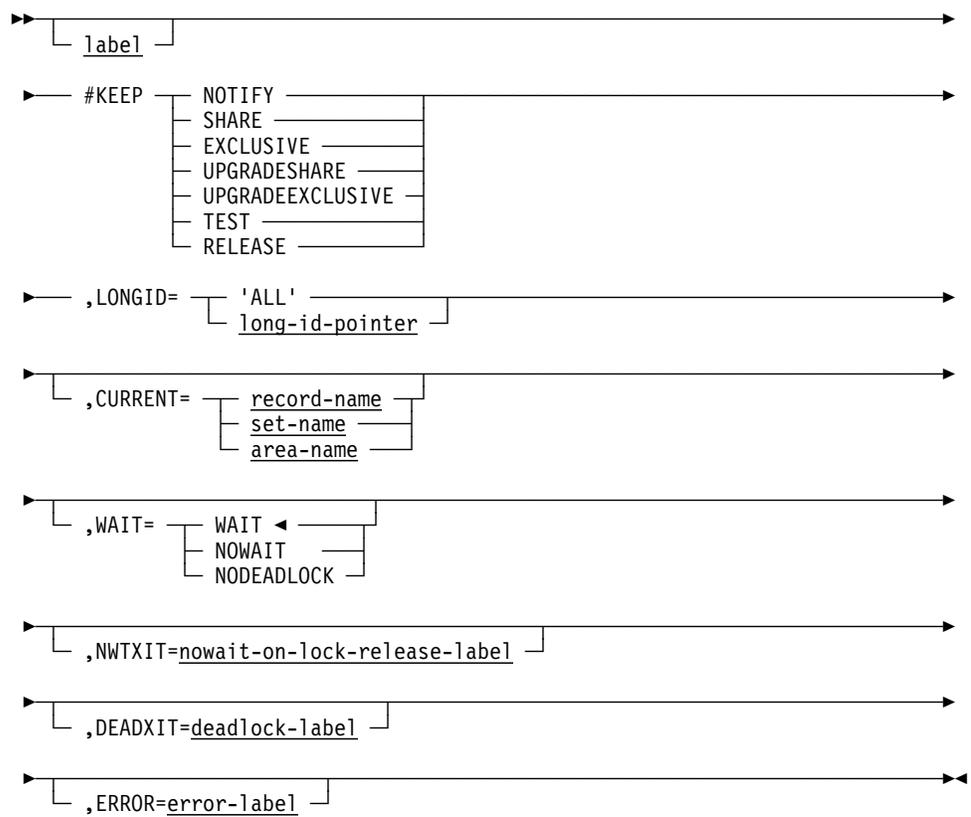
@IF



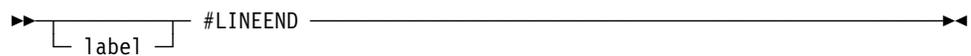
@KEEP

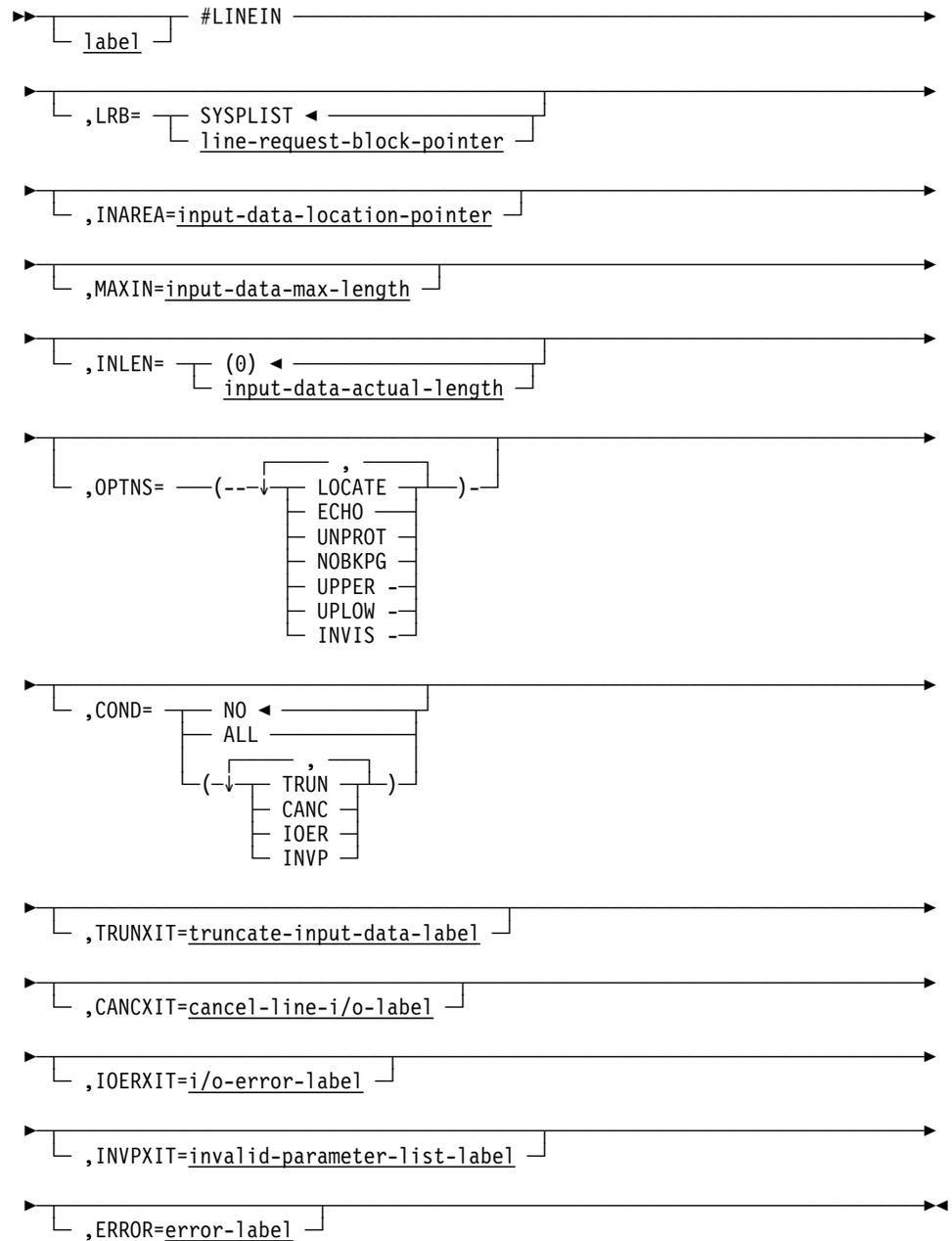


#KEEP

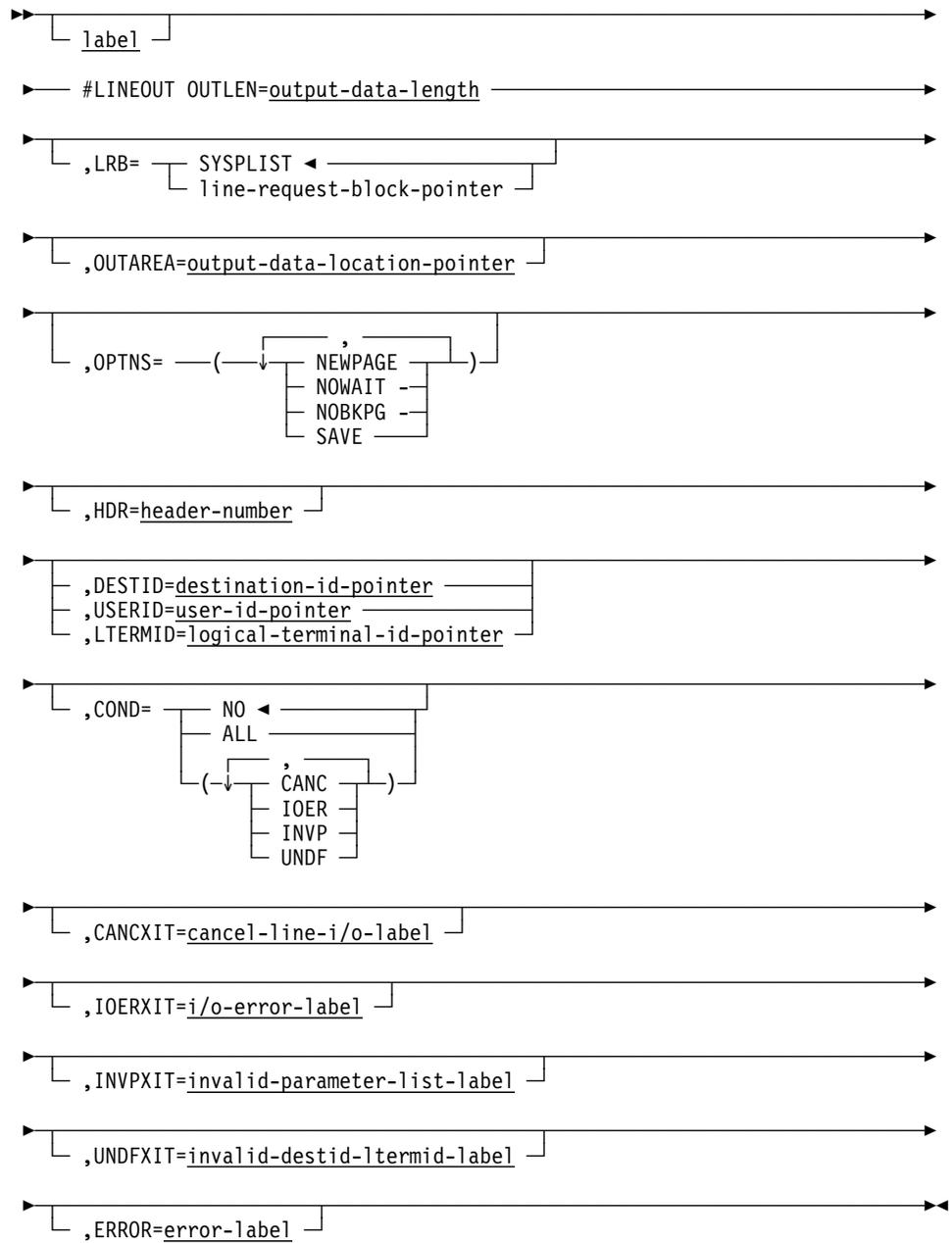


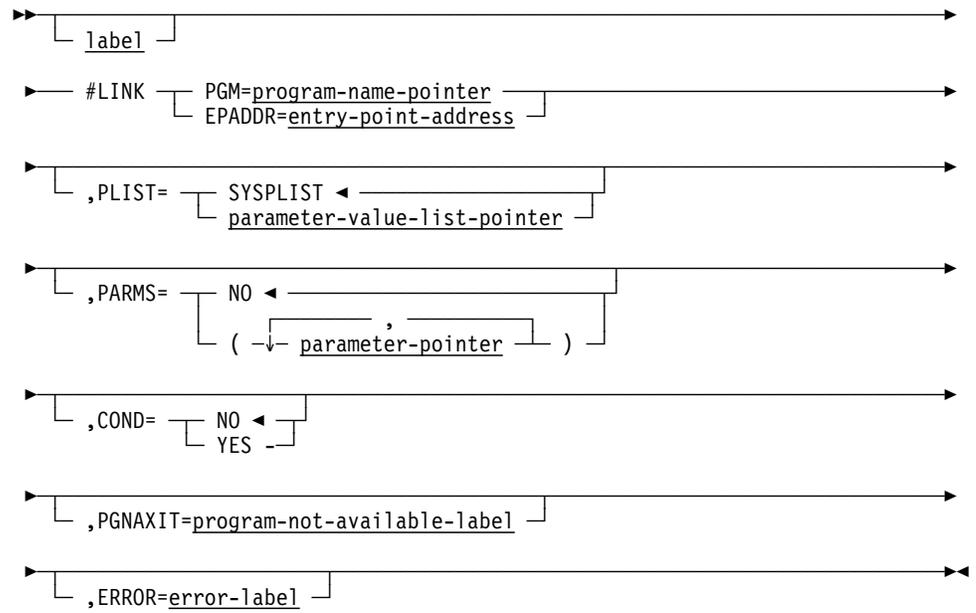
#LINEEND



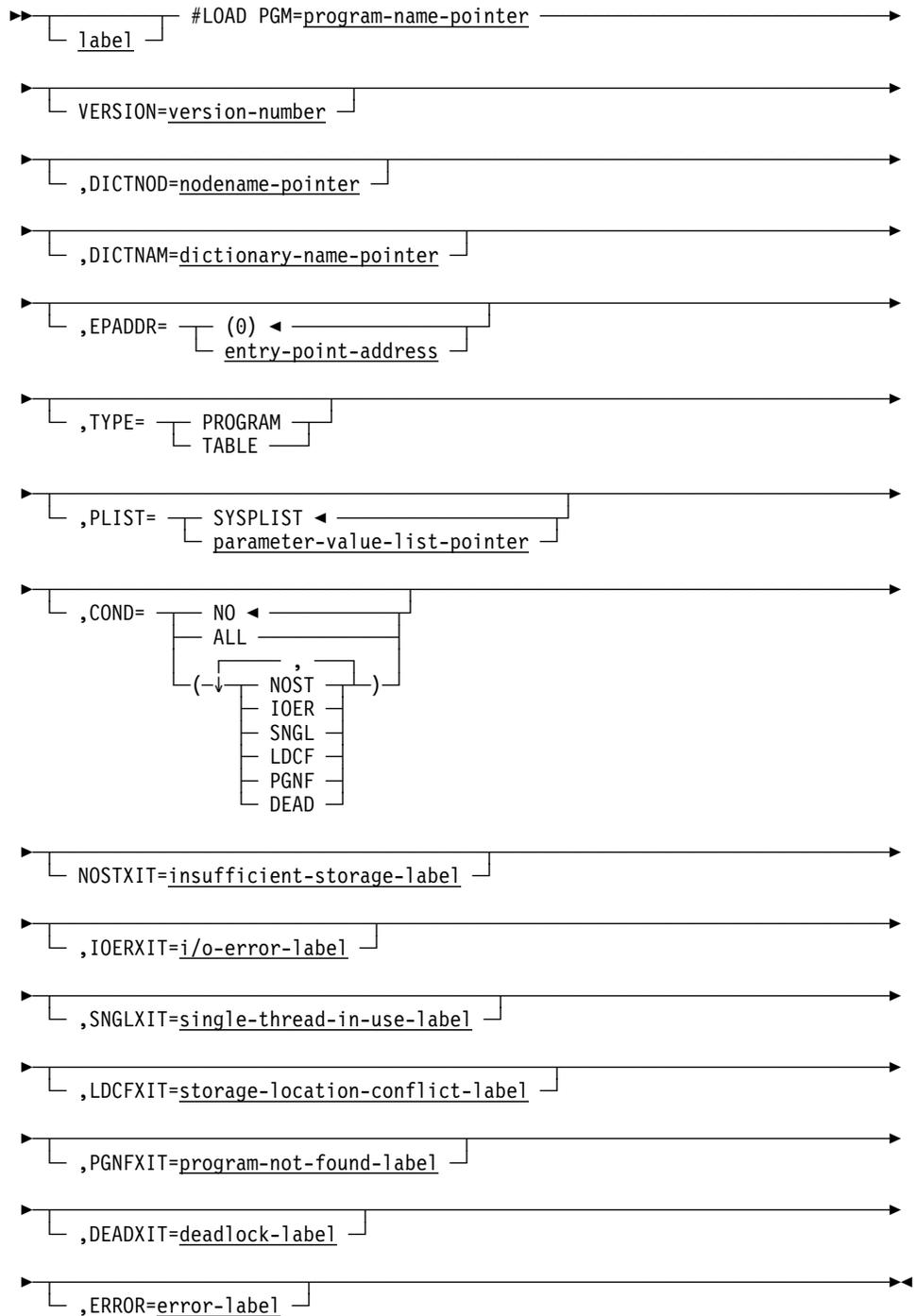
#LINEIN

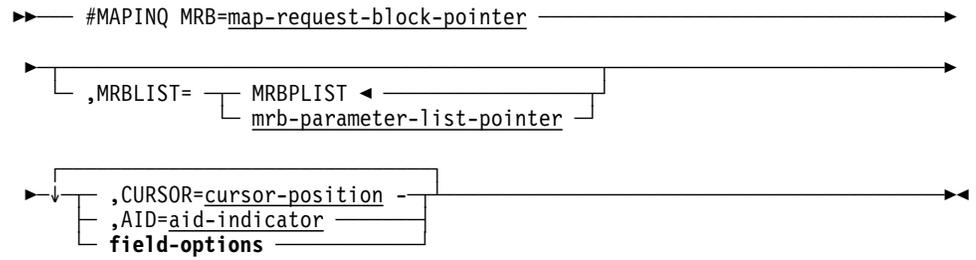
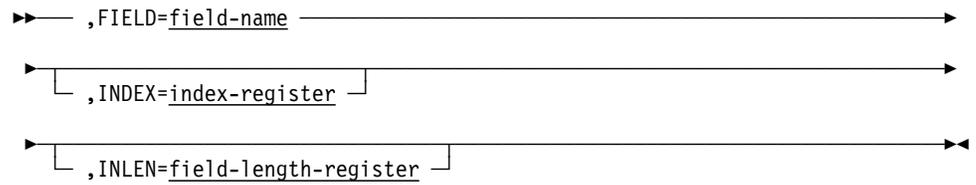
#LINEOUT



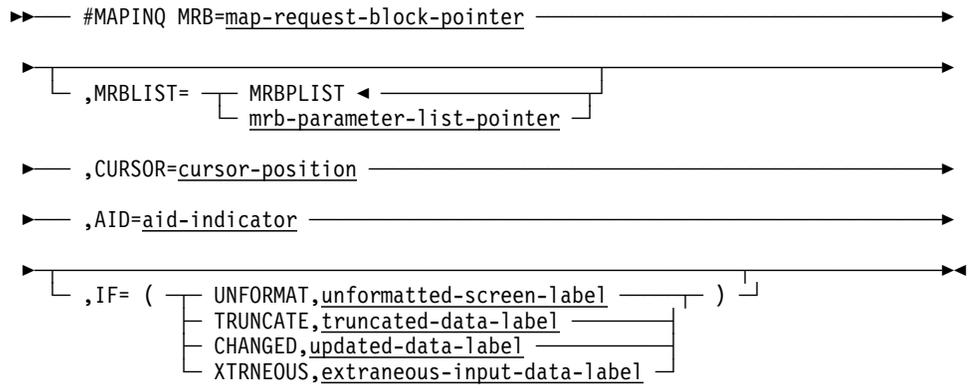
#LINK

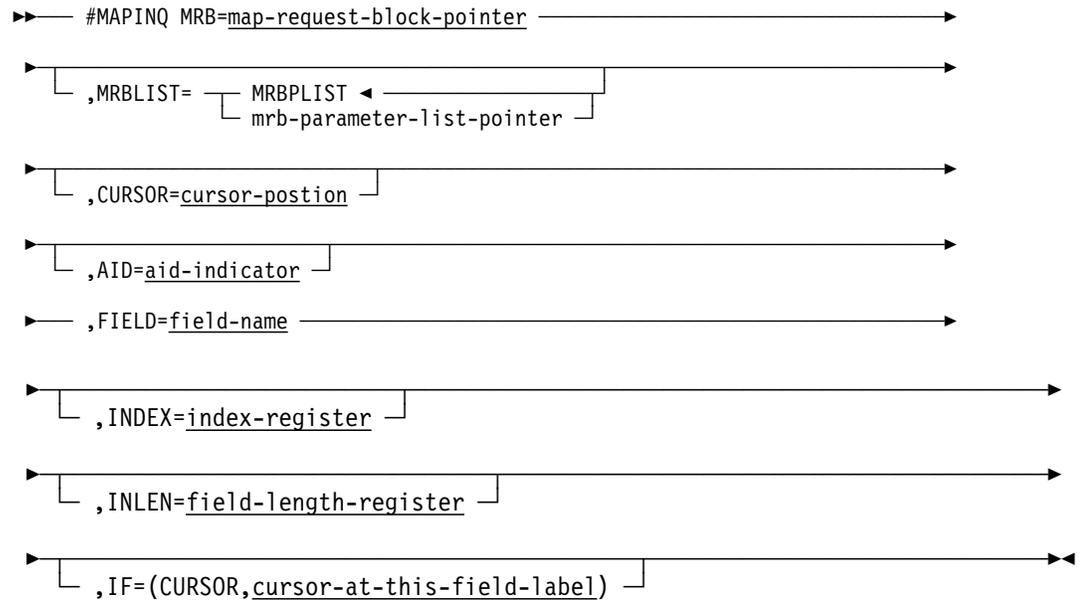
#LOAD



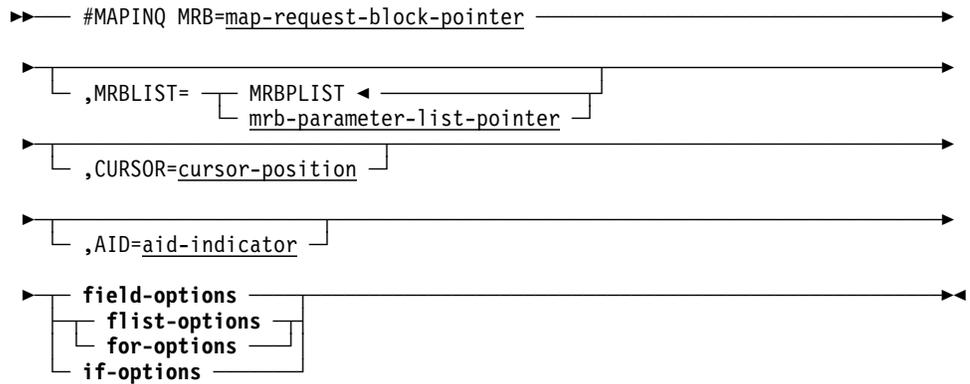
#MAPINQ (moving map-related data)**Expansion of field-options**

#MAPINQ (testing for global map input conditions)

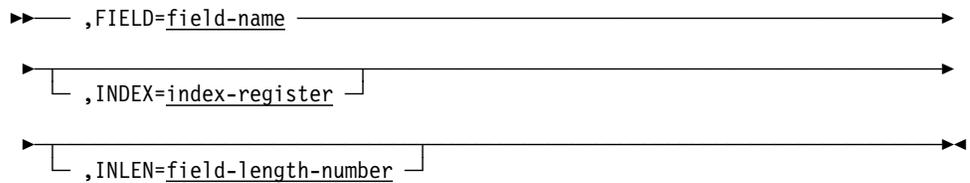


#MAPINQ (testing cursor position)

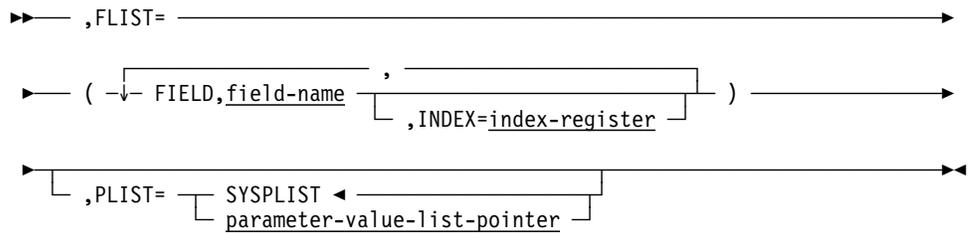
#MAPINQ (testing for input conditions)



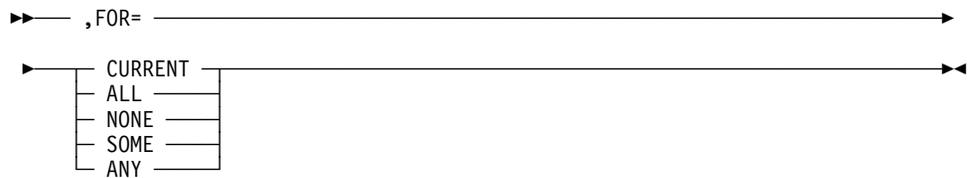
Expansion of field-options



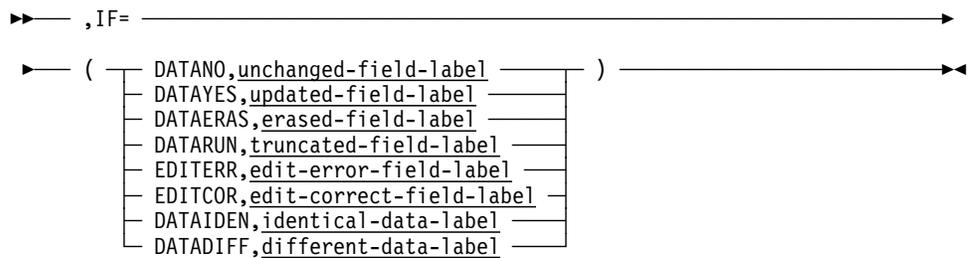
Expansion of FLIST-options

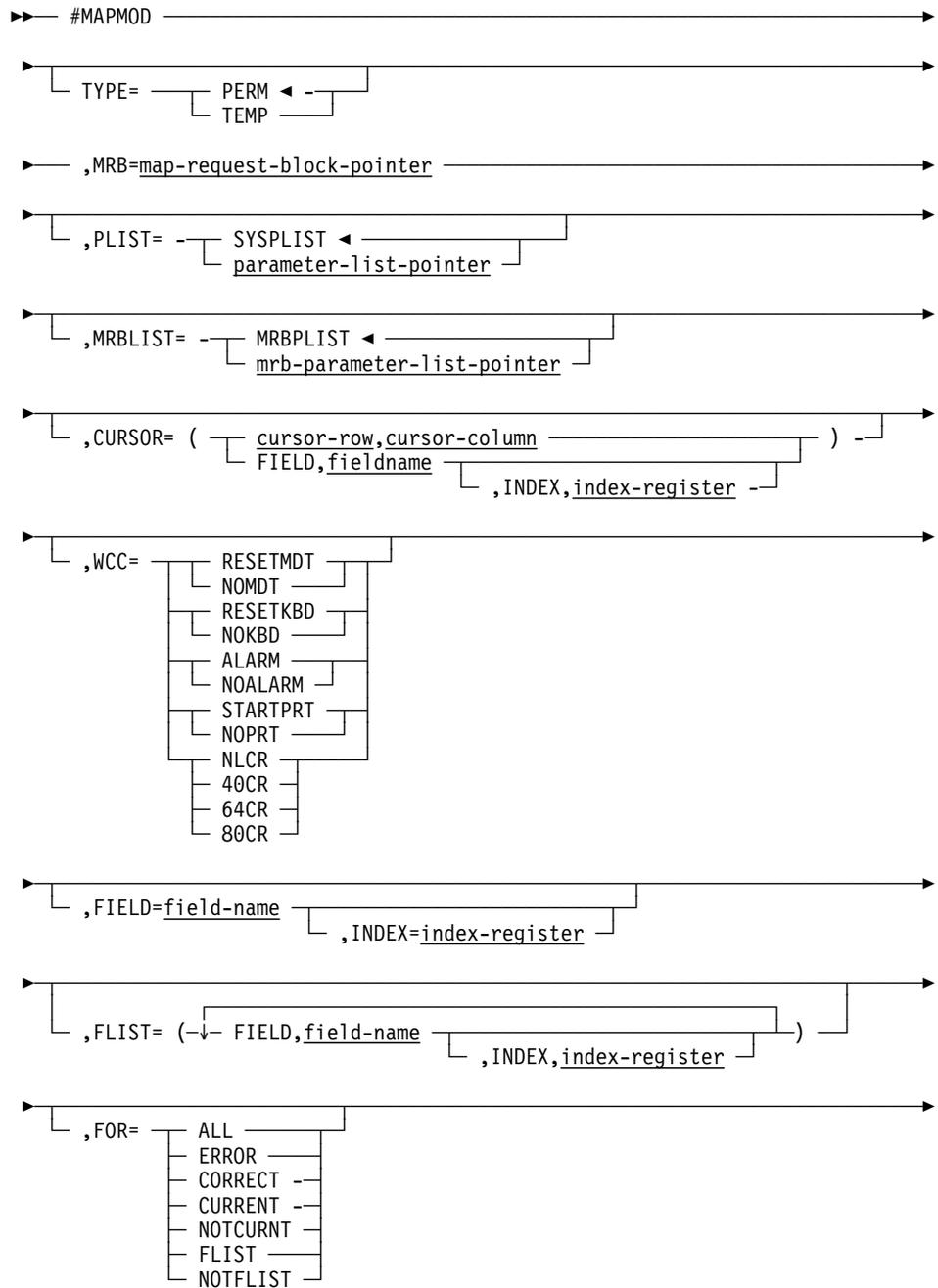


Expansion of for-options

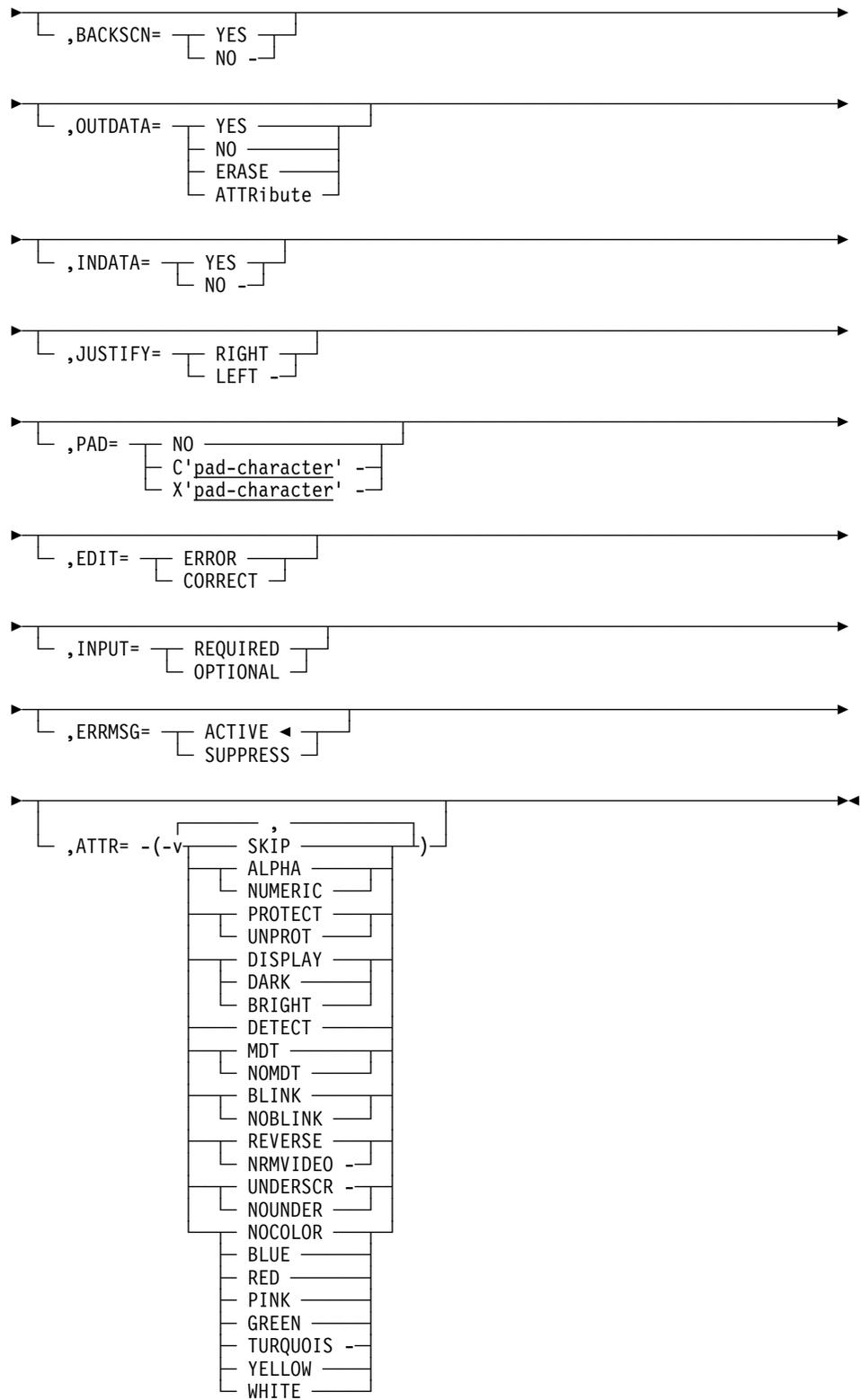


Expansion of if-options



#MAPMOD

4.3 Assembler DML statements



@MODIFY

▶— @MODIFY REC=record-name —▶

@MODIFY (LRF)

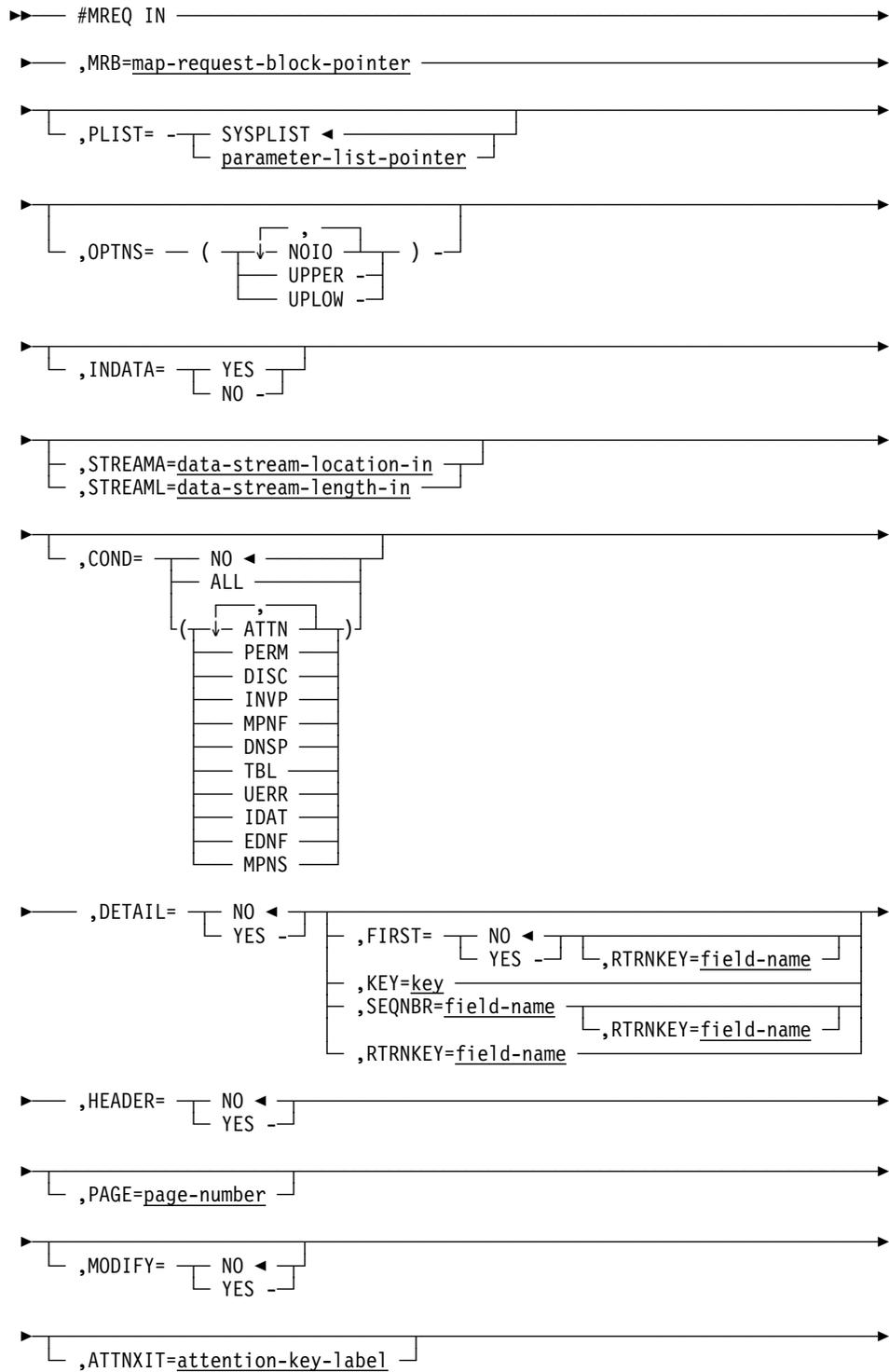
▶— @MODIFY REC=logical-record-name —▶

└ ,IOAREA=alt-logical-record-location ─┘

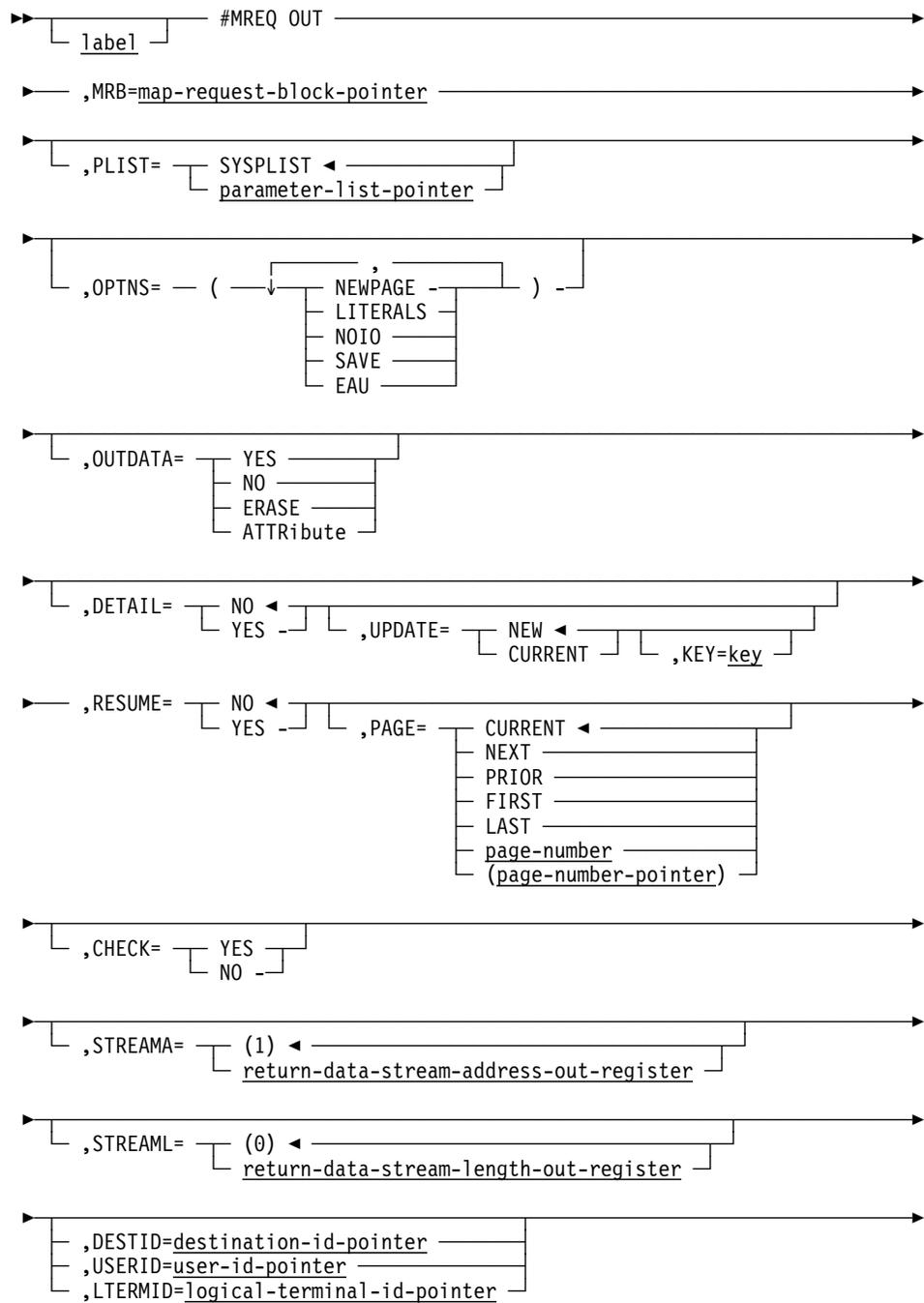
└ ,ONLRSTS=path-status,GOTO=branch-location -┘

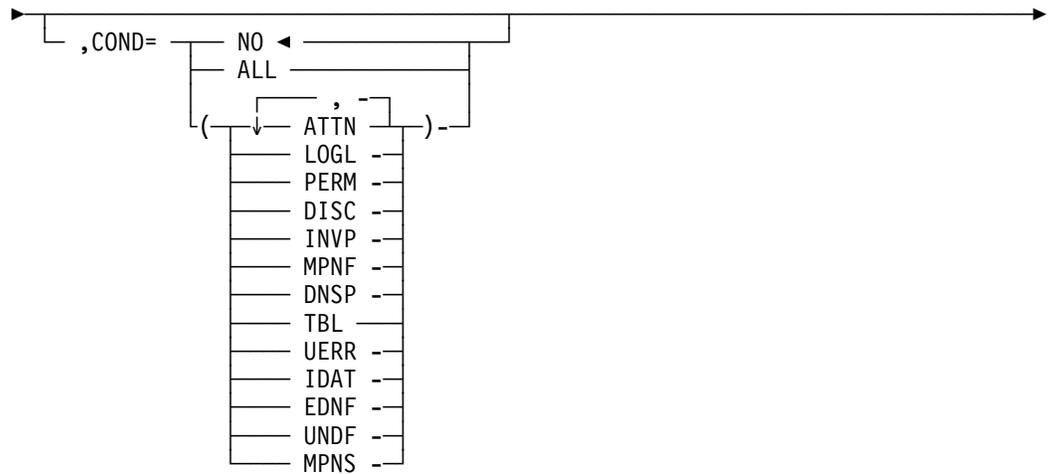
└ ,WHERE **boolean-expression** ─┘

#MREQ



| | | |
|---|--|---|
| ▶ | ┌ ,PERMXIT= <u>permanent-i/o-error-label</u> ┐ | ▶ |
| ▶ | ┌ ,DISCXIT= <u>terminal-disconnected-label</u> ┐ | ▶ |
| ▶ | ┌ ,INVPXIT= <u>invalid-mrb-information-label</u> ┐ | ▶ |
| ▶ | ┌ ,MPNFXIT= <u>map-not-found-label</u> ┐ | ▶ |
| ▶ | ┌ ,DNSPXIT= <u>terminal-device-not-supported-label</u> ┐ | ▶ |
| ▶ | ┌ ,TBLXIT= <u>error-in-table-label</u> ┐ | ▶ |
| ▶ | ┌ ,UERRXIT= <u>error-in-return-user-edit-mod-label</u> ┐ | ▶ |
| ▶ | ┌ ,IDATXIT= <u>internal-data-error-label</u> ┐ | ▶ |
| ▶ | ┌ ,EDNFXIT= <u>edit-module-not-found-label</u> ┐ | ▶ |
| ▶ | ┌ ,MPNSXIT= <u>paging-session-error-label</u> ┐ | ▶ |
| ▶ | ┌ ,ERROR= <u>error-label</u> ┐ | ▶ |





└── ,ATTXIT=attention-key-label

└── ,LOGLXIT=logical-output-error-label

└── ,PERMXIT=permanent-i/o-error-label

└── ,DISCXIT=terminal-disconnected-label

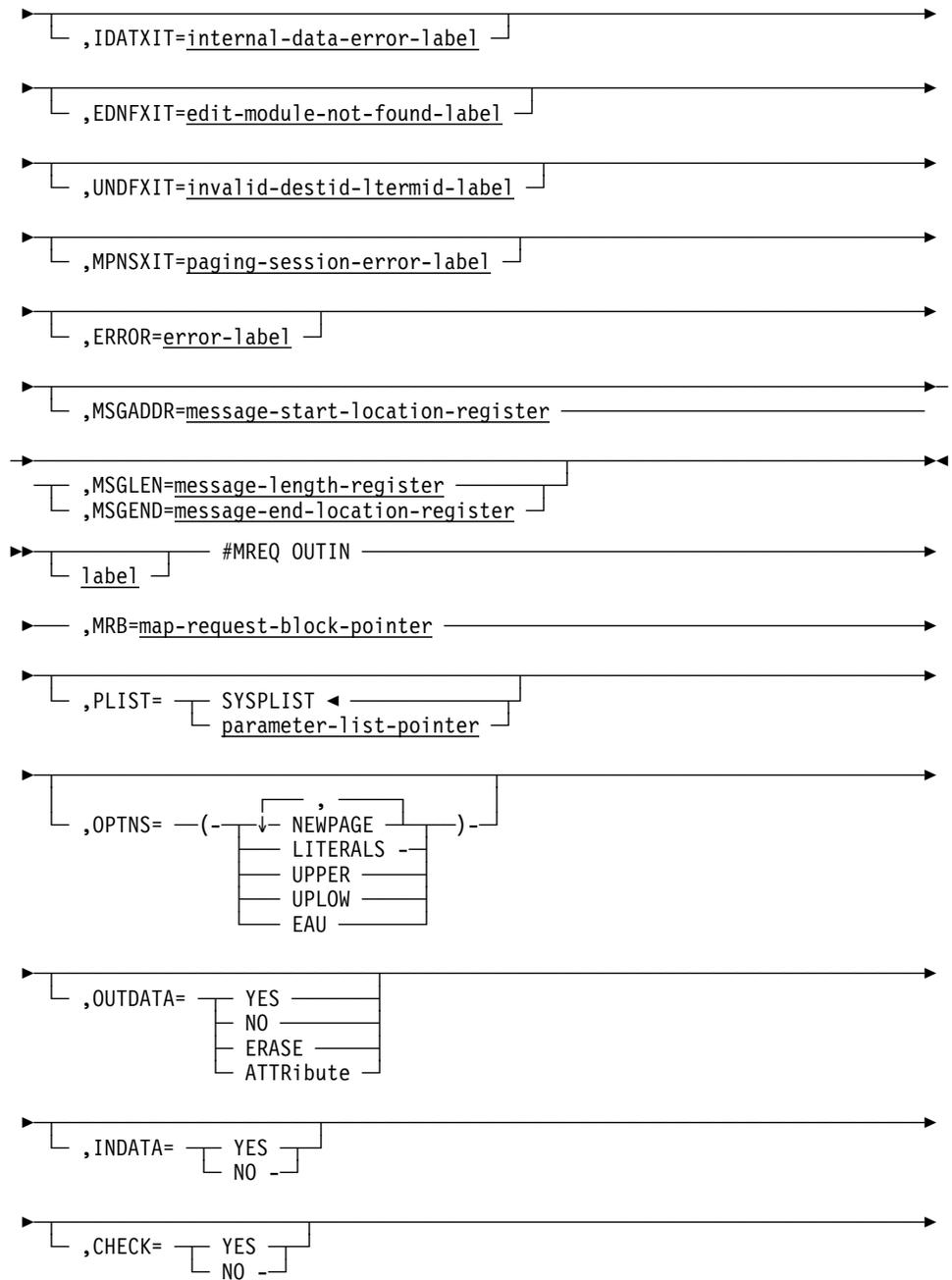
└── ,INVPXIT=invalid-mrb-information-label

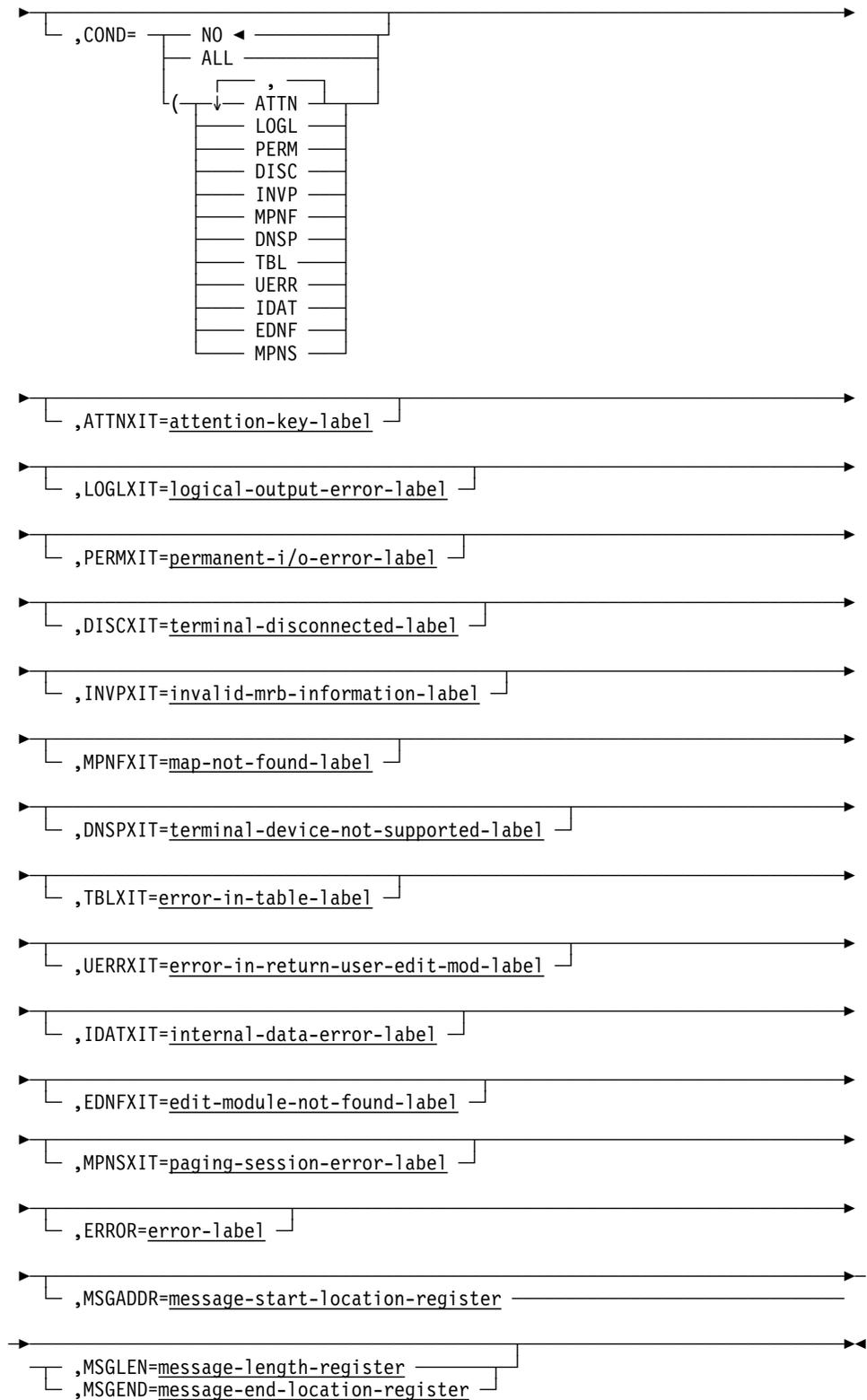
└── ,MPNFXIT=map-not-found-label

└── ,DNSPXIT=terminal-device-not-supported-label

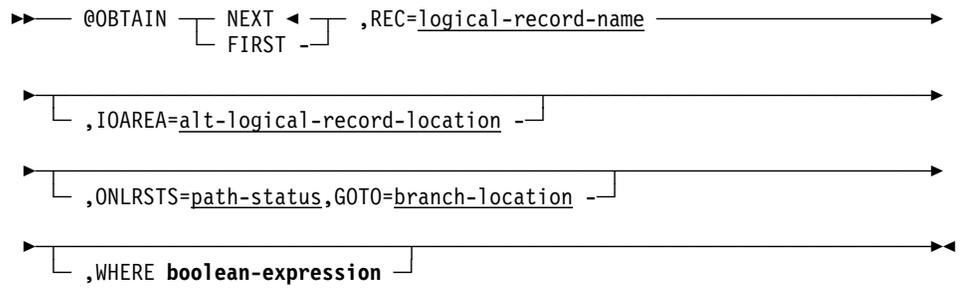
└── ,TBLXIT=error-in-table-label

└── ,UERRXIT=error-in-return-user-edit-mod-label

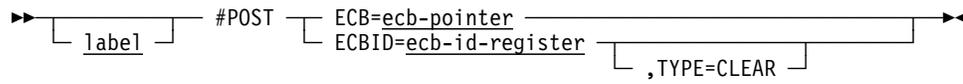


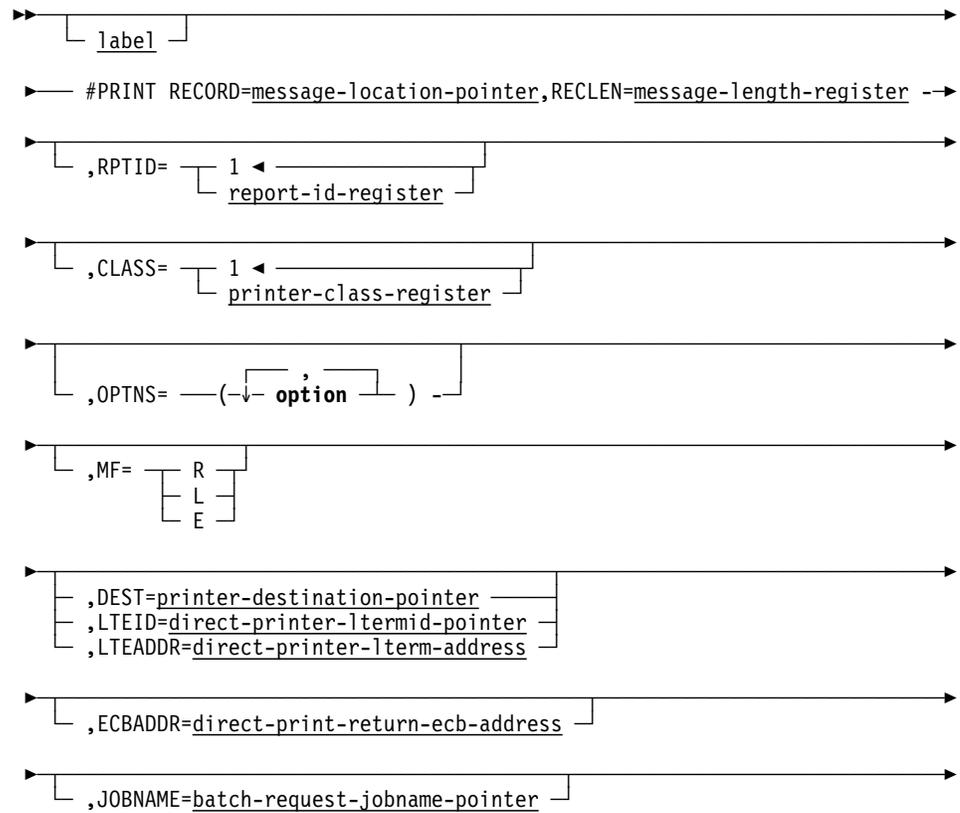


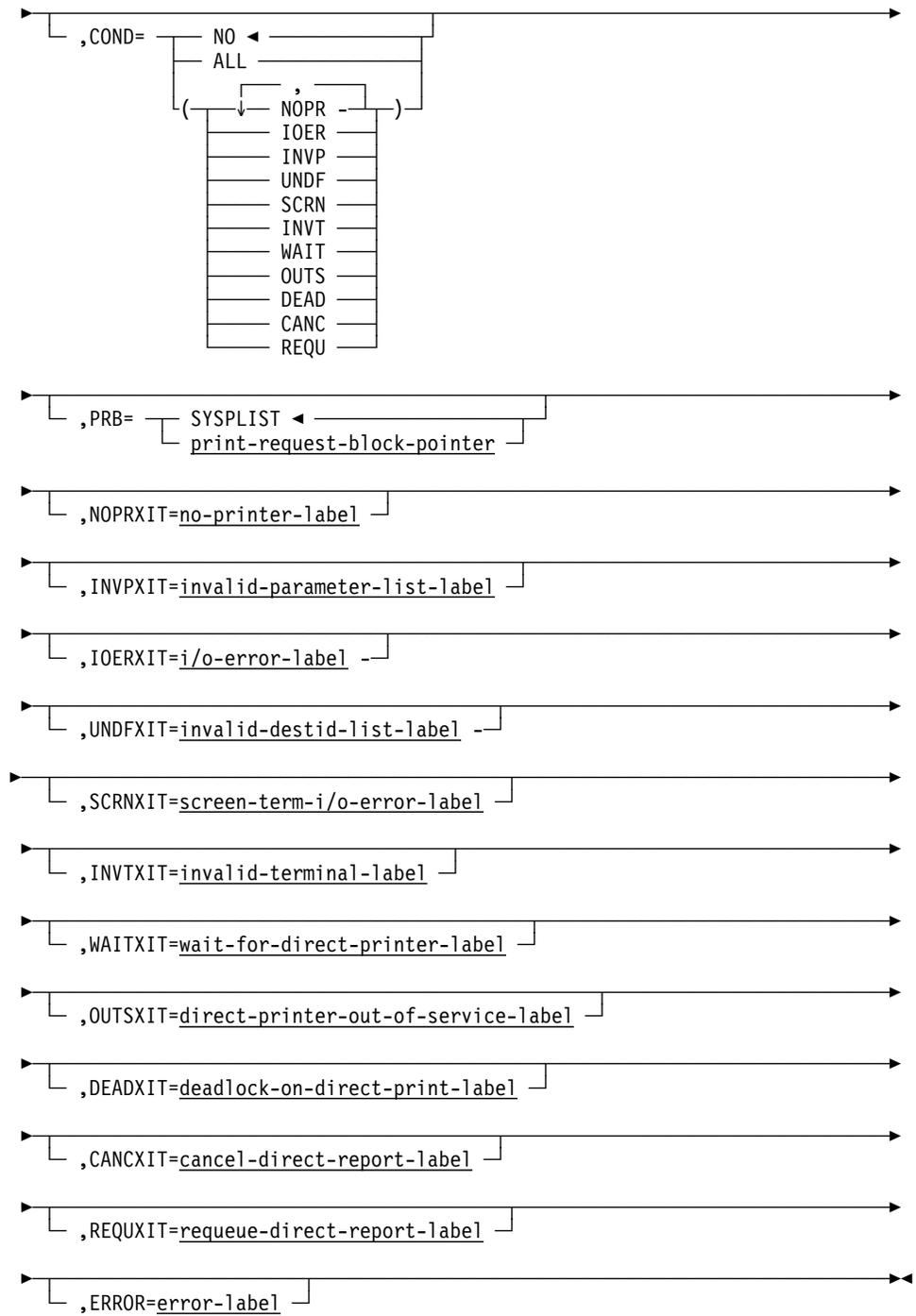
@OBTAIN (LRF)

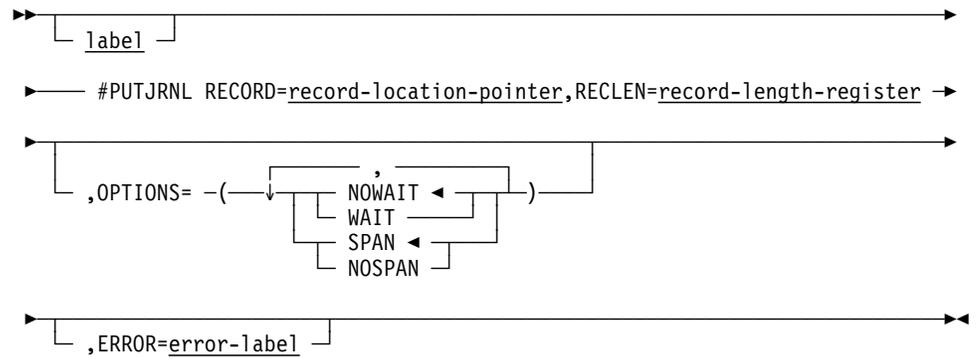
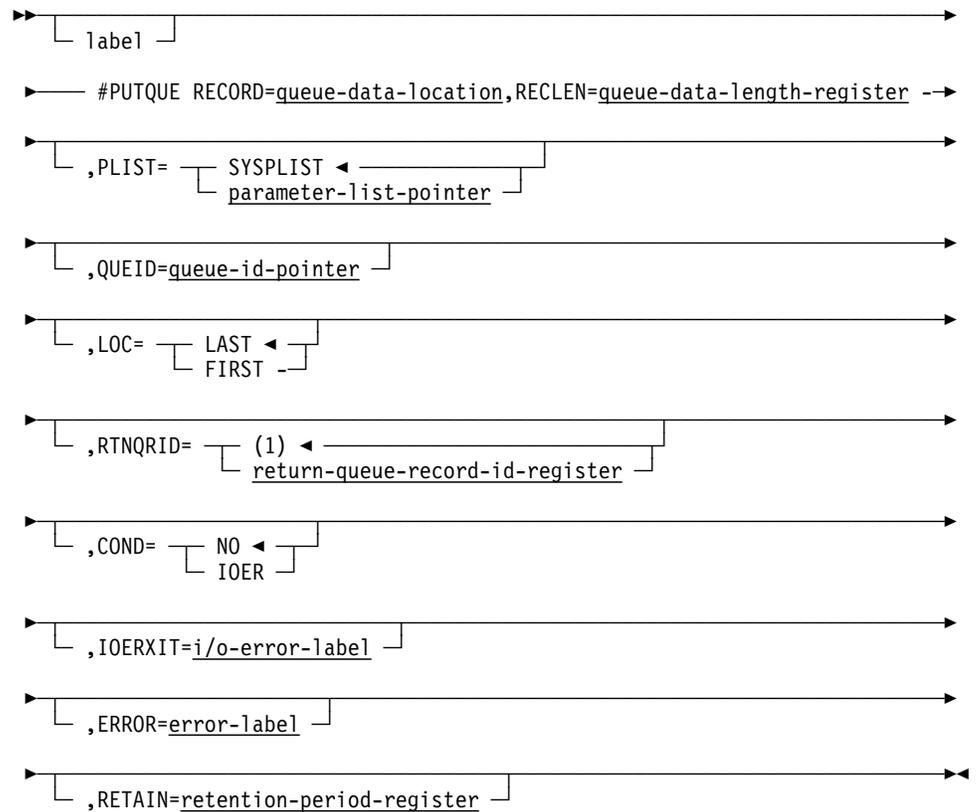


#POST

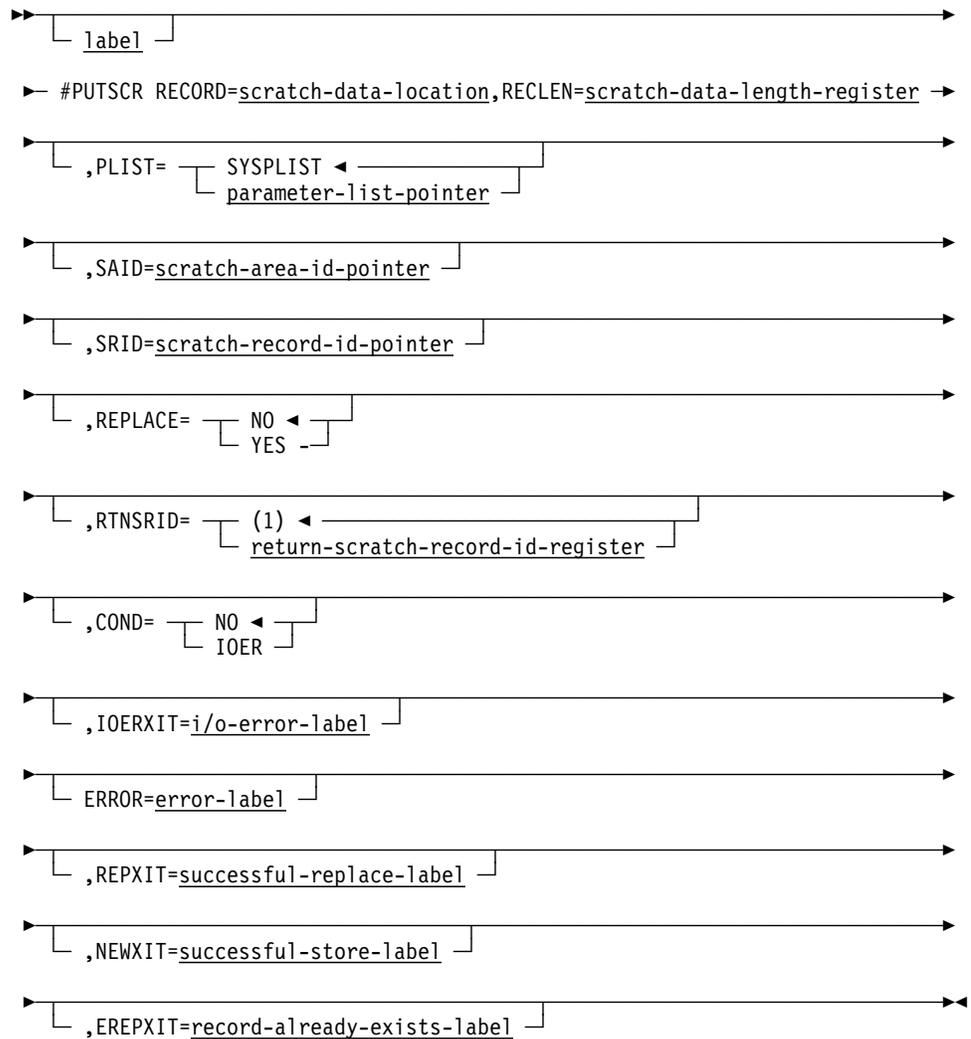


#PRINT

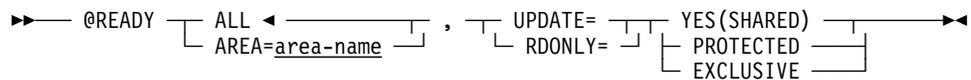


#PUTJRNL**#PUTQUE**

#PUTSCR

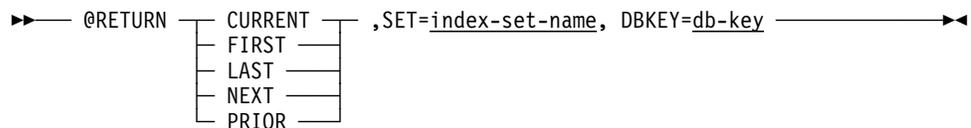


@READY

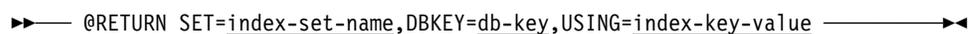


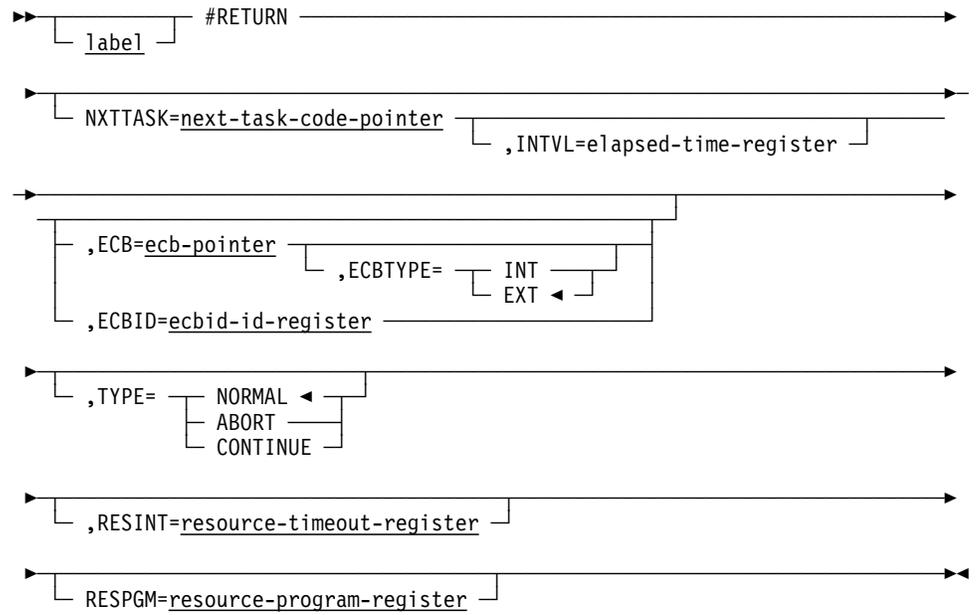
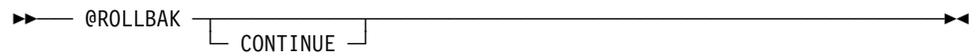
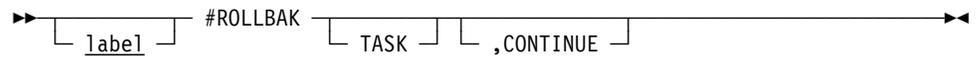
@RETURN

Navigational @RETURN

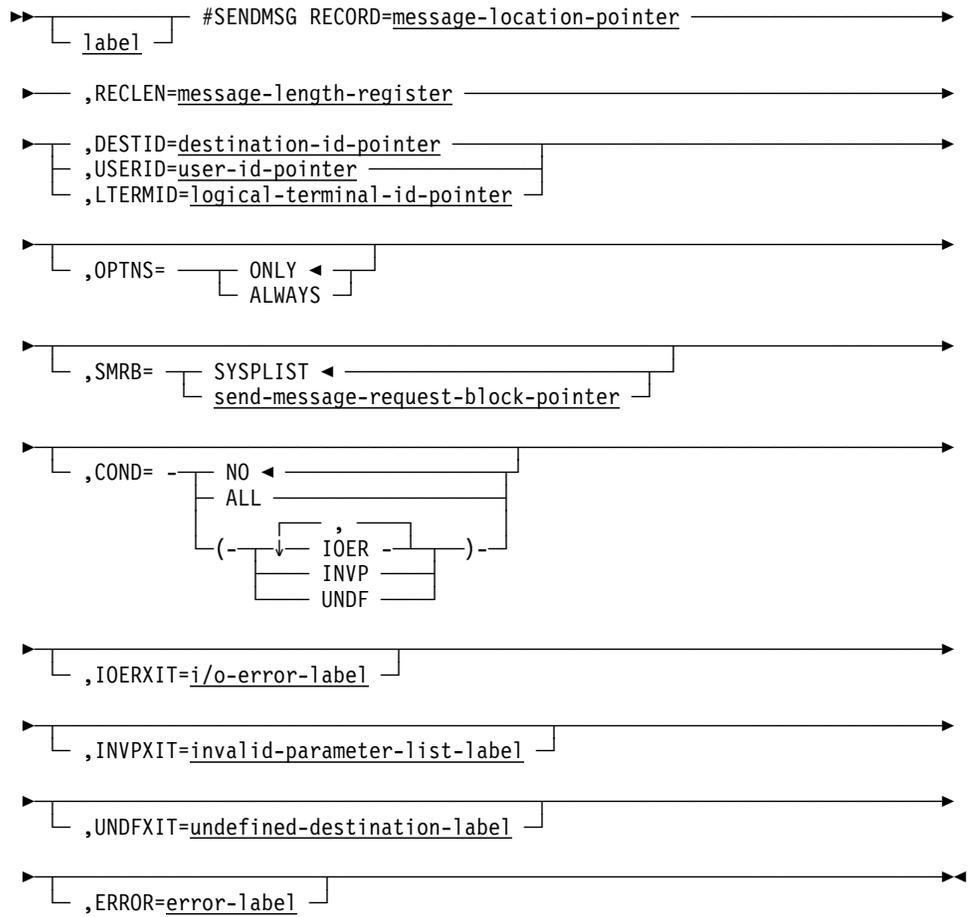


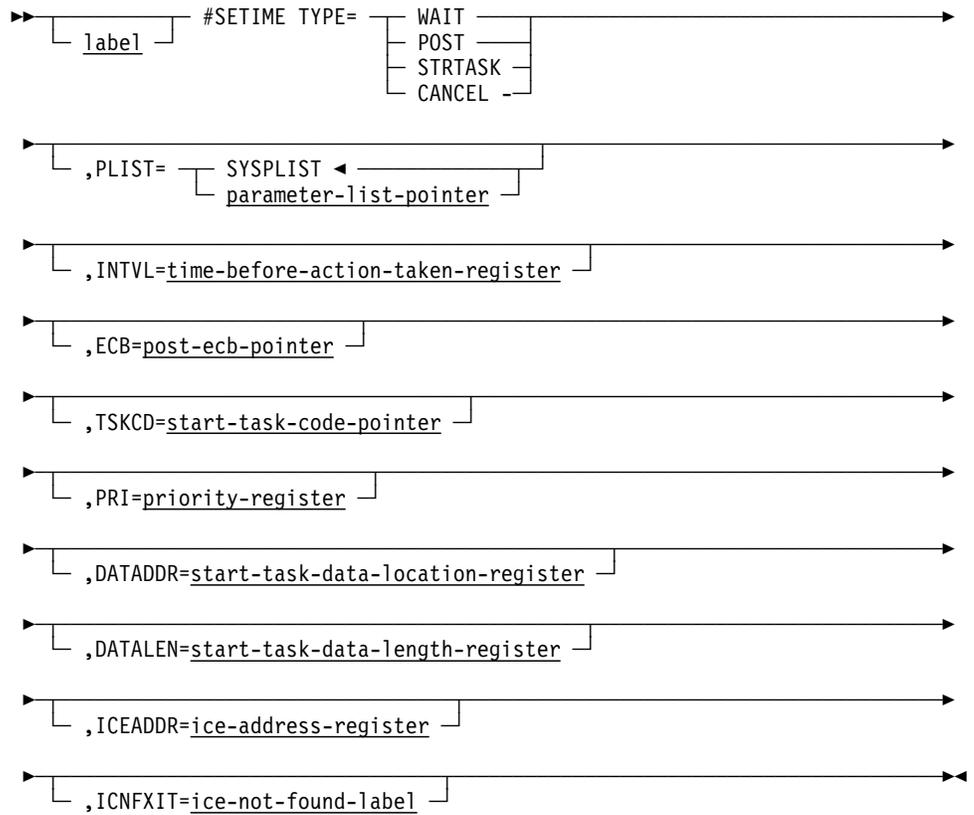
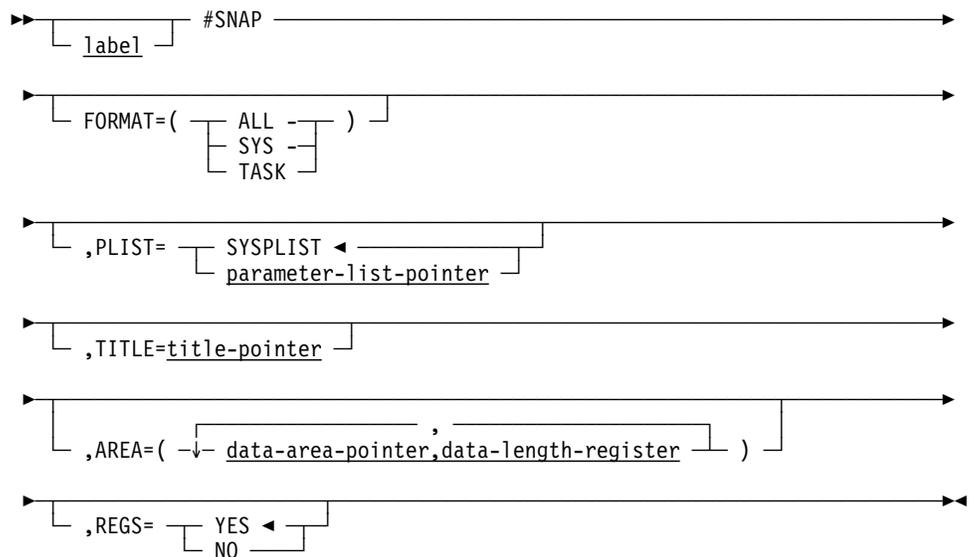
LRF @RETURN



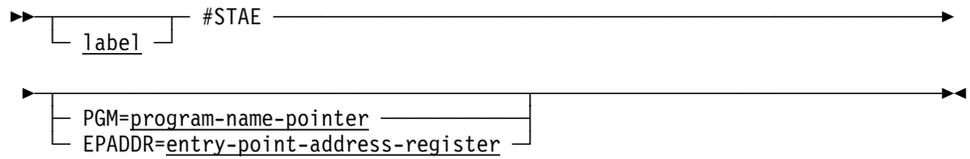
#RETURN**@ROLLBAK****#ROLLBAK**

#SENDMSG

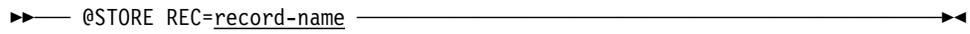


#SETIME**#SNAP**

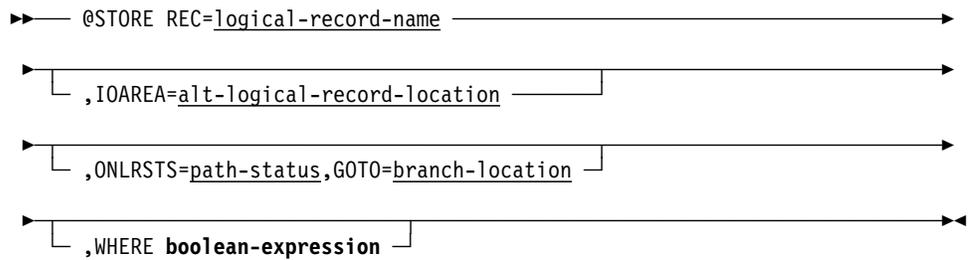
#STAE



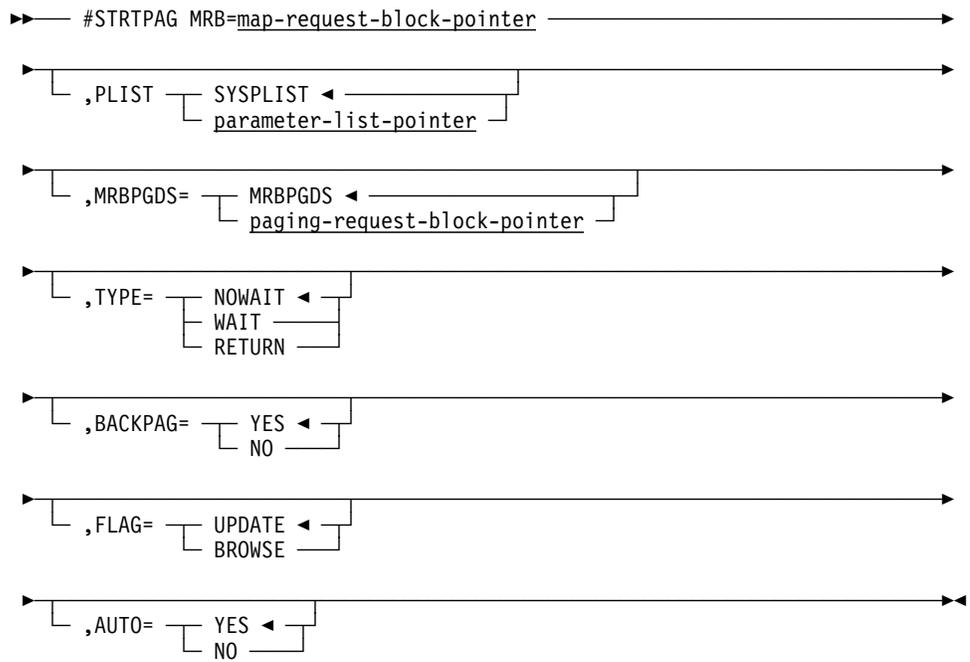
@STORE

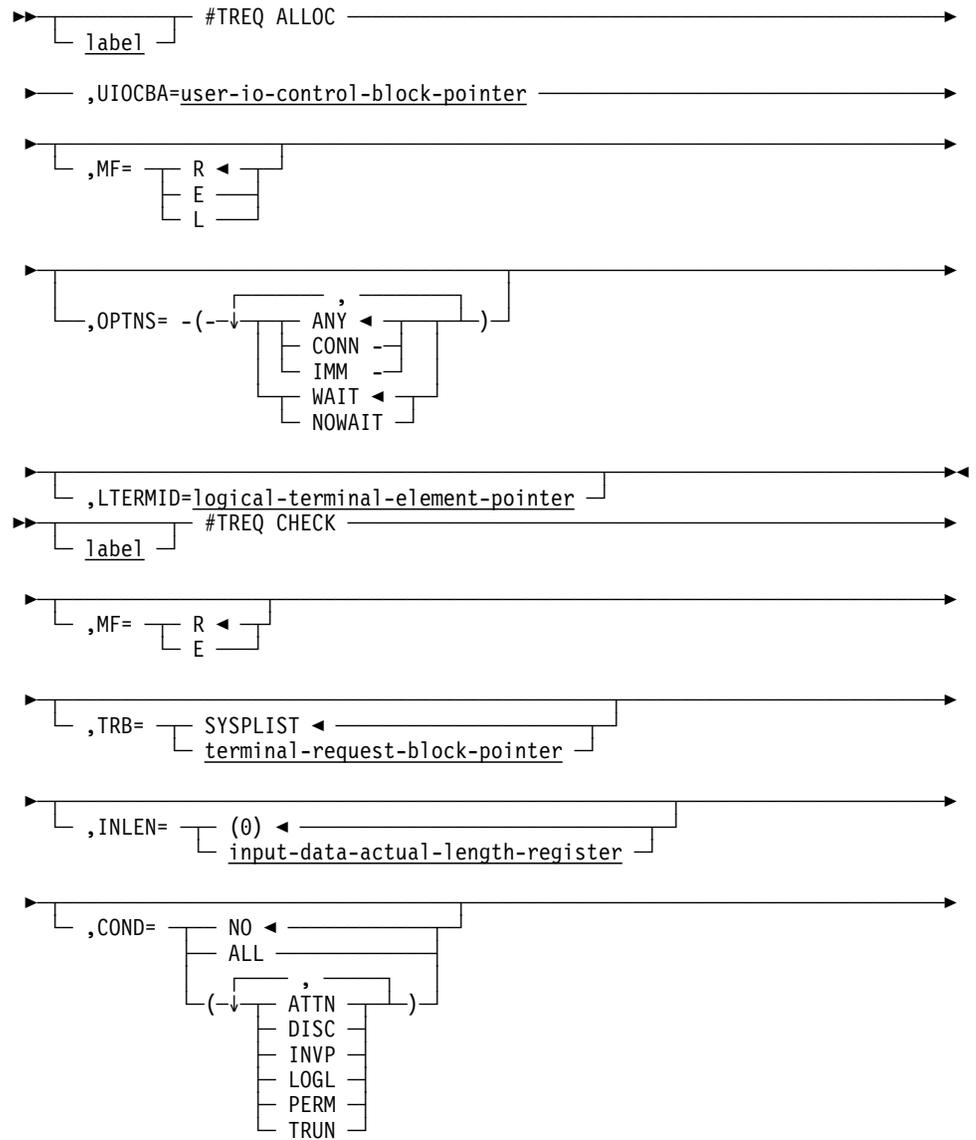


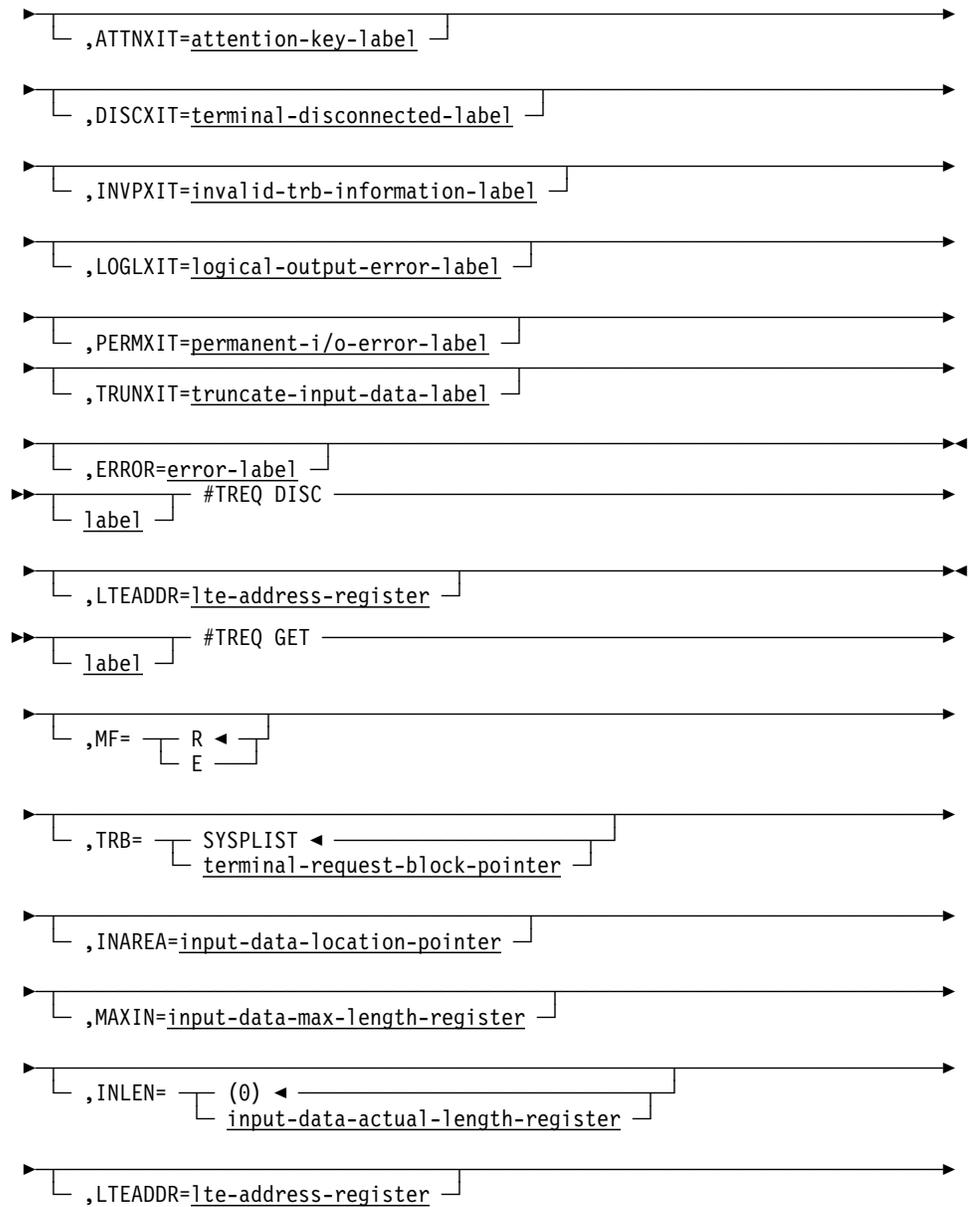
@STORE (LRF)

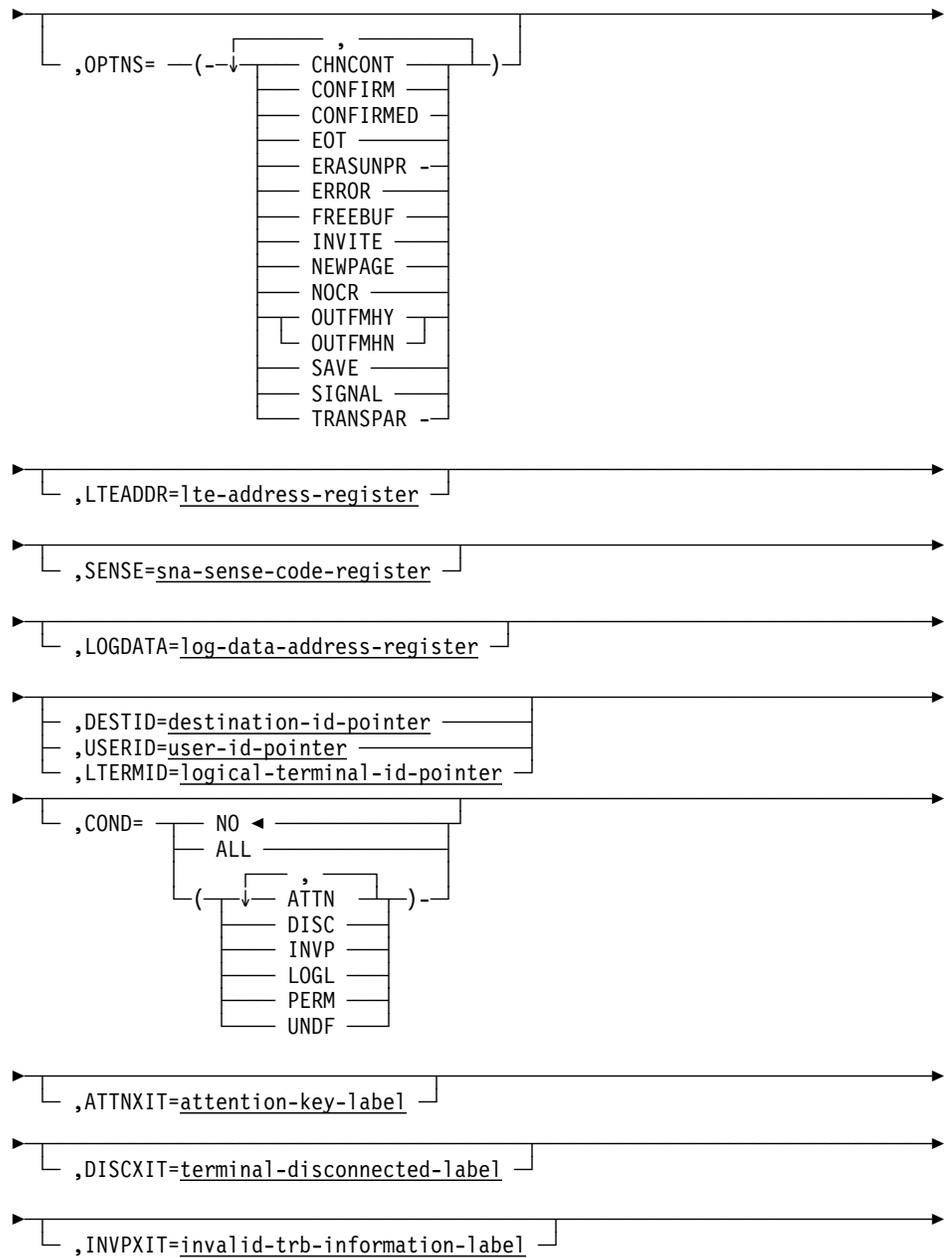


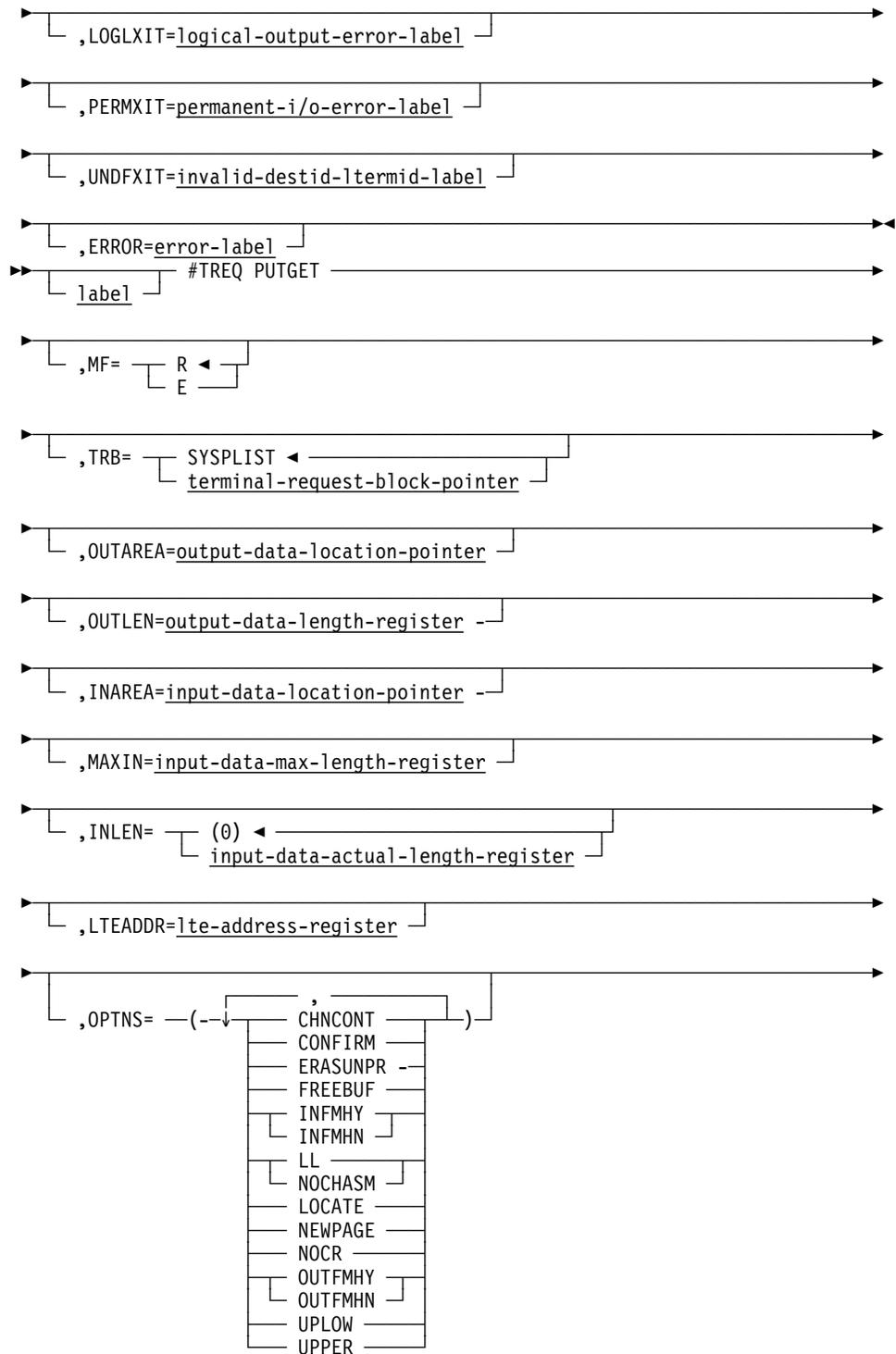
#STRTPAG

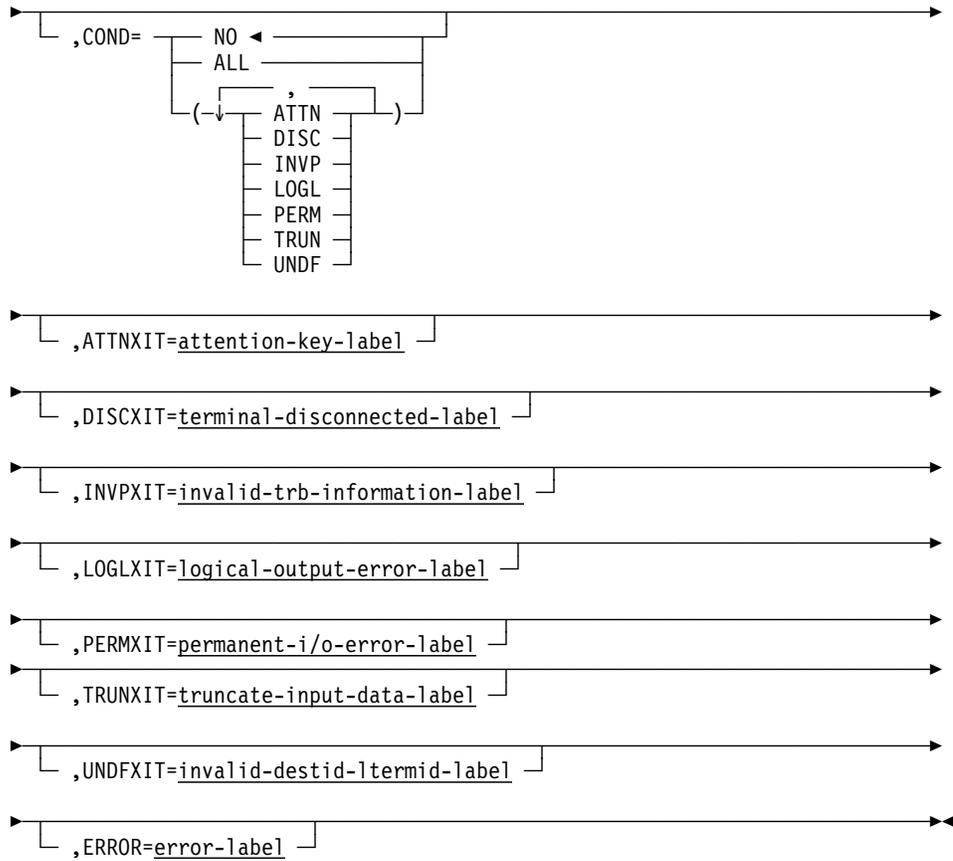


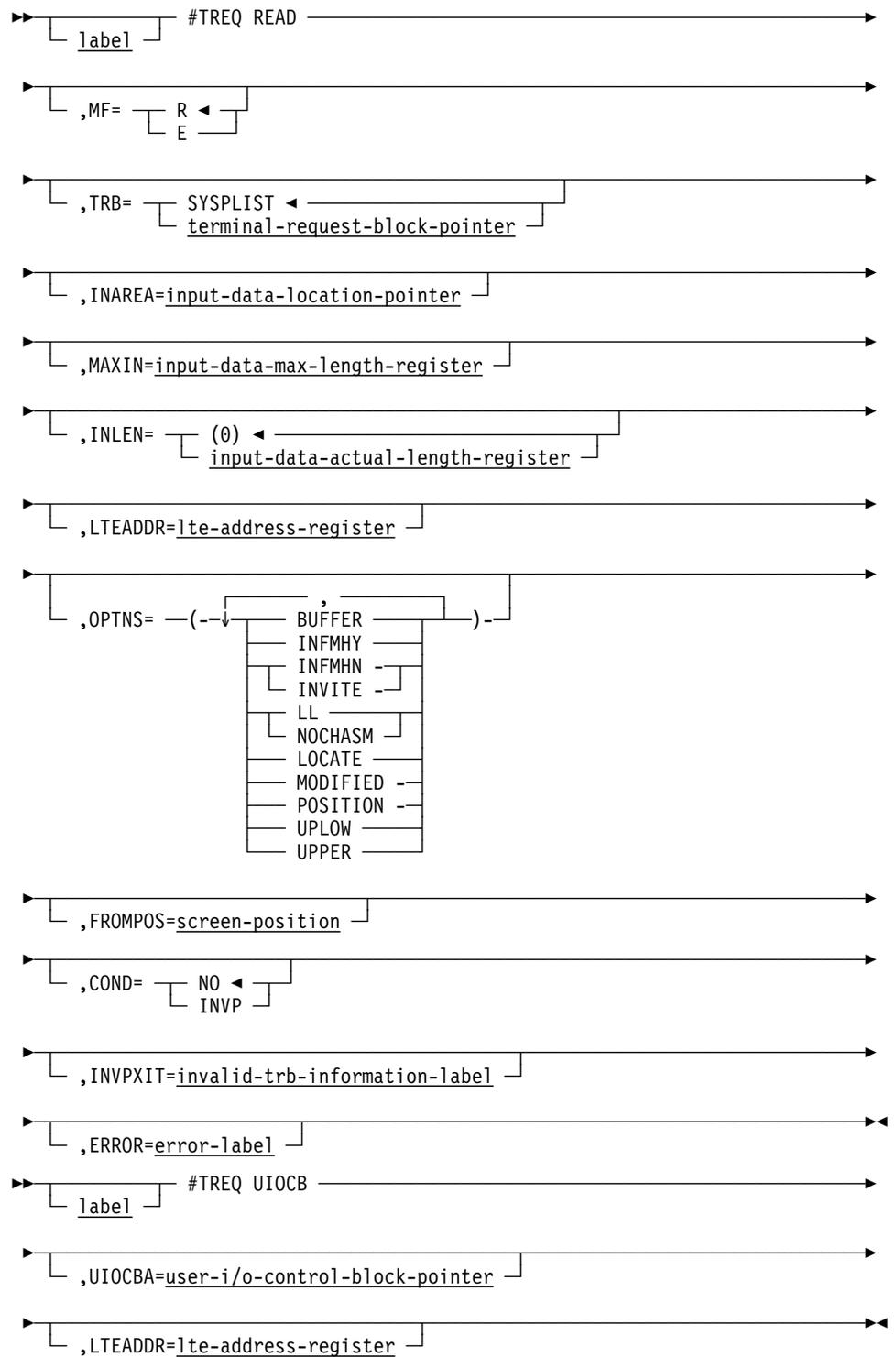
#TREQ



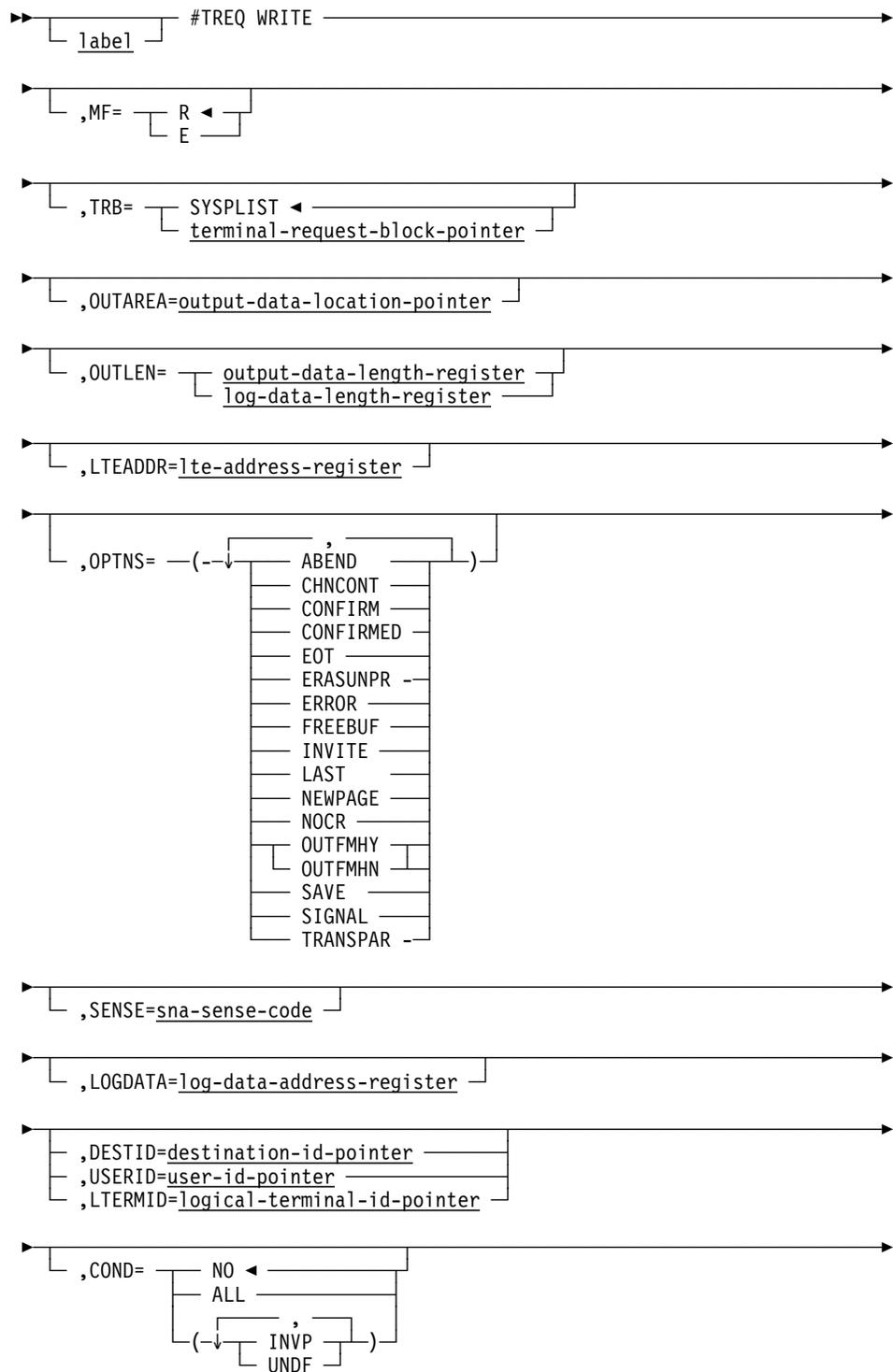


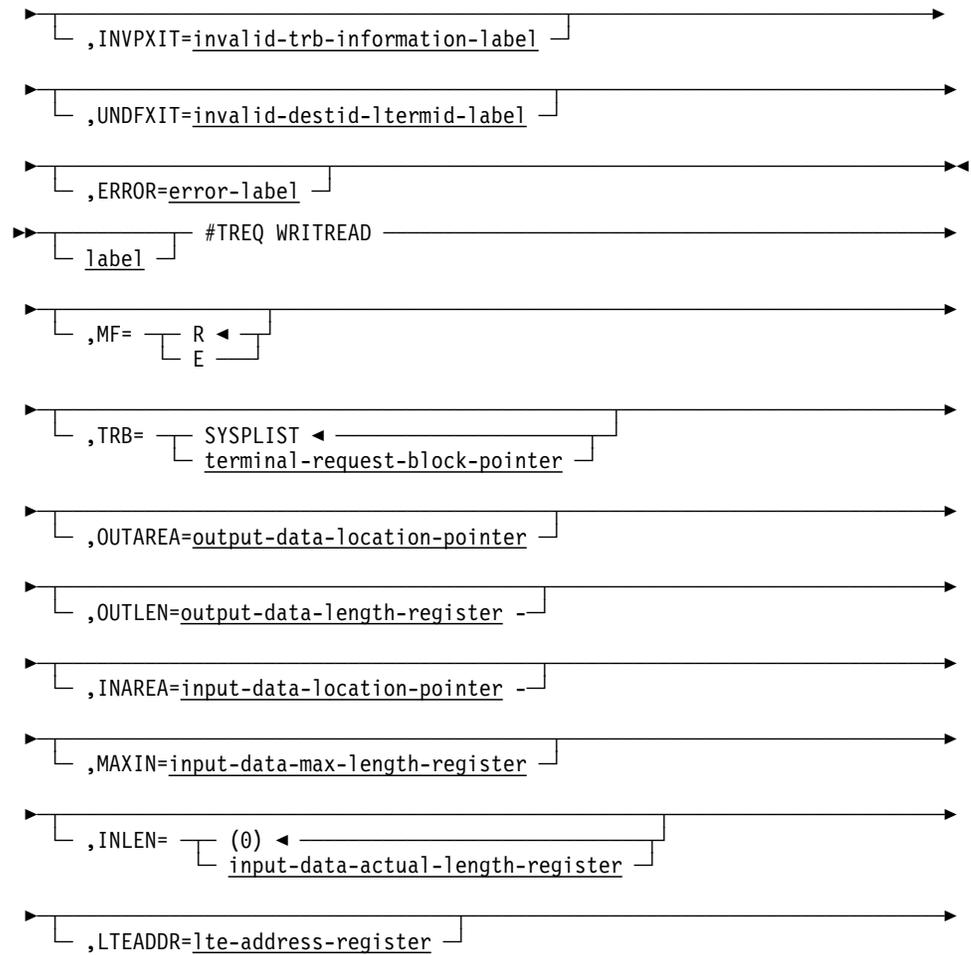


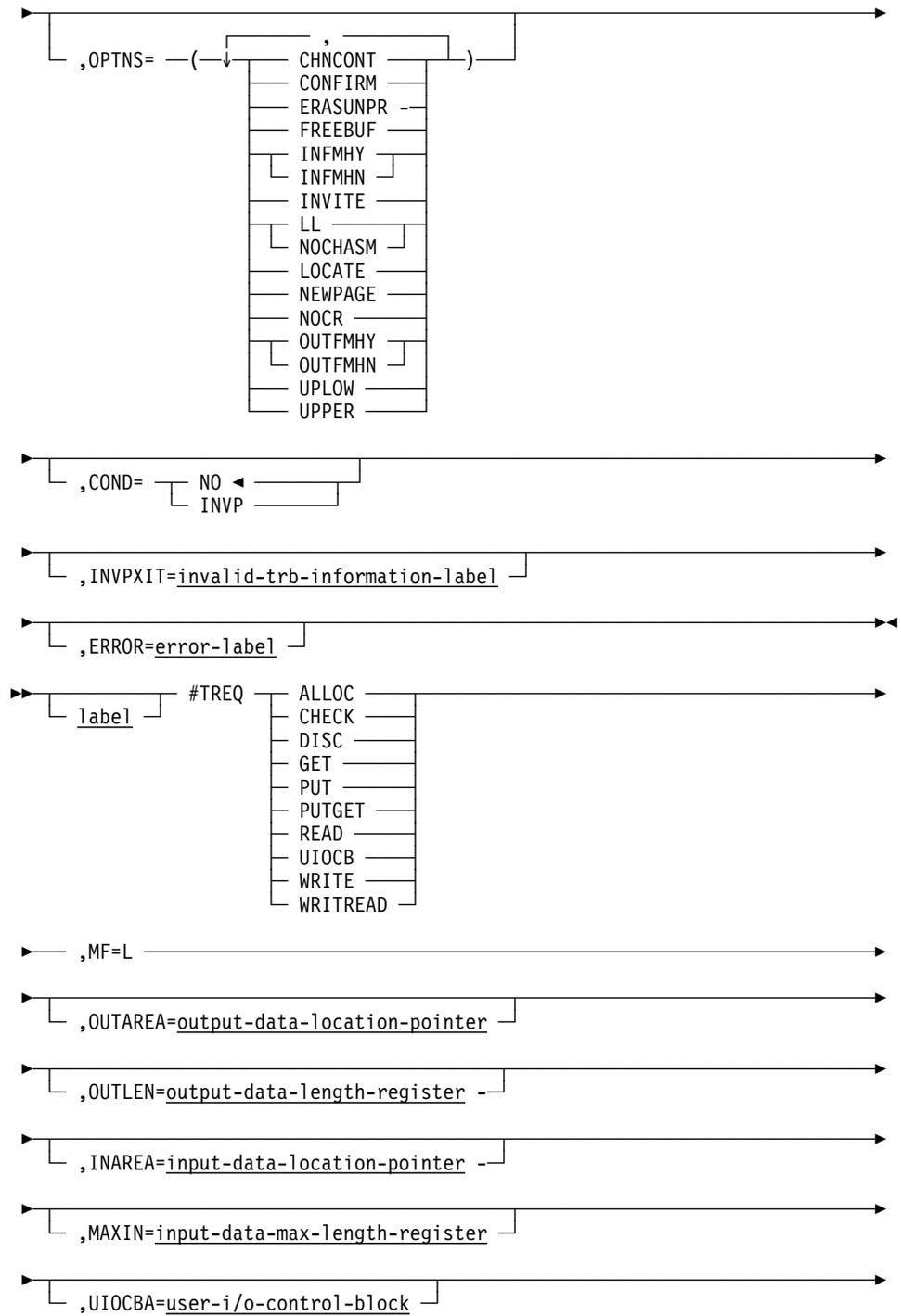


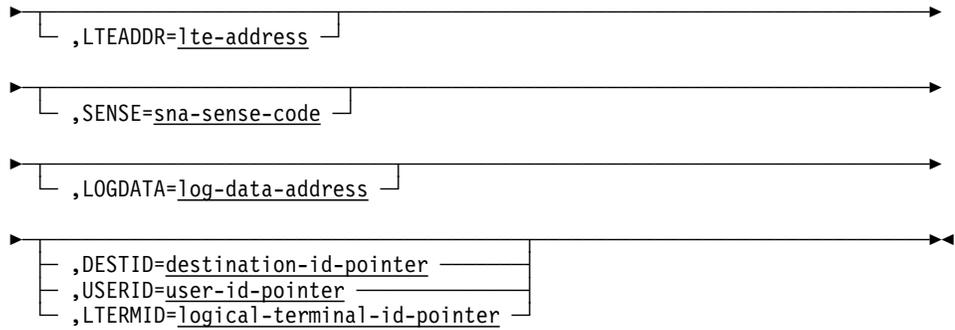


4.3 Assembler DML statements

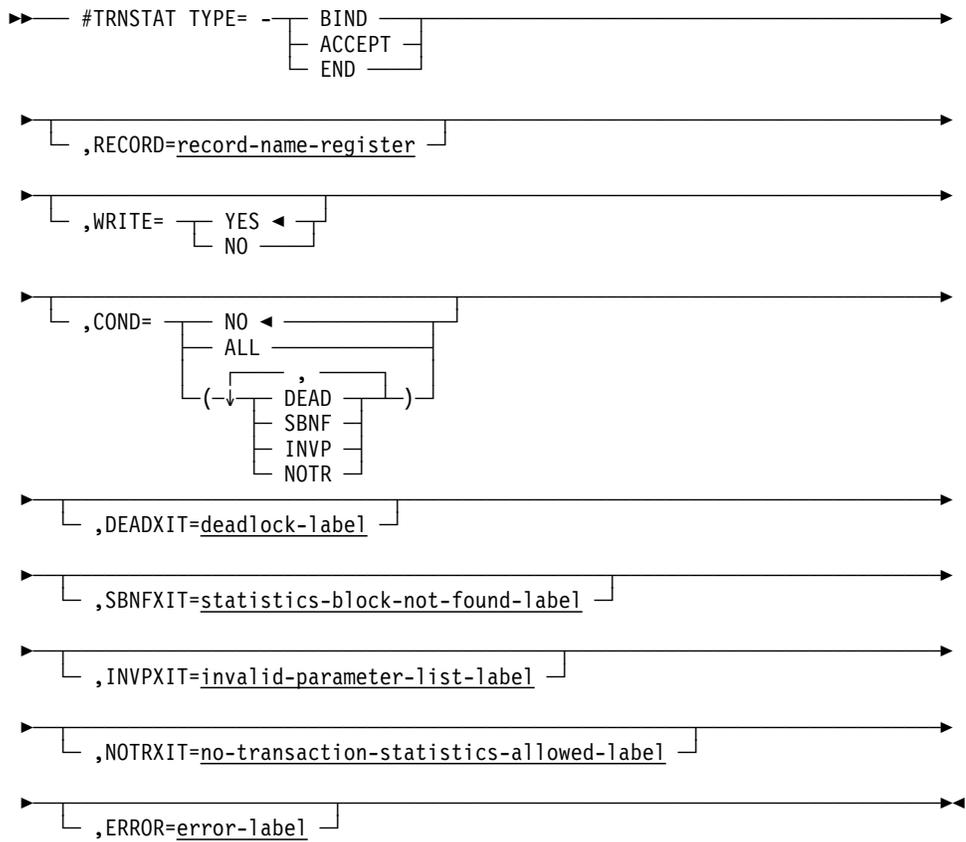




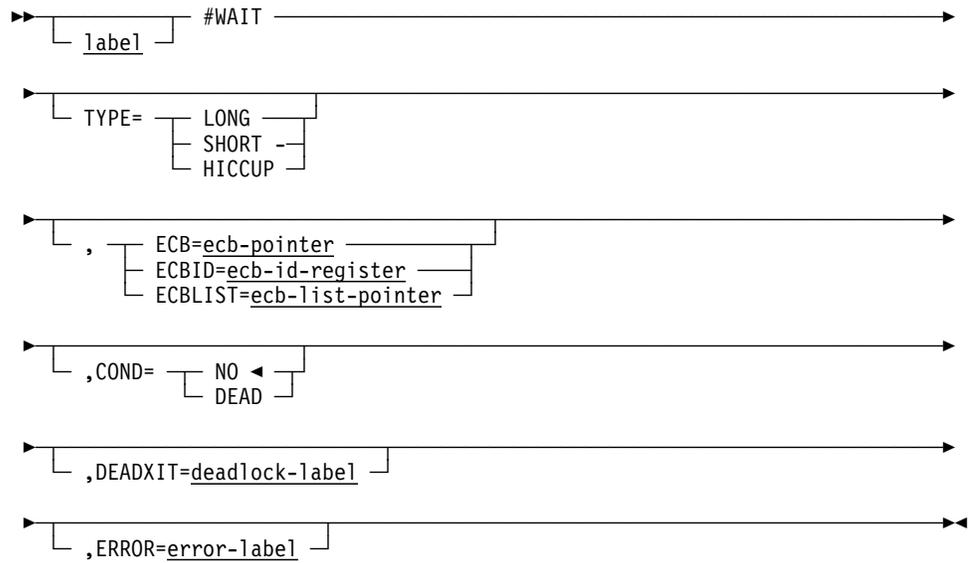




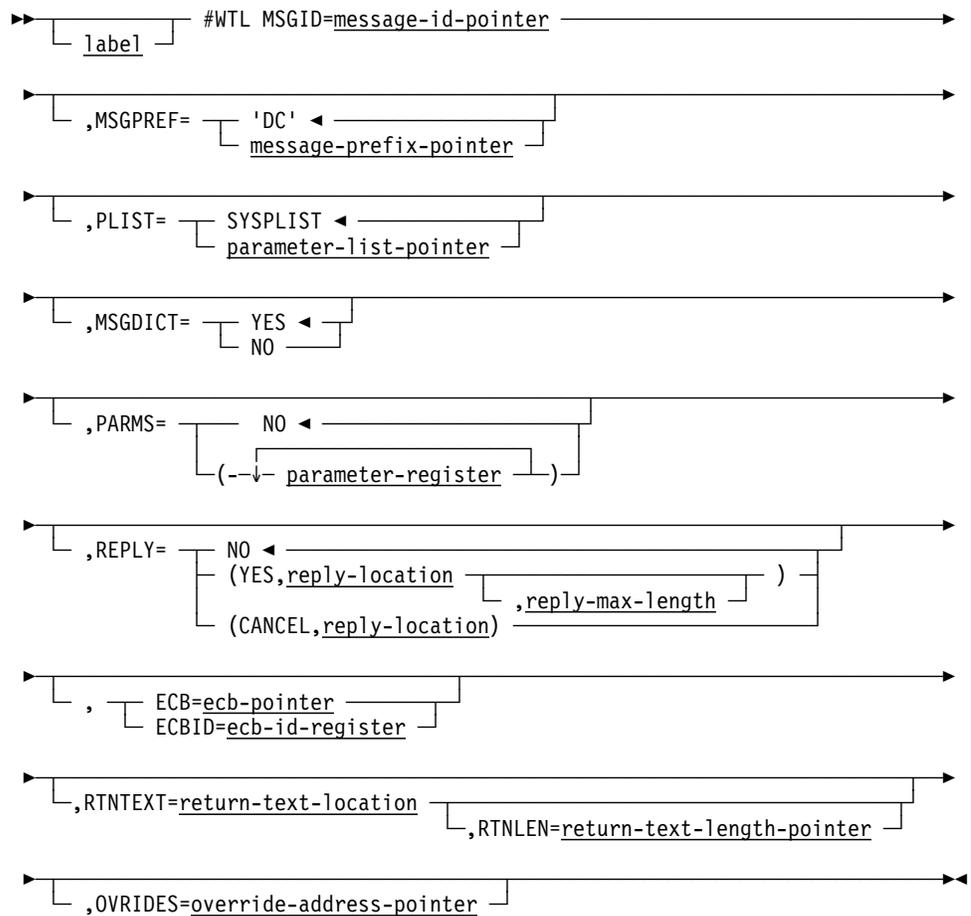
#TRNSTAT



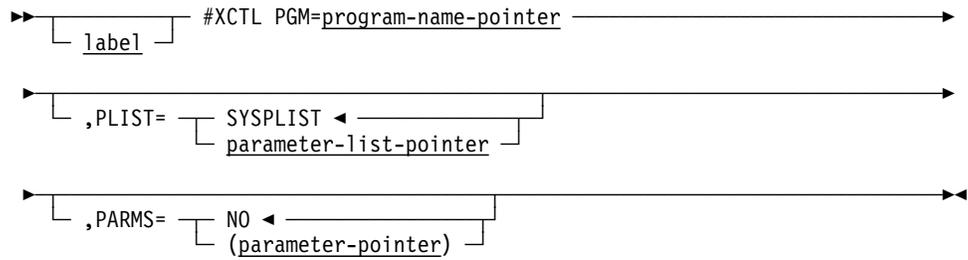
#WAIT



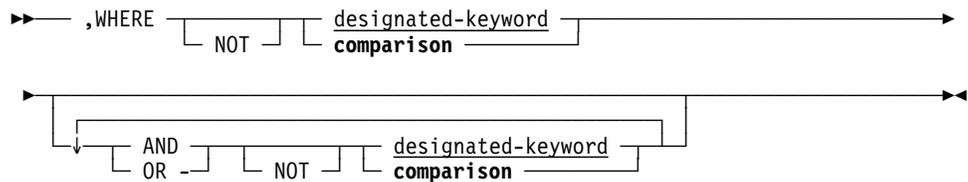
#WTL



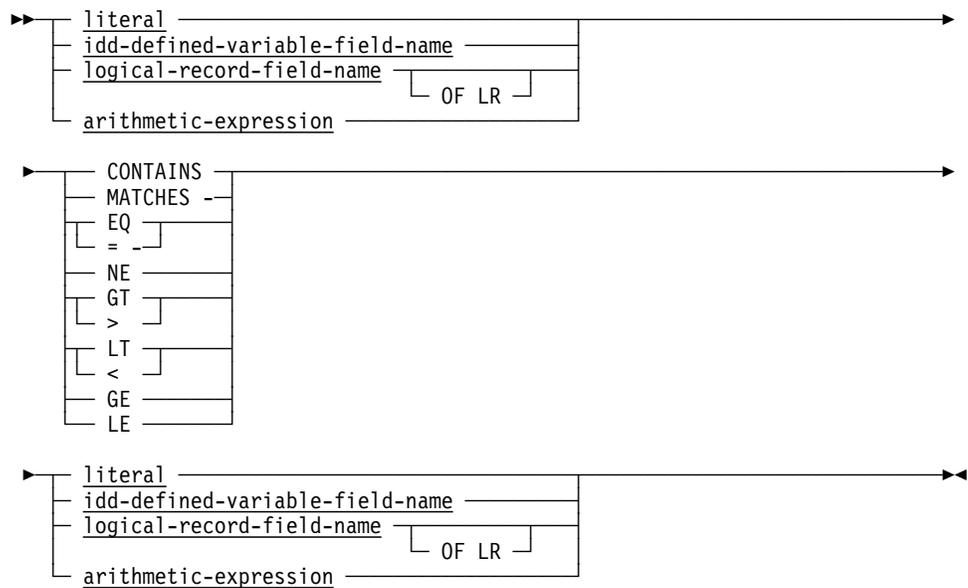
#XCTL



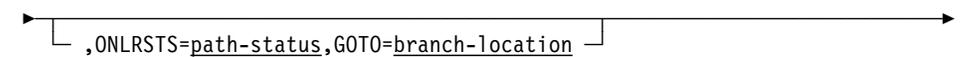
Logical record clauses (WHERE)



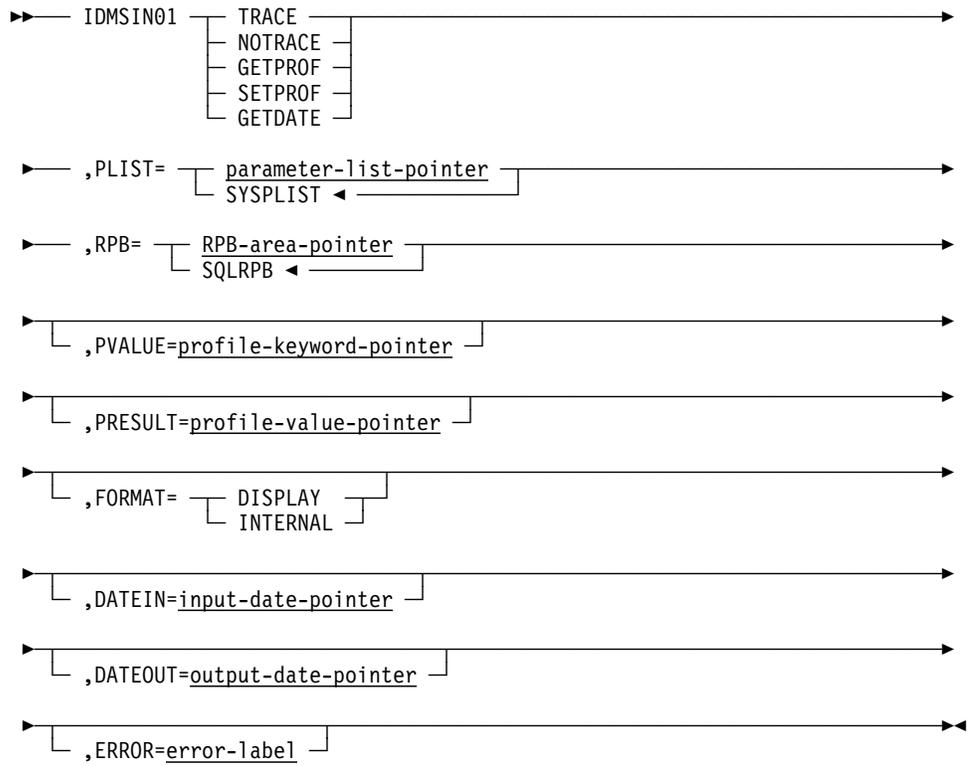
Expansion of comparison



Logical record clauses (ON)



IDMSIN01 macro



Chapter 5. DML statement function table

| Function | DML Statement | Description |
|-----------------------------|----------------------|--|
| Control Statements | BIND RUN-UNIT | Signs on the application program to the DBMS |
| | BIND RECORD | Establishes addressability in variable storage for one or more records included in the program's subschema |
| | BIND PROCEDURE | Establishes communication between the application program and a DBA-defined database procedure |
| | READY | Prepares database areas for processing |
| | FINISH | Releases database areas from program control |
| | IF | Evaluates the presence of records in a set or a record's membership status and specifies action based on the outcome |
| | COMMIT | Writes a checkpoint to the journal file and releases record locks |
| | ROLLBACK | Requests recovery of database, scratch, and queue areas |
| | KEEP | Places locks on record occurrences |
| | Retrieval Statements | FIND/OBTAIN DBKEY |
| FIND/OBTAIN CURRENT | | Accesses a record using previously established currencies |
| FIND/OBTAIN WITHIN SET/AREA | | Accesses a record based on its logical location within a set or its physical location within an area |
| FIND/OBTAIN OWNER | | Accesses the owner record of a set occurrence |
| FIND/OBTAIN CALC/DUPLICATE | | Accesses a record using its CALC-key value |
| FIND/OBTAIN USING SORT KEY | | Accesses a record in a sorted set using its sort-key value |

| Function | DML Statement | Description |
|-------------------------|---|---|
| | GET | Moves all data associated with a previously located record into program variable storage |
| | RETURN | Retrieves the database key and symbolic key of an indexed record entry |
| Modification Statements | STORE | Adds a new record to the database |
| | MODIFY | Changes the contents of an existing record |
| | CONNECT | Links a record to a set |
| | DISCONNECT | Removes a member record from a set |
| | ERASE | Deletes a record from the database |
| Recovery Functions | COMMIT | Establishes a checkpoint in the journal file for database, scratch, and queue record activity |
| | FINISH | Relinquishes control of database, scratch, and queue areas |
| | ROLLBACK | Rolls back database, scratch, and queue areas to the last checkpoint |
| Accept Statements | ACCEPT DBKEY FROM CURRENCY | Saves the db-key of the current record of run unit, record type, set, or area |
| | ACCEPT DBKEY RELATIVE TO CURRENCY | Saves the db-key of the next, prior, or owner record relative to the current record of a set |
| | ACCEPT PAGE INFO LOCATION FOR RECORD NAME | Moves the page information for a given record to a specified location in program variable storage |
| | ACCEPT IDMS STATISTICS | Returns system run-time statistics to the program |
| | ACCEPT BIND RECORD | Returns a record's bind address to the program |

| Function | DML Statement | Description |
|-------------------------|-------------------------|--|
| | ACCEPT PROCEDURE | Returns information from the application program information block associated with a database procedure to the program |
| Logical Record Facility | ERASE | Deletes a logical record |
| | MODIFY | Modifies a logical record |
| | OBTAIN | Accesses a logical record |
| | STORE | Stores a logical record |
| Program Management | TRANSFER CONTROL (LINK) | Passes control to another program with the expectation of receiving it back |
| | TRANSFER CONTROL (XCTL) | Passes control to another program with no expectation of receiving it back |
| | DC RETURN | Returns control to the next higher level calling program |
| | LOAD TABLE | Loads a program or table into the DC/UCF program pool |
| | DELETE TABLE | Signals that a program has finished using a program or a table in the program pool |
| | SET ABEND EXIT (STAE) | Establishes linkage to a program or routine that will receive control in the event of an abend |
| | ABEND | Abnormally terminates the issuing task |
| Storage Management | GET STORAGE | Allocates variable storage from an DC/UCF storage pool |
| | FREE STORAGE | Frees all or part of a block of variable storage |
| Task Management | ATTACH | Attaches a new task within DC/UCF |
| | CHANGE PRIORITY | Changes the dispatching priority of the issuing task |
| | ENQUEUE | Acquires a resource or a list of resources |

| Function | DML Statement | Description |
|----------------------------------|--------------------------|--|
| | DEQUEUE | Releases a resource |
| | WAIT | Relinquishes control to DC/UCF while awaiting completion of an event |
| | POST | Posts an event control block (ECB) |
| Time Management | GET TIME | Obtains the time and date from the system |
| | SET TIMER | Defines a time-delayed event |
| Scratch Management | PUT SCRATCH | Stores a scratch record |
| | GET SCRATCH | Retrieves a scratch record |
| | DELETE SCRATCH | Deletes a scratch record |
| Queue Management | PUT QUEUE | Stores a queue record |
| | GET QUEUE | Retrieves a queue record |
| | DELETE QUEUE | Deletes a queue record |
| Terminal Management (Basic Mode) | READ TERMINAL | Requests a synchronous or asynchronous data transfer from the terminal to program variable storage |
| | WRITE TERMINAL | Requests a synchronous or asynchronous data transfer from program variable storage to the terminal buffer |
| | WRITE THEN READ TERMINAL | Requests a synchronous or asynchronous data transfer from program variable storage to the terminal buffer and, upon an operator signal, back to variable storage |
| | CHECK TERMINAL | Ensures that a previously issued asynchronous I/O operation is complete |
| Terminal Management (Line Mode) | READ LINE FROM TERMINAL | Requests a synchronous data transfer from the terminal to the issuing program |
| | WRITE LINE TO TERMINAL | Requests a synchronous or asynchronous data transfer from the issuing program to the terminal |

| Function | DML Statement | Description |
|------------------------------------|-------------------------------|--|
| | END LINE TERMINAL SESSION | Terminates the current line I/O session |
| | WRITE PRINTER | Requests transmission of data from a task to a printer |
| Terminal Management (Mapping Mode) | BIND MAP | Identifies the location of the map request block (MRB) and initializes MRB fields |
| | MAP IN | Requests a transfer of data from the terminal to program variable storage |
| | MAP OUT | Requests a transfer of data from program variable storage to the terminal |
| | MAP OUTIN | Requests a transfer of data from program variable storage to the terminal and, upon an operator signal, back to variable storage |
| | INQUIRE MAP | Obtains information or tests conditions concerning the previous map operation |
| | MODIFY MAP | Requests modifications of mapping options for a map |
| | STARTPAGE | Begins a map paging session and specifies options for that session |
| | ENDPAGE | Terminates a map paging session |
| Utility Functions | ACCEPT | Retrieves task-related information |
| | SNAP | Requests a memory dump of selected parts of storage |
| | SEND MESSAGE | Sends a message to a user, logical terminal, or list of users or logical terminals |
| | BIND TRANSACTION STATISTICS | Defines the beginning of a transaction for the purpose of collecting transaction statistics |
| | ACCEPT TRANSACTION STATISTICS | Returns the contents of the transaction statistics block (TSB) to program variable storage |

| Function | DML Statement | Description |
|-----------------|-------------------------------|---|
| | END TRANSACTION STATISTICS | Defines the end of a transaction |
| | KEEP LONGTERM | Either modifies a prior KEEP LONGTERM request or enables database longterm locks or database monitoring for records, sets, or areas |
| | WRITE JOURNAL | Writes user-defined records to the journal file |
| | WRITE LOG | Retrieves a message from the data dictionary and sends it to a predefined destination |

Chapter 6. Call formats

6.1 CA-IDMS/DB call formats

CA-IDMS/DB passes the IDBMSCOM array as argument 1.

Arguments marked with asterisks have default values.

| Major Function Code | Database Service (in COBOL DML) | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|---------------------------|--|---|----------------------------------|-----------------------------------|---|----------------------|
| | | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| CONTROL STATEMENTS | | | | | | |
| 14 | BIND RUN-UNIT | 59 | IDMS Communications Block* | <u>subschema-name*</u> | | |
| | BIND RUN-UNIT FOR <u>subschema-name</u> | 59 | IDMS Communications Block* | <u>subschema-name</u> | | |
| | BIND RUN-UNIT NODENAME <u>dds-node-name</u> | 59 | IDMS Communications Block* | <u>subschema-name*</u> | <u>subschema-ctrl*</u> OR <u>subschema-lr-ctrl*</u> | <u>dds-node-name</u> |
| | BIND RUN-UNIT FOR <u>subschema-name</u> NODENAME <u>dds-node-name</u> | 59 | IDMS Communications Block* | <u>subschema-name</u> | <u>subschema-ctrl*</u> OR <u>subschema-lr-ctrl*</u> | <u>dds-node-name</u> |
| | BIND RUN-UNIT FOR <u>subschema-name</u> DBNAME <u>database-name</u> | 59 | IDMS Communications Block* | <u>subschema-name</u> | <u>subschema-ctrl*</u> OR <u>subschema-lr-ctrl*</u> | <u>dds-node-name</u> |
| | BIND RUN-UNIT NODENAME <u>dds-node-name</u> DBNAME <u>database-name</u> | 59 | IDMS Communications Block* | <u>subschema-name*</u> | <u>subschema-ctrl*</u> OR <u>subschema-lr-ctrl*</u> | <u>dds-node-name</u> |
| | BIND RUN-UNIT FOR <u>subschema-name</u> NODENAME <u>dds-node-name</u> DBNAME <u>database-name</u> | 59 | IDMS Communications Block* | <u>subschema-name</u> | <u>subschema-ctrl*</u> OR <u>subschema-lr-ctrl*</u> | <u>dds-node-name</u> |
| | BIND <u>record-name</u> | 48 | <u>record-id</u> | <u>record-location*</u> | | |
| | BIND <u>record-name</u> TO <u>record-location</u> | 48 | <u>record-id</u> | <u>record-location</u> | | |
| | BIND <u>record-location</u> WITH <u>record-name</u> | 48 | <u>record-id</u> | <u>record-location</u> | | |
| | BIND PROCEDURE FOR <u>procedure-name</u> TO <u>procedure-control-location</u> | 73 | <u>procedure-name</u> | <u>procedure-control-location</u> | | |

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|---------------------------|---|---|--------------------|-----|-----|-----|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| CONTROL STATEMENTS | | | | | | |
| 09 | READY | 37 | | | | |
| | READY <u>area-name</u> | 37 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS RETRIEVAL | 37 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS PROTECTED RETRIEVAL | 39 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS EXCLUSIVE RETRIEVAL | 40 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS UPDATE | 36 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS PROTECTED UPDATE | 38 | <u>area-name</u> | | | |
| | READY <u>area-name</u> USAGE-MODE IS EXCLUSIVE UPDATE | 41 | <u>area-name</u> | | | |
| | READY USAGE-MODE IS ... | ** | | | | |
| | **Choose function code from 36-41 as shown above. | | | | | |
| 01 | FINISH | 02 | | | | |
| | FINISH TASK | 113 | | | | |
| 18 | COMMIT | 66 | | | | |
| | COMMIT ALL | 95 | | | | |
| | COMMIT TASK | 114 | | | | |
| | COMMIT TASK ALL | 115 | | | | |
| 19 | ROLLBACK | 67 | | | | |
| | ROLLBACK ALL | 96 | | | | |
| | ROLLBACK TASK | 116 | | | | |
| | ROLLBACK TASK ALL | 117 | | | | |
| 06 | KEEP CURRENT | 87 | | | | |
| | KEEP EXCLUSIVE CURRENT | 90 | | | | |
| | KEEP CURRENT <u>record-name</u> | 89 | <u>record-name</u> | | | |
| | KEEP EXCLUSIVE CURRENT <u>record-name</u> | 90 | <u>record-name</u> | | | |

6.1 CA-IDMS/DB call formats

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|--|--|---|--------------------|-----------------|-----|-----|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| CONTROL STATEMENTS | | | | | | |
| 16 | KEEP CURRENT WITHIN <u>set-name</u> | 91 | <u>set-name</u> | | | |
| | KEEP EXCLUSIVE CURRENT WITHIN <u>set-name</u> | 92 | <u>set-name</u> | | | |
| | KEEP CURRENT WITHIN <u>area-name</u> | 93 | <u>area-name</u> | | | |
| | KEEP EXCLUSIVE CURRENT WITHIN <u>area-name</u> | 92 | <u>area-name</u> | | | |
| | IF <u>set-name</u> IS EMPTY... | 64 | <u>set-name</u> | | | |
| | IF <u>set-name</u> IS NOT EMPTY... | 65 | <u>set-name</u> | | | |
| <p>(Upon return to user run-unit, the error-status indicator = '0000' if set is empty; '1601' if not empty.)</p> | | | | | | |
| | IF <u>set-name</u> MEMBER... | 60 | <u>set-name</u> | | | |
| | IF NOT <u>set-name</u> MEMBER... | 62 | <u>set-name</u> | | | |
| <p>(Upon return to user run-unit, the error-status indicator = '0000' if the record (current of run unit) is linked into the specified set; '1601' if it is not a member.)</p> | | | | | | |
| MODIFICATION STATEMENTS | | | | | | |
| 12 | STORE <u>record-name</u> | 42 | <u>record-name</u> | | | |
| 07 | CONNECT <u>record-name</u> TO <u>set-name</u> | 44 | <u>record-name</u> | <u>set-name</u> | | |
| 08 | MODIFY <u>record-name</u> | 35 | <u>record-name</u> | | | |
| 11 | DISCONNECT <u>record-name</u> FROM <u>set-name</u> | 46 | <u>record-name</u> | <u>set-name</u> | | |
| 02 | ERASE <u>record-name</u> | 52 | <u>record-name</u> | | | |
| | ERASE <u>record-name</u> PERMANENT MEMBERS | 03 | <u>record-name</u> | | | |
| | ERASE <u>record-name</u> SELECTIVE MEMBERS | 53 | <u>record-name</u> | | | |
| | ERASE <u>record-name</u> ALL MEMBERS | 4 | <u>record-name</u> | | | |

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|-----------------------------|--|---|--------------------|------------------|---------------|-----|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| RETRIEVAL STATEMENTS | | | | | | |
| 03 | FIND DB-KEY <u>db-key</u> | 75 | <u>db-key</u> | | | |
| | FIND DB-KEY IS <u>db-key</u> PAGE-INFO <u>page-info</u> | 29 | <u>db-key</u> | <u>page-info</u> | | |
| | FIND <u>record-name</u> DB-KEY IS <u>db-key</u> | 06 | <u>record-name</u> | <u>db-key</u> | | |
| | FIND CURRENT | 30 | | | | |
| | FIND CURRENT <u>record-name</u> | 07 | <u>record-name</u> | | | |
| | FIND CURRENT WITHIN <u>set-name</u> | 08 | <u>set-name</u> | | | |
| | FIND CURRENT WITHIN <u>area-name</u> | 09 | <u>area-name</u> | | | |
| | FIND NEXT WITHIN <u>set-name</u> | 14 | <u>set-name</u> | | | |
| | FIND NEXT <u>record-name</u> WITHIN <u>set-name</u> | 10 | <u>record-name</u> | <u>set-name</u> | | |
| | FIND PRIOR WITHIN <u>set-name</u> | 16 | <u>set-name</u> | | | |
| | FIND PRIOR <u>record-name</u> WITHIN <u>set-name</u> | 12 | <u>record-name</u> | <u>set-name</u> | | |
| | FIND FIRST WITHIN <u>set-name</u> | 20 | <u>set-name</u> | | | |
| | FIND FIRST <u>record-name</u> WITHIN <u>set-name</u> | 18 | <u>record-name</u> | <u>set-name</u> | | |
| | FIND LAST WITHIN <u>set-name</u> | 24 | <u>set-name</u> | | | |
| | FIND LAST <u>record-name</u> WITHIN <u>set-name</u> | 22 | <u>record-name</u> | <u>set-name</u> | | |
| | FIND <u>number</u> WITHIN <u>set-name</u> | 78 | <u>set-name</u> | <u>number</u> | | |
| | FIND <u>number</u> <u>record-name</u> WITHIN <u>set-name</u> | 76 | <u>record-name</u> | <u>set-name</u> | <u>number</u> | |
| | FIND NEXT WITHIN <u>area-name</u> | 15 | <u>area-name</u> | | | |
| | FIND NEXT <u>record-name</u> WITHIN <u>area-name</u> | 11 | <u>record-name</u> | <u>area-name</u> | | |
| | FIND PRIOR WITHIN <u>area-name</u> | 17 | <u>area-name</u> | | | |
| | FIND PRIOR <u>record-name</u> WITHIN <u>area-name</u> | 13 | <u>record-name</u> | <u>area-name</u> | | |

6.1 CA-IDMS/DB call formats

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | | |
|-----------------------------|---|---|--------------------|------------------|-----------------|-----|--|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) | |
| RETRIEVAL STATEMENTS | | | | | | | |
| 03 | FIND FIRST WITHIN <u>area-name</u> | 21 | <u>area-name</u> | | | | |
| | FIND FIRST <u>record-name</u> WITHIN <u>area-name</u> | 19 | <u>record-name</u> | <u>area-name</u> | | | |
| | FIND LAST WITHIN <u>area-name</u> | 25 | <u>area-name</u> | | | | |
| | FIND LAST <u>record-name</u> WITHIN <u>area-name</u> | 23 | <u>record-name</u> | <u>area-name</u> | | | |
| | FIND <u>number</u> WITHIN <u>area-name</u> | 79 | <u>area-name</u> | <u>number</u> | | | |
| | FIND <u>number</u> <u>record-name</u> WITHIN <u>area-name</u> | 77 | <u>record-name</u> | <u>area-name</u> | <u>number</u> | | |
| | FIND OWNER WITHIN <u>set-name</u> | 31 | <u>set-name</u> | | | | |
| | FIND CALC (ANY) <u>record-name</u> | 32 | <u>record-name</u> | | | | |
| | FIND DUPLICATE <u>record-name</u> | 50 | <u>record-name</u> | | | | |
| | FIND <u>record-name</u> WITHIN <u>set-name</u> USING <u>sort-key</u> | 33 | <u>record-name</u> | <u>set-name</u> | <u>sort-key</u> | | |
| | FIND <u>record-name</u> WITHIN <u>set-name</u> CURRENT USING <u>sort-key</u> | 51 | <u>record-name</u> | <u>set-name</u> | <u>sort-key</u> | | |
| | <p>Any of the above FIND record selection expressions. Call generated consists of arguments described above for the FIND in question plus an additional argument of IDBMSCOM(43) function. For example:</p> | | | | | | |
| | OBTAIN CALC <u>record-name</u> | 32 | <u>record-name</u> | | IDBMSCOM(43) | | |
| | OBTAIN PRIOR <u>record-name</u> WITHIN <u>set-name</u> | 12 | <u>record-name</u> | | | | |

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|-----------------------------|--|---|-----------------------|---------------|---------------------|-----|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| RETRIEVAL STATEMENTS | | | | | | |
| 03 | KEEP KEEP EXCLUSIVE | | | | | |
| | Any of the above FIND/OBTAIN record selection expressions. Call generated consists of arguments described above for the FIND/OBTAIN in question plus one of the following additional IDBMSCOM functions: KEEP IDBMSCOM(87) KEEP EXCLUSIVE IDBMSCOM(88) For example: OBTAIN KEEP CALC <u>record-name</u> FIND KEEP EXCLUSIVE CURRENT | | | | | |
| 05 | GET | 43 | | | | |
| | GET <u>record-name</u> | 34 | <u>record-name</u> | IDBMSCOM(43) | IDBMSCOM(87) | |
| 17 | RETURN <u>db-key</u> | 81 | <u>index-set-name</u> | <u>db-key</u> | <u>symbolic-key</u> | |
| | FROM <u>index-set-name</u> CURRENCY KEY INTO <u>symbolic-key</u> | | | | | |
| | RETURN <u>db-key</u> | 82 | <u>index-set-name</u> | <u>db-key</u> | <u>symbolic-key</u> | |
| | FROM <u>index-set-name</u> FIRST KEY INTO <u>symbolic-key</u> | | | | | |
| | RETURN <u>db-key</u> | 83 | <u>index-set-name</u> | <u>db-key</u> | <u>symbolic-key</u> | |
| | FROM <u>index-set-name</u> LAST KEY INTO <u>symbolic-key</u> | | | | | |
| | RETURN <u>db-key</u> | 84 | <u>index-set-name</u> | <u>db-key</u> | <u>symbolic-key</u> | |
| | FROM <u>index-set-name</u> NEXT KEY INTO <u>symbolic-key</u> | | | | | |
| | RETURN <u>db-key</u> | 85 | <u>index-set-name</u> | <u>db-key</u> | <u>symbolic-key</u> | |
| | FROM <u>index-set-name</u> PRIOR KEY INTO <u>symbolic-key</u> | | | | | |

6.1 CA-IDMS/DB call formats

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|-----------------------------|---|---|---------------------------|--|------------------------|---------------------|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| RETRIEVAL STATEMENTS | | | | | | |
| 17 | RETURN <u>db-key</u> FROM <u>index-set-name</u> USING <u>index-key-value</u> KEY INTO <u>symbolic-key</u> | 86 | <u>index-set-name</u> | <u>db-key</u> | <u>index-key-value</u> | <u>symbolic-key</u> |
| ACCEPT STATEMENTS | | | | | | |
| 15 | ACCEPT <u>db-key</u> FROM CURRENCY | 54 | <u>db-key</u> | | | |
| | ACCEPT <u>db-key</u> FROM <u>record-name</u> CURRENCY | 55 | <u>record-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>db-key</u> FROM <u>set-name</u> CURRENCY | 57 | <u>set-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>db-key</u> FROM <u>area-name</u> CURRENCY | 56 | <u>area-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>db-key</u> FROM <u>set-name</u> NEXT CURRENCY | 68 | <u>set-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>db-key</u> FROM <u>set-name</u> PRIOR CURRENCY | 69 | <u>set-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>db-key</u> FROM <u>set-name</u> OWNER CURRENCY | 70 | <u>set-name</u> | <u>db-key</u> | | |
| | ACCEPT <u>page-info- location</u> FOR <u>record-name</u> | 28 | <u>page-info-location</u> | <u>record-name</u> | | |
| | ACCEPT <u>db-statistics</u> FROM IDMS-STATISTICS | 71 | <u>db-statistics</u> | | | |
| | ACCEPT <u>bind-address</u> FROM <u>record-name</u> BIND | 72 | <u>record-name</u> | <u>bind-address</u> | | |
| | ACCEPT <u>procedure- control- location</u> FROM <u>procedure-name</u> PROCEDURE | 74 | <u>procedure-name</u> | <u>procedure-control- location</u> | | |

| Major Function Code | Database Service (in COBOL DML) | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|---------------------------|--|---|---------------------------|--------------------------------------|-----|-----|
| | | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| LRF DML STATEMENTS | | | | | | |
| 20 | OBTAIN FIRST <u>logical- record-name</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location*</u> | | |
| | OBTAIN FIRST <u>logical- record-name</u> INTO <u>alt-logical- record- location</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location</u> | | |
| | OBTAIN NEXT <u>logical- record-name</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location*</u> | | |
| | OBTAIN NEXT <u>logical- record-name</u> INTO <u>alt-logical- record- location</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location</u> | | |
| | MODIFY <u>logical- record-name</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location*</u> | | |
| | MODIFY <u>logical- record-name</u> FROM <u>alt-logical- record- location</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location</u> | | |
| | STORE <u>logical- record-name</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location*</u> | | |
| | STORE <u>logical- record-name</u> FROM <u>alt-logical- record- location</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location</u> | | |
| | ERASE <u>logical- record-name</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location*</u> | | |
| | ERASE <u>logical- record-name</u> FROM <u>alt-logical- record- location</u> | 99 | <u>subschema-lr-ctrl*</u> | <u>logical-record- location</u> | | |
| | To differentiate between the LRF DML statements the DML compiler places the name of the verb issued into the LRC Block (<u>subschema-lr-ctrl</u>). | | | | | |

6.2 CA-IDMS/DC call formats

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|--------------------------------------|---------------------------------|---|---------------------------------------|------------------------------------|----------------------|----------------------|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| PROGRAM MANAGEMENT STATEMENTS | | | | | | |
| 30 | TRANSFER CONTROL | 23 | DCFLG1 | DCSTR2 | <u>parameter</u> | |
| 30 | DC RETURN | 19 | | | | |
| 34 | LOAD TABLE | 15 | <u>01-level-program-location</u> | <u>end-program-location</u> | | |
| 34 | DELETE TABLE | 5 | <u>01-level-program-location</u> | | | |
| 33 | SET ABEND EXIT (STAE) | 20 | | | | |
| 33 | ABEND | 1 | | | | |
| STORAGE MANAGEMENT STATEMENTS | | | | | | |
| 32 | GET STORAGE | 13 | <u>01-level-storage-data-location</u> | <u>end-storage-data-location</u> | | |
| 32 | FREE STORAGE | 10 | <u>01-level-storage-data-location</u> | <u>start-free-storage-location</u> | | |
| TASK MANAGEMENT STATEMENTS | | | | | | |
| 37 | ATTACH | 3 | | | | |
| 37 | CHANGE PRIORITY | 4 | | | | |
| 39 | ENQUEUE | 9 | DCFLG1 | DCBMSCOM (mode) | DCBMSCOM (length) | <u>resource-id..</u> |
| 39 | DEQUEUE | 8 | DCFLG1 | DCBMSCOM (length) | <u>resource-id..</u> | |
| 31 | WAIT | 24 | <u>ecb</u> | | | |
| 31 | POST | 16 | <u>ecb</u> | | | |
| TIME MANAGEMENT STATEMENTS | | | | | | |
| 35 | GET TIME | 14 | <u>return-time</u> | <u>return-data</u> | | |
| 35 | SET TIMER | 21 | <u>task-data-location</u> | <u>end-task-data-location</u> | | |
| 35 | SET TIMER (<u>post</u>) | 21 | <u>post-ecb</u> | | | |
| SCRATCH MANAGEMENT STATEMENTS | | | | | | |
| 43 | PUT SCRATCH | 18 | <u>scratch-data-location</u> | <u>end-scratch-data-location</u> | | |
| 43 | GET SCRATCH | 12 | <u>return-scratch-data-location</u> | <u>end-scratch-data-location</u> | | |
| 43 | DELETE SCRATCH | 6 | | | | |
| QUEUE MANAGEMENT STATEMENTS | | | | | | |
| 44 | PUT QUEUE | 17 | <u>queue-data-location</u> | <u>end-queue-data-location</u> | | |
| 44 | GET QUEUE | 11 | <u>return-queue-data-location</u> | <u>end-queue-data-location</u> | | |
| 44 | DELETE QUEUE | 6 | | | | |

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|---------------------------------------|---------------------------------|---|----------------------------------|-------------------------------------|---|--------------------------------|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| TERMINAL MANAGEMENT STATEMENTS | | | | | | |
| 45 | READ TERMINAL | 30 | <u>input-data- location</u> | <u>end-input-data location</u> | | |
| 45 | WRITE TERMINAL | 30 | <u>output-data- location</u> | <u>end-output-data location</u> | | |
| 45 | WRITE THEN READ TERMINAL | 30 | <u>output-data- location</u> | <u>end-output-data location</u> | <u>input-data location</u> | <u>end-input- location</u> |
| 45 | CHECK TERMINAL | 31 | <u>input-data- location</u> | <u>end-input-data location</u> | | |
| 47 | READ LINE FROM TERMINAL | 32 | <u>input-data- location</u> | <u>end-input-data location</u> | | |
| 47 | WRITE LINE TO TERMINAL | 32 | <u>output-data- location</u> | <u>end-output-data location</u> | | |
| 47 | END LINE TERMINAL SESSION | 32 | | | | |
| 48 | WRITE PRINTER | 37 | <u>message-location</u> | <u>end-message- location</u> | | |
| 46 | MAP IN (IO) | 34 | MRB- <u>mapname</u> | | <u>end-mapped-data location</u> | |
| 46 | MAP IN (NOIO) | 34 | MRB- <u>mapname</u> | <u>mapped-data- location</u> | <u>end-mapped-data location</u> | |
| 46 | MAP IN (<u>paging</u>) (a) | 34 | MRB- <u>mapname</u> | <u>data-field-name</u> | <u>sequence-field- name</u> | <u>page-number</u> |
| 46 | MAP IN (<u>paging</u>) (b) | 34 | MRB- <u>mapname</u> | <u>key</u> | <u>page-number</u> | |
| 46 | MAP OUT (IO) | 34 | MRB- <u>mapname</u> | <u>message-text</u> | <u>end-message-data location</u> OR DCBMSCOM (length) | |
| 46 | MAP OUT (NOIO) | 34 | MRB- <u>mapname</u> | <u>mapped-data- location</u> | <u>end-mapped-data location</u> | |
| 46 | MAP OUT (<u>paging</u>) | 34 | MRB- <u>mapname</u> | <u>message-text</u> | <u>end-message-data location</u> OR DCBMSCOM (length) | <u>key</u> |
| 46 | MAP OUTIN | 34 | MRB- <u>mapname</u> | <u>message-text</u> | <u>end-message-data location</u> OR DCBMSCOM (length) | |
| 46 | MODIFY MAP | 93 | MRB- <u>mapname</u> | MRE | MRB-FLDLST | |
| 46 | INQUIRE MAP (a) | 92 | MRB- <u>mapname</u> | MRE | | |
| 46 | INQUIRE MAP (b) | 92 | MRB- <u>mapname</u> | | | |
| 46 | INQUIRE MAP (c) | 92 | MRB- <u>mapname</u> | | | |
| 46 | INQUIRE MAP (d) | 92 | MRB- <u>mapname</u> | MRE | | |
| 46 | STARTPAGE | 40 | MRB- <u>mapname</u> | MRB-FLDLST | | |
| 46 | ENDPAGE | 41 | | | | |

6.2 CA-IDMS/DC call formats

| | | CALLING ARGUMENTS (argument 0 contains SUBSCHEMA-CTRL) | | | | |
|----------------------------|---------------------------------|---|---|---|---|---|
| Major Function Code | Database Service (in COBOL DML) | (1) IDBMSCOM (nn) | (2) | (3) | (4) | (5) |
| UTILITY STATEMENTS | | | | | | |
| 48 40 | ACCEPT SNAP | 2 22 | <u>return-location</u> DCSTR1 | DCSTR1 (6) <u>begin-dump- location</u> <u>message-location</u> | DCSTR1 (7) <u>end-dump- location</u> <u>end-message location</u> | <u>title</u> (0) DCBMSCOM(1) |
| 49 | SEND MESSAGE | 38 | <u>user-id</u> | | | |
| 38 | BIND TRANSACTION STATISTICS | 28 | | | | |
| 38 | ACCEPT TRANSACTION STATISTICS | 28 | <u>return-statistics data-location</u> | | | |
| 38 | END TRANSACTION STATISTICS | 28 | <u>return-statistics data-location</u> | | | |
| 51 | KEEP LONGTERM | 29 | <u>record-name</u> <u>set-name</u> <u>area-name</u> | | | |
| 36 | WRITE LOG | 25 | <u>text-return location</u> | <u>end-text-return location</u> | <u>reply-location</u> (6) <u>parameter- location</u> | <u>end-reply- location</u> (7) <u>end-parameter- location</u> |
| RECOVERY STATEMENTS | | | | | | |
| 50 | COMMIT | 66 | | | | |
| 50 | COMMIT TASK | 27 | | | | |
| 50 | FINISH | 2 | | | | |
| 50 | FINISH TASK | 27 | | | | |
| 50 | ROLLBACK | 67 | | | | |
| 50 | ROLLBACK TASK | 27 | | | | |
| 50 | WRITE JOURNAL | 26 | <u>record-location</u> | <u>end-record- location</u> | | |
| DC-BATCH STATEMENTS | | | | | | |
| 14 | BIND-TASK | 28 | DCSTR2 | | | |

Chapter 7. IDMS-DB/DC Communications block

7.1 16-byte IDMS communications block

| |
|---|
| 16-CHARACTER IDMS COMMUNICATIONS BLOCK |
|---|

| | Field | Data Type | Length (bytes) | Suggested Initial Value | |
|--------------|--------------------------|-----------------|----------------|-------------------------|--------------|
| * | 1 8 | PROGRAM-NAME | Alphanumeric | 8 | Program Name |
| | 9 12 | ERROR-STATUS | Alphanumeric | 4 | '1400' |
| | 13 16 | DBKEY | Binary | 4 (Fullword) | 0000 |
| | 17 32 | RECORD-NAME | Alphanumeric | 16 | Spaces |
| | 33 48 | AREA-NAME | Alphanumeric | 16 | Spaces |
| | 49 64 | ERROR-SET | Alphanumeric | 16 | Spaces |
| | 65 80 | ERROR-RECORD | Alphanumeric | 16 | Spaces |
| | 81 96 | ERROR-AREA | Alphanumeric | 16 | Spaces |
| ** | 97 100 | PAGE-INFO | Binary | 4 (Fullword) | 0000 |
| | 97 ... 196 | IDSSCOM-AREA | Alphanumeric | 100 | Low Values |
| | 197 200 | DIRECT-DBKEY | Binary | 4 (Fullword) | 0000 |
| NON- CICS | 201 207 | DATABASE-STATUS | Alphanumeric | 7 | Spaces |
| | 208 | FILLER | ... | 1 | ... |
| | 209 212 | RECORD-OCCUR | Binary | 4 (Fullword) | 0000 |
| | 213 216 | DML-SEQUENCE | Binary | 4 (Fullword) | 0000 |
| CICS | 201 216 | FILLER | ... | 16 | Spaces |
| | 217 223 | DATABASE-STATUS | Alphanumeric | 7 | Spaces |
| | 224 | FILLER | ... | 1 | ... |
| | 225 228 | RECORD-OCCUR | Binary | 4 (Fullword) | 0000 |
| | 229 232 | DML-SEQUENCE | Binary | 4 (Fullword) | 0000 |

* word aligned

** PAGE-INFO-GROUP overlays bytes 97 and 98 and PAGE-INFO-DBK-FORMAT overlays bytes 99 and 100. Both of these fields are binary datatype, each with a length of two bytes. Suggested initial values for both are 00. Together these two fields represent PAGE-INFO.

7.2 IDMS communications block fields

| | SUCCESSFUL | | | | | | | | | UNSUCCESSFUL | | | | | | | | | | |
|---------------------------|--|--|-----------------------|---|---|---|---|--|---|--|--|--|-----------------------|---|---|---|---|--|---|--|
| | P R O G R A M - N A M E | E R R O R - S T A T U S | D B K E Y | R E C O R D - N A M E | A R E A - N A M E | E R R O R - S E T | E R R O R R - R E C O R D | E R R O R - A R E A | P A G E - I N F O | D I R E C T - D B K E Y | P R O G R A M - N A M E | E R R O R - S T A T U S | D B K E Y | R E C O R D - N A M E | A R E A - N A M E | E R R O R - S E T | E R R O R R - R E C O R D | E R R O R - A R E A | P A G E - I N F O | D I R E C T - D B K E Y |
| Control statements | | | | | | | | | | | | | | | | | | | | |
| BIND RUN-UNIT | | 0 | | | | | | | | | 14nn | | | | | | | | | |
| BIND RECORD | | 0 | | | | | | | | | 14nn | | | | Y | Y | Y | | | |
| BIND PROCEDURE | | 0 | | | | | | | | | 14nn | | | | Y | Y | Y | | | |
| READY | | 0 | | | | | | | | | 09nn | | | | C | C | C | | | |
| FINISH | | 0 | N | C | | C | C | C | | | 01nn | | | | C | C | C | | | |
| COMMIT (ALL) | | 0 | N | C | | C | C | C | | | 18nn | | | | C | C | C | | | |
| ROLLBAK (CONTINUE) | | 0 | N | C | | C | C | C | | | 19nn | | | | C | C | C | | | |
| KEEP (EXCLUSIVE) | | 0 | Y | Y | Y | C | C | C | Y | | 06nn | | | | Y | Y | Y | | | |
| IF SET | | * | Y | Y | Y | C | C | C | Y | | 16nn | | | | Y | Y | Y | | | |
| IF NOT SET | | * | Y | Y | Y | C | C | C | Y | | 16nn | | | | Y | Y | Y | | | |
| Retrieval statements | | | | | | | | | | | | | | | | | | | | |
| FIND/OBTAIN RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 03nn | | | | Y | Y | Y | | | |
| GET RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 05nn | | | | Y | Y | Y | | | |
| RETURN RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 17nn | | | | Y | Y | Y | | | |
| Modification statements | | | | | | | | | | | | | | | | | | | | |
| STORE RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 12nn | | | | Y | Y | Y | | | |
| CONNECT RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 07nn | | | | Y | Y | Y | | | |
| MODIFY RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 08nn | | | | Y | Y | Y | | | |
| DISCONNECT RECORD | | 0 | Y | Y | Y | C | C | C | Y | | 11nn | | | | Y | Y | Y | | | |
| ERASE RECORD | | 0 | N | Y | Y | C | C | C | | | 02nn | | | | Y | Y | Y | | | |
| Accept statements | | | | | | | | | | | | | | | | | | | | |
| ACCEPT DBKEY OF CURRENCY | | 0 | | | | C | C | C | | | 15nn | | | | Y | Y | Y | | | |
| ACCEPT DBKEY OF N/P/O | | 0 | | | | C | C | C | | | 15nn | | | | Y | Y | Y | | | |
| ACCEPT IDMS STATISTICS | | 0 | | | | C | C | C | | | 15nn | | | | Y | Y | Y | | | |
| ACCEPT BIND RECORD | | 0 | | | | C | C | C | | | 15nn | | | | Y | Y | Y | | | |
| ACCEPT PROCEDURE | | 0 | | | | C | C | C | | | 82nn | | | | Y | Y | Y | | | |
| ACCEPT page-info-location | | 0 | | | | C | C | C | | | 15nn | | | | Y | Y | Y | | | |

7.2 IDMS communications block fields

| Accept statements | | | | | | | | | | | | | | | | |
|-----------------------------|--|---|--|--|--|---|---|---|--|------|--|--|--|---|---|---|
| ACCEPT FROM CURRENCY | | 0 | | | | C | C | C | | 15nn | | | | Y | Y | Y |
| ACCEPT FROM N/P/O CURRENCY | | 0 | | | | C | C | C | | 15nn | | | | Y | Y | Y |
| ACCEPT FROM IDMS-STATISTICS | | 0 | | | | C | C | C | | 15nn | | | | Y | Y | Y |
| ACCEPT FROM BIND | | 0 | | | | C | C | C | | 15nn | | | | Y | Y | Y |
| ACCEPT FROM PROCEDURE | | 0 | | | | C | C | C | | 15nn | | | | Y | Y | Y |

| |
|---|
| <p>* If true, field is set to zone decimal zeroes (0000) If false, field is set to 1601</p> <p>0 Field is set to zone decimal zeroes</p> <p>Y Field is updated</p> <p>C Field is cleared to spaces</p> <p>N Field is set to null db-key value (-1)</p> <p><u>nn</u> Specific minor error code</p> |
|---|

7.3 LRC block

| LRC BLOCK | | Field | Data Type | Length (bytes) | Suggested Initial Value |
|-----------|-----------------------|--------------|--------------|----------------|-------------------------|
| 1 | 2 | LRC-LRPXELNG | Binary | 2 (Halfword) | 00 |
| 3 | 4 | LRC-MAXVXP | Binary | 2 (Halfword) | 00 |
| 5 | 8 | LRIDENT | Alphanumeric | 4 | 'LRC' |
| 9 | 16 | LRVERB | Alphanumeric | 8 | Spaces |
| 17 | 32 | LRNAME | Alphanumeric | 16 | Spaces |
| 33 | 48 | LR-STATUS | Alphanumeric | 16 | Spaces |
| 49 | 64 | FILLER | ... | 16 | ... |
| 65 | ... (variable-length) | PXE | Mixed | ... | ... |

* word aligned

7.4 IDMS/DC communications block

| | | IDMS-DC COMMUNICATIONS BLOCK | | | |
|----|-------------|------------------------------|---------------------|-------------------------|--|
| | Field | Data Type | Length (bytes) | Suggested Initial Value | |
| * | 1 8 | PROGRAM-NAME | Alphanumeric 8 | Program Name | |
| | 9 12 | ERROR-STATUS | Alphanumeric 4 | '1400' | |
| | 13 16 | DBKEY | Binary 4 (Fullword) | 0000 | |
| | 17 32 | RECORD-NAME | Alphanumeric 16 | Spaces | |
| | 33 48 | AREA-NAME | Alphanumeric 16 | Spaces | |
| | 49 64 | ERROR-SET | Alphanumeric 16 | Spaces | |
| | 65 80 | ERROR-RECORD | Alphanumeric 16 | Spaces | |
| | 81 96 | ERROR-AREA | Alphanumeric 16 | Spaces | |
| ** | 97 100 | PAGE-INFO | Binary 4 (Fullword) | 0000 | |
| | 97 ... 196 | IDMSSCOM-AREA | Alphanumeric 100 | Low Values | |
| | 197 200 | DIRECT-DBKEY | Binary 4 (Fullword) | 0000 | |
| | 201 ... 300 | DCBMSCOM-AREA | Alphanumeric 100 | Low Values | |
| | 301 304 | SSC-ERRSTAT-SAVE | Alphanumeric 4 | Spaces | |
| | 305 308 | SSC-DMLSEQ-SAVE | Binary 4 (Fullword) | 0000 | |
| | 309 312 | DML-SEQUENCE | Binary 4 (Fullword) | 0000 | |
| | 313 316 | RECORD-OCCUR | Binary 4 (Fullword) | 0000 | |
| | 317 320 | SUBSCHEMA-CTRL-END | Alphanumeric 4 | Spaces | |

* word aligned

** PAGE-INFO-GROUP overlays bytes 97 and 98 and PAGE-INFO-DBK-FORMAT overlays bytes 99 and 100. Both of these fields are binary datatype each having a length of two bytes. Suggested initial values for both are 00. Together these two fields represent PAGE-INFO.

Chapter 8. Run-time error-status codes

8.1 Major DB status codes

| Code | Database Function |
|-------------|--------------------------|
| 00 | Any DML statement |
| 01 | FINISH |
| 02 | ERASE |
| 03 | FIND/OBTAIN |
| 05 | GET |
| 06 | KEEP |
| 07 | CONNECT |
| 08 | MODIFY |
| 09 | READY |
| 11 | DISCONNECT |
| 12 | STORE |
| 14 | BIND |
| 15 | ACCEPT |
| 16 | IF |
| 17 | RETURN |
| 18 | COMMIT |
| 19 | ROLLBACK |
| 20 | LRF requests |

8.2 Minor DB status codes

| Code | Database Function Status |
|------|---|
| 00 | Combined with a major code of 00, this code indicates successful completion of the DML operation. Combined with a non-zero major code, this code indicates that the DML operation was not completed successfully due to central version causes, such as time-outs and program checks. |
| 01 | An area has not been readied. When this code is combined with a major code of 16, an IF operation has resulted in a valid false condition. |
| 02 | Either the db-key used with a FIND/OBTAIN DB-KEY statement or the direct db-key suggested for a STORE is not within the page range for the specified record name. |
| 04 | The occurrence count of a variably occurring element has been specified as either less than zero or greater than the maximum number of occurrences defined in the control element. |
| 05 | The specified DML function would have violated a duplicates-not-allowed option for a CALC, sorted, or index set. |
| 06 | No currency has been established for the named record, set, or area. |
| 07 | The end of a set, area, or index has been reached or the set is empty. |
| 08 | The specified record, set, procedure, or LR verb is not in the subschema or the specified record is not a member of the set. |
| 09 | The area has been readied with an incorrect usage mode. |
| 10 | An existing access restriction or subschema usage prohibits execution of the specified DML function. For LRF users, the subschema in use allows access to database records only. Combined with a major code of 00, this code means the program has attempted to access a database record, but the subschema in use allows access to logical records only. |
| 11 | The record cannot be stored in the specified area due to insufficient space. |
| 12 | There is no db-key for the record to be stored. This is a system internal error and should be reported. |
| 13 | A current record of run unit either has not been established or has been nullified by a previous ERASE statement. |
| 14 | The CONNECT statement cannot be executed because the requested record has been defined as a mandatory automatic member of the set. |
| 15 | The DISCONNECT statement cannot be executed because the requested record has been defined as a mandatory member of the set. |
| 16 | The record cannot be connected to a set of which it is already a member. |

| Code | Database Function Status |
|-------------|--|
| 18 | The record has not been bound. |
| 20 | The current record is not the same type as the specified record name. |
| 21 | Not all areas being used have been readied in the correct usage mode. |
| 22 | The record name specified is not currently a member of the set name specified. |
| 23 | The area name specified is either not in the subschema or not an extent area; or the record name specified has not been defined within the area name specified. |
| 25 | No currency has been established for the named set. |
| 26 | No duplicates exist for the named record or the record occurrences cannot be found. |
| 28 | The run unit has attempted to ready an area that has been readied previously. |
| 29 | The run unit has attempted to place a lock on a record that is locked already by another run unit. A deadlock results. Unless the run unit issued either a FIND/OBTAIN KEEP EXCLUSIVE or a KEEP EXCLUSIVE, the run unit is aborted. |
| 30 | An attempt has been made to erase the owner record of a nonempty set. |
| 31 | The retrieval statement format conflicts with the record's location mode. |
| 32 | An attempt to retrieve a CALC/DUPLICATE record was unsuccessful; the value of the CALC field in variable storage is not equal to the value of the CALC control element in the current record of run unit. |
| 33 | At least one set in which the record participates has not been included in the subschema. |
| 40 | The WHERE clause in an OBTAIN NEXT logical-record request is inconsistent with a previous OBTAIN FIRST or OBTAIN NEXT command for the same record. Previously specified criteria, such as reference to a key field, have been changed. A path status of LR-ERROR is returned to the LRC block. |
| 41 | The subschema contains no path that matches the WHERE clause in a logical-record request. A path status of LR-ERROR is returned to the LRC block. |
| 42 | An ON clause included in the path by the DBA specified return of the LR-ERROR path status to the LRC block; an error has occurred while processing the LRF request. |

| Code | Database Function Status |
|-------------|---|
| 43 | A program check has been recognized during evaluation of a WHERE clause; the program check indicates that either a WHERE clause has specified comparison of a packed decimal field to an unpacked nonnumeric data field, or data in variable storage or a database record does not conform to its description. A path status of LR-ERROR is returned to the LRC block unless the DBA has included an ON clause to override this action in the path. |
| 44 | The WHERE clause in a logical-record request does not supply a key element (sort key, CALC key, or db-key) expected by the path. A path status of LR-ERROR is returned to the LRC block. |
| 45 | During evaluation of a WHERE clause, a program check has been recognized because a subscript value is neither greater than 0 nor less than its maximum allowed value plus 1. A path status of LR-ERROR is returned to the LRC block unless the DBA has included an ON clause to override this action in the path. |
| 46 | A program check has revealed an arithmetic exception (for example: overflow, underflow, significance, divide) during evaluation of a WHERE clause. A path status of LR-ERROR is returned to the LRC block unless the DBA has included an ON clause to override this action in the path. |
| 53 | The subschema definition of an indexed set does not match the indexed set's physical structure in the database. |
| 54 | Either the prefix length of an SR51 record is less than zero or the data length is less than or equal to zero. |
| 55 | An invalid length has been defined for a variable-length record. |
| 56 | An insufficient amount of memory to accommodate the CA-IDMS/DB compression/decompression routines is available. |
| 60 | A record occurrence type is inconsistent with the set named in the ERROR-SET field in the IDMS communications block. This code usually indicates a broken chain. |
| 61 | No record can be found for an internal db-key. This code usually indicates a broken chain. |
| 62 | A system-generated db-key points to a record occurrence, but no record with that db-key can be found. This code usually indicates a broken chain. |
| 63 | The DBMS cannot interpret the DML function to be performed. When combined with a major code of 00, this code means invalid function parameters have been passed on the call to the DBMS. For LRF users, a WHERE clause includes a keyword that is longer than the 32 characters allowed. |

| Code | Database Function Status |
|-------------|--|
| 64 | The record cannot be found; the CALC control element has not been defined properly in the subschema. |
| 65 | The database page read was not the page requested. |
| 66 | The area specified is not available in the requested usage mode. |
| 67 | The subschema invoked does not match the subschema object tables. |
| 68 | The CICS interface was not started. |
| 69 | A BIND RUN-UNIT may not have been issued; the CV may be inactive or not accepting new run units; or the connection with the CV may have been broken due to time out or other factors. When combined with a major code of 00, this code means the program has been disconnected from the DBMS. |
| 70 | The database will not ready properly; a JCL error is the probable cause. |
| 71 | The page range or page group for the area being readied or the page requested cannot be found in the DMCL. |
| 72 | There is insufficient memory to dynamically load a subschema or database procedure. |
| 73 | A central version run unit will exceed the MAXERUS value specified at system generation. |
| 74 | The dynamic load of a module has failed. If operating under the central version, a subschema or database procedure module either was not found in the data dictionary or the load (core image) library or, if loaded, will exceed the number of subschema and database procedures provided for at system generation. |
| 75 | A read error has occurred. |
| 76 | A write error has occurred. |
| 77 | The run unit has not been bound or has been bound twice. When combined with a major code of 00, this code means either the program is no longer signed on to the subschema or the variable subschema tables have been overwritten. |
| 78 | An area wait deadlock has occurred. |
| 79 | The run unit has requested more db-key locks than are available to the system. |
| 80 | The target node is either not active or has been disabled. |
| 81 | The converted subschema requires specified database name to be in the DBNAME table. |
| 82 | The subschema must be named in the DBNAME table. |
| 83 | An error has occurred in accessing native VSAM data sets. |

| Code | Database Function Status |
|-------------|---|
| 87 | The owner and member records for a set to be updated are not in the same page group or do not have the same db-key radix. |
| 91 | The subschema requires a DBNAME to do the bind run unit. |
| 92 | No subschema areas map to DMCL. |
| 93 | A subschema area symbolic was not found in DMCL. |
| 94 | The specified dbname is neither a dbname defined in the DBNAME table, nor a SEGMENT defined in the DMCL. |
| 95 | The specified subschema failed DBTABLE mapping using the specified dbname. |

8.3 Major DC status codes

| Code | CA-IDMS/DC Function |
|-------------|---|
| 00 | Any DML statement |
| 30 | TRANSFER CONTROL |
| 31 | WAIT/POST |
| 32 | GET STORAGE/FREE STORAGE |
| 33 | SET ABEND EXIT/ABEND CODE |
| 34 | LOAD/DELETE TABLE |
| 35 | GET TIME/SET TIMER |
| 36 | WRITE LOG |
| 37 | ATTACH/CHANGE PRIORITY |
| 38 | BIND/ACCEPT/END TRANSACTION STATISTICS |
| 39 | ENQUEUE/DEQUEUE |
| 40 | SNAP |
| 43 | PUT/GET/DELETE SCRATCH |
| 44 | PUT/GET/DELETE QUEUE |
| 45 | BASIC MODE TERMINAL MANAGEMENT |
| 46 | MAPPING MODE TERMINAL MANAGEMENT |
| 47 | LINE MODE TERMINAL MANAGEMENT |
| 48 | ACCEPT/WRITE PRINTER |
| 49 | SEND MESSAGE |
| 50 | COMMIT TASK/ROLLBACK TASK/FINISH TASK/WRITE JOURNAL |
| 51 | KEEP LONGTERM |

8.4 Minor DC status codes

| Code | CA-IDMS/DC Function Status |
|-------------|--|
| 00 | Combined with a major code of 00, this code indicates either successful completion of the DML function or that all tested resources have been enqueued. |
| 01 | The requested operation cannot be performed immediately; waiting will cause a deadlock. |
| 02 | Either there is insufficient storage in the storage pool or the storage required for control blocks is unavailable. |
| 03 | The scratch area ID cannot be found. |
| 04 | Either the queue ID (header) cannot be found or a paging session was in progress when a second STARTPAGE command was received (that is, an implied ENDPAGE was processed before this STARTPAGE was executed successfully). |
| 05 | The specified scratch record ID or queue record cannot be found. |
| 06 | No resource control element (RCE) exists for the queue record; currency has not been established. |
| 07 | Either an I/O error has occurred or the queue upper limit has been reached. |
| 08 | The requested resource is not available. |
| 09 | The requested resource is available. |
| 10 | New storage has been assigned. |
| 11 | A maximum task condition exists. |
| 12 | The named task code is invalid. |
| 13 | The named resource cannot be found. |
| 14 | The requested module is defined as nonconcurrent and is currently in use. |
| 15 | The named module has been overlaid and cannot be reloaded immediately. |
| 16 | The specified interval control element (ICE) address cannot be found. |
| 17 | The record has been replaced. |
| 18 | No printer terminals have been defined for the current CA-IDMS/DC system. |
| 19 | The return area is too small; data has been truncated. |
| 20 | An I/O, program-not-found, or potential-deadlock status condition exists. |

| Code | CA-IDMS/DC Function Status |
|-------------|---|
| 21 | The message destination is undefined, the longterm ID cannot be found, or a KEEP LONGTERM request was issued by a nonterminal task. |
| 22 | A record already exists for the scratch area specified. |
| 23 | No storage or resource control element (RCE) could be allocated for the reply area. |
| 24 | The maximum number of outstanding replies has been exceeded. |
| 25 | An attention interrupt has been received. |
| 26 | There is a logical error in the output data stream. |
| 27 | A permanent I/O error has occurred. |
| 28 | The terminal dial-up line is disconnected. |
| 29 | An invalid parameter has been passed in the list set up by the DML processor. |
| 30 | The named function has not yet been implemented. |
| 31 | An invalid parameter has been passed; the TRB, LRB, or MRB contains an invalid field; or the request is invalid because of a possible logic error in the application program. In a DC-BATCH environment, a possible cause is that the record length specified by the command exceeds the maximum length based on the packet size. |
| 32 | The derived length of the specified variable storage is negative or zero. |
| 33 | Either the named table or the named map cannot be found in the data dictionary load area. |
| 34 | The named variable-storage area must be an 01-level entry in the LINKAGE SECTION. |
| 35 | A GET STORAGE request is invalid because the LINKAGE SECTION variable has already been allocated. |
| 36 | The program either was not defined during CA-IDMS/DC sysgen or is marked out-of-service. |
| 37 | A GET STORAGE operand is invalid because the specified variable storage area is in the WORKING-STORAGE SECTION instead of the LINKAGE SECTION. |
| 38 | Either no GET STORAGE operand was specified or the specified LINKAGE SECTION variable has not been allocated. |
| 39 | The terminal device being used is out of service. |
| 40 | NOIO has been specified but the datastream cannot be found. |
| 41 | An IF operation resulted in a valid true condition. |
| 42 | The named map does not support the terminal device in use. |

| Code | CA-IDMS/DC Function Status |
|-------------|---|
| 43 | A line I/O session has been cancelled by the terminal operator. |
| 44 | The referenced field does not participate in the specified map; a possible cause is an invalid subscript. |
| 45 | An invalid terminal type is associated with the issuing task. |
| 46 | A terminal I/O error has occurred. |
| 47 | The named area has not been readied. |
| 48 | The run unit has not been bound. |
| 49 | NOWAIT has been specified but WAIT is required. |
| 50 | Statistics are not being kept. |
| 51 | A lock manager error occurred during processing of the request. |
| 52 | The specified table is missing or invalid. |
| 53 | An error occurred from a user-written edit routine. |
| 54 | Either there is invalid internal data or a data conversion error has occurred. |
| 55 | The user-written edit routine cannot be found. |
| 56 | No DFLDS have been defined for the map. |
| 57 | The ID cannot be found, is not a long-term permanent ID, or is being used by another run unit. |
| 58 | Either the LRID cannot be found or the maximum number of concurrent task threads was exceeded. |
| 59 | An error occurred in transferring the KEEP LONGTERM request to IDMSKEEP. |
| 60 | The requested KEEP LONGTERM lock id is already in use with a different page group. |
| 61 | The requested KEEP LONMGTERM lock id is already in use with a different DBKey format. |
| 63 | Invalid function parameters have been passed on the call to the DBMS. |
| 64 | No detail exists currently for update; no action has been taken. Alternatively, the requested node for a header or detail is either not present or not updated. |
| 68 | There are no more updated details to MAP IN or the amount of storage defined for pageable maps at sysgen is insufficient. In the latter case, subsequent MAP OUT DETAIL statements are ignored. |

| Code | CA-IDMS/DC Function Status |
|-------------|--|
| 72 | No detail occurrence, footer, or header fields exist to be mapped out by a MAP OUT RESUME command, or the scratch record that contains the requested detail could not be accessed. The latter case is a mapping internal error and should be reported. |
| 76 | The first screen page has been transmitted to the terminal. |
| 77 | Either the program is no longer signed on to the subschema or the variable subschema tables have been overwritten. |
| 80 | The target node is either not active or has been disabled. |
| 96 | There are too many active run units for the internal table. |
| 97 | An invalid status has been received from DBIO/DBMS; check the CA-IDMS/DC log file for details. |
| 98 | An unsupported COBOL compiler option (for example, DEBUG) may have been specified. |
| 99 | An unexpected internal return code has been received; the terminal device is out of service. |

8.5 ERROR-STATUS condition names

| Code | Condition name | Explanation |
|------------------------------|-----------------------|--------------------------------|
| 0000 | DB-STATUS-OK | No error |
| 0307 | DB-END-OF-SET | End of set, area, or SPF index |
| 0326 | DB-REC-NOT-FOUND | No record found |
| 0001 to 9999 | ANY-ERROR-STATUS | Any non-zero status |
| 0000 to 9999 | ANY-STATUS | Any status |
| 3101 3201 3401 3901 | DC-DEADLOCK | Waiting will cause a deadlock |
| 3202 3402 | DC-NO-STORAGE | Insufficient space available |
| 4303 | DC-AREA-ID-UNK | ID cannot be found |
| 4404 | DC-QUEUE-ID-UNK | Queue header cannot be found |
| 4305 4405 | DC-REC-NOT-FOUND | Record cannot be found |
| 3908 | DC-RESOURCE-NOT-AVAIL | Resource not available |
| 3909 | DC-RESOURCE-AVAIL | Resource is available |
| 3210 | DC-NEW-STORAGE | New space allocated |
| 3711 | DC-MAX-TASKS | Maximum attached tasks |
| 4317 | DC-REC-REPLACED | Record has been replaced |
| 4319 4419 4519 4719 | DC-TRUNCATED-DATA | Return area too small |
| 4525 4625 | DC-ATTN-INT | Attention interrupt received |
| 4743 | DC-OPER-CANCEL | Session cancelled |

8.6 IDMS-DC/UCF DMLA run-time R(15) values

| R15 Value | DML Verb | Return Condition |
|-----------|-----------|--|
| X'00' | All verbs | No error |
| | #ENQ | <ul style="list-style-type: none"> ■ ACQUIRE — All requested resources have been acquired. ■ TEST — All tested resources have already been enqueued by the issuing task with the EXCLUSIVE/SHARED option specified by the test request. |
| | #SETIME | The request to cancel a previously issued #SETIME has been serviced successfully. |
| X'04' | #ATTACH | The maximum number of tasks has already been attached; no new tasks can be attached at this time. |
| | #COMMIT | Internal run-unit table full; check the CA-IDMS/DC log for details. |
| | #DELQUE | The parameter list is invalid. |
| | #DELSCR | The parameter list is invalid. |
| | #DEQ | At least one resource id (RSCID) cannot be found; all that were located have been dequeued. |
| | #ENQ | <ul style="list-style-type: none"> ■ ACQUIRE — At least one of the resources indicated is currently owned by another task and is not available for the EXCLUSIVE/SHARED option specified; no new resources have been acquired. ■ TEST — At least one of the tested resources is owned by another task and is not available to this task for the EXCLUSIVE/SHARED option specified. |
| | #FINISH | There are too many run units for the internal run-unit table. This is a system internal error and should be reported. |
| | #GETQUE | The parameter list is invalid. |
| | #GETSCR | The parameter list is invalid. |
| | #GETSTG | The request specified a storage id that did not previously exist; the indicated space has been allocated. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|---|
| | #LINEIN | The input area specified for return of data to the issuing program is too small to accommodate the full data stream; the returned data has been truncated accordingly. |
| | #LINK | Either the request cannot be serviced because of an I/O, program-not-found, or potential deadlock error or no null program definition elements (PDEs) have been allocated. If the load fails, the link will fail and a minor code will be returned in register 1. |
| | #LOAD | There is not enough space in the program pool to load the program. |
| | #MREQ | The specified edit or code table cannot be found or is invalid for use with the named map. |
| | #PRINT | An I/O error occurred during processing. |
| | #PUTJRNL | The derived journal record length is zero or negative. |
| | #PUTQUE | Invalid #PUTQUE request. Check for proper queue-id specification and logical selection of options. |
| | #PUTSCR | Invalid request. Check for proper scratch-id specification and logical selection of options as specified in the #PUTSCR statement. |
| | #ROLLBAK | Internal run-unit table full; check the CA-IDMS/DC log for details. |
| | #SENDMSG | An I/O error occurred during processing. |
| | #SETIME | For a #SETIME TYPE=CANCEL request, the internal control element (ICE) address specified cannot be found. |
| | #STRTPAG | A paging session was already in progress when another #STRTPAG command was issued. An implied #ENDPAG has been processed and the #STRTPAG has been executed successfully. |

| R15 Value | DML Verb | Return Condition |
|-----------|----------|---|
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, the input area specified for the return of data to the issuing program is too small to accommodate the full data stream; the returned data has been truncated accordingly. |
| | #TRNSTAT | A new transaction statistics block (TSB) has been allocated. |
| X'08' | #ATTACH | The requested task code is invalid. |
| | #COMMIT | An invalid request has been issued. #COMMIT is valid only if the program accesses CA-IDMS/DB database or dictionary entities (that is, CA-IDMS/DB records or DC/UCF scratch/queue records). Typically, #COMMIT need be specified only when CA-IDMS/DB database or dictionary entities are accessed in an update usage mode. |
| | #DELQUE | The requested queue header record (QUEID) cannot be found. |
| | #DELSER | The requested scratch area id (SAID) cannot be found. |
| | #ENQ | <ul style="list-style-type: none"> ■ ACQUIRE — Not applicable. ■ TEST — At least one of the tested resources is not already owned by any task and is available for the EXCLUSIVE/SHARED option specified. If both conditions described for return codes X'04' and X'08' exist, the register 15 value will be X'04'. |
| | #FINISH | An invalid request has been issued. #FINISH is only valid if the program accesses CA-IDMS/DB database or dictionary entities (that is, CA-IDMS/DB records or DC/UCF scratch/queue records). #FINISH need be specified only when the program performs database or dictionary accessing activities. |
| | #GETQUE | The requested queue header record (QUEID) cannot be found. |
| | #GETSER | The requested scratch area id (SAID) cannot be found. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|--|
| | #GETSTG | There is insufficient storage in the storage pool to process the request. |
| | #LINEIN | The I/O session has been canceled; the terminal operator has pressed the CLEAR (3270), ATTENTION (2741), or BREAK (teletype) key. |
| | #LINEOUT | The I/O session has been canceled; the terminal operator has pressed the CLEAR (3270), ATTENTION (2741), or BREAK (teletype) key. |
| | #LOAD | An I/O error occurred during a load from a load library. |
| | #MREQ | I/O has been interrupted; the terminal operator has pressed the ATTENTION (2741) or CLEAR (3270) key. |
| | #PRINT | The parameter list passed to #PRINT contains an invalid field. |
| | #PUTJRNL | The required storage is not available for the necessary control blocks. |
| | #ROLLBAK | An invalid request has been issued. There is a possible logic error in the program. Ensure that checkpoints are made (by means of #COMMIT) in the program logic before the #ROLLBAK request. |
| | #SENDMSG | The parameter list is invalid. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, output has been interrupted; the terminal operator has pressed the ATTENTION (2741) or CLEAR (3270) key. |
| | #TRNSTAT | Storage for the transaction statistics block (TSB) is not available; waiting would cause a deadlock. |
| | #WAIT | Waiting on the specified ECBs would cause a deadlock. |
| X'0C' | #ATTACH | The request cannot be serviced due to a security violation. |
| | #COMMIT | An invalid status has been issued from DBIO/DBMS; check the CA-IDMS/DC log for details. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|--|
| | #DELQUE | The requested queue record cannot be found |
| | #DELSCR | The requested scratch record id (SRID) cannot be found within the named SAID. |
| | #ENQ | <ul style="list-style-type: none"> ■ ACQUIRE — A requested resource cannot be enqueued immediately and waiting would cause a deadlock; no new resources have been acquired. ■ TEST — Not applicable. |
| | #FINISH | An invalid status has been issued from DBIO/DBMS; check the CA-IDMS/DC log for details. |
| | #GETQUE | The requested queue record cannot be found. |
| | #GETSCR | The requested scratch record id (SRID) cannot be found within the named SAID. |
| | #GETSTG | The parameter list is invalid. |
| | #LINEIN | A logical or permanent I/O error has been encountered in the input data stream. |
| | #LINEOUT | A logical or permanent I/O error has been encountered in the output data stream. |
| | #LOAD | The requested program is nonconcurrent and in use. |
| | #MREQ | A logical error (for example, invalid control character) has been encountered in the output data stream. |
| | #PRINT | No printer logical terminals have been defined in this DC/UCF system. |
| | #PUTJRNL | An invalid error status has been issued from DBIO/DBMS; check the IDMS/DC log for details. |
| | #ROLLBAK | An invalid error status has been issued from DBIO/DBMS; check the IDMS/DC log for details. |
| | #SENDMSG | The message destination is undefined. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request a logical error (for example, invalid control character) has been encountered in the output data stream. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|---|
| | #TRNSTAT | No transaction statistics block (TSB) exists; #TRNSTAT TYPE=BIND has not been issued. This return code is valid only for #TRNSTAT TYPE=ACCEPT and #TRNSTAT TYPE=END statements. |
| X'10' | #DELQUE | No resource control element (RCE) exists for the queue record; currency has not been established. |
| | #GETSTG | The requested storage cannot be allocated immediately (insufficient storage) and waiting would cause a deadlock. |
| | #LINEIN | The line request block (LRB) contains an invalid field. |
| | #LINEOUT | The line request block (LRB) contains an invalid field. |
| | #LOAD | The requested program has been temporarily overlaid in the program pool, resulting in a storage conflict. |
| | #MREQ | A permanent I/O error occurred during processing. |
| | #PRINT | A print screen request has been made from a non-3270-type terminal or from a 3270-type terminal without read buffer support. |
| | #PUTSCR | The request to replace a scratch record has been serviced successfully. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, a permanent I/O error occurred during processing. |
| | #TRNSTAT | Either the task in question is not associated with a terminal or the request is invalid. |
| X'14' | #LINEOUT | The name specified for DESTID, USERID, or LTERMID is unknown to this DC/UCF system. |
| | #LOAD | The requested program is not defined to the program definition table (PDT), the requested program is marked as out of service, or a null program definition element (PDE) could not be allocated for the program. |
| | #MREQ | The dial-up line for the terminal is disconnected. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|--|
| | #PRINT | Either the specified printer destination is invalid or, for OPTNS=DIRECT, LTEID or LTEADDR is invalid. |
| | #PUTSCR | The request to add a new scratch record cannot be processed because the record id specified by the SRID operand already exists for the named scratch area. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, the dial-up line for the terminal is disconnected. |
| | #TRNSTAT | Transaction statistics or task statistics are not enabled in this DC/UCF system. |
| X'18' | #GETQUE | The user area specified for the return of the queue record is too small; the returned record has been truncated to fit in the available storage space. |
| | #GETSCR | The user area specified for the return of the scratch record is too small; the returned record has been truncated to fit in the available storage space. |
| | #GETSTG | Allocated XA storage above the 16 megabyte line cannot be addressed by a 24-bit task. |
| | #LOAD | The requested program cannot be loaded immediately due to insufficient space; waiting would cause a deadlock. |
| | #MREQ | The terminal being used is out of service. |
| | #PRINT | A terminal I/O error occurred during a #PRINT request. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, the terminal being used is out of service. |
| X'1C' | #DELQUE | An I/O error occurred during a delete queue operation. |
| | #DELSCR | An I/O error occurred during a delete scratch operation. |
| | #GETQUE | An I/O error occurred during get queue processing. |
| | #GETSCR | An I/O error occurred during get scratch processing. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|---|
| | #PRINT | No printer can be found to satisfy the print-direct request and OPTNS=NOWAIT has been specified. |
| | #PUTSCR | An I/O error occurred during processing. |
| | #TREQ | For a #TREQ GET, #TREQ PUTGET, or #TREQ CHECK request, the terminal is closed or was never opened. |
| X'20' | #ATTACH | The maximum number of concurrent tasks has been reached. |
| | #LOAD | An I/O error occurred during a load from the dictionary DDLDCLOD area. |
| | #MREQ | The map request block (MRB) contains an invalid field, indicating a possible error in application program parameters. |
| | #PRINT | The print-direct request specified an LTEID or LTEADDR that is out of service. |
| | #TREQ | The terminal request block (TRB) contains an invalid field. |
| X'24' | #MREQ | The map load module requested by the map request block (MRB) cannot be found. |
| | #PRINT | The print-direct request specified a wait; waiting would cause a deadlock. |
| | #TREQ | The name specified for DESTID, LTERMID, or USERID is invalid. |
| X'28' | #MREQ | The requested map does not support the terminal device type being used. |
| | #PRINT | A DCMT VARY PRINTER CANCEL command has been issued in the DC/UCF system for this direct printer. |
| X'2C' | #MREQ | An error was detected upon return from a user-written edit module. An invalid pointer to the data stream has been returned to register 1. |
| | #PRINT | A DCMT VARY PRINTER REQUEUE command had been issued in the DC/UCF system for this direct printer. |

| R15 Value | DML Verb | Return Condition |
|------------------|-----------------|--|
| X'30' | #MREQ | Invalid internal data has been encountered. Either the data in the record does not match the internal data or the internal data cannot be converted to the external format, as specified in the external picture. |
| X'34' | #MREQ | The named user-written edit module cannot be found. |
| X'38' | #MREQ | An invalid immediate write request to DESTID, LTERMID, or USERID has been issued. |
| X'3C' | #MREQ | The map load module is invalid. |
| X'40' | #MREQ | For an #MREQ IN request, the requested node for a header or detail was either not present or not updated. For an #MREQ OUT request, there is no current detail occurrence to be updated. No action is taken. |
| X'44' | #MREQ | No more modified detail occurrences require a mapin. For an #MREQ OUT request, the maximum amount of storage defined for pageable maps during system generation is insufficient. |
| X'48' | #MREQ | For an #MREQ IN request, the scratch record that contains the requested detail could not be accessed (internal error). For an #MREQ OUT,RESUME request, no detail occurrence, footer, or header fields exist. |
| X'4C' | #MREQ | For an #MREQ OUT request, the first screen page has been transmitted to the terminal. |
| X'50' | #MREQ | An #MREQ IN,COND=MPNS or #MREQ OUT,COND=MPNS request has been received when no map paging session is in progress. Either a #STRTPAG command was not issued prior to this #MREQ IN command or a #ROLLBAK was issued that rendered the scratch area for the pageable map (area id MPGPSCRA) unavailable. If the COND specification is not MPNS, this condition abends the map paging task. |

Chapter 9. TP monitor coding requirements

| TP MONITOR | IF MODE IS | | | | PROCEDURE DIVISION |
|--|---------------------------------|----------------------|--|---|---|
| | | IDMS-CONTROL SECTION | WORKING-STORAGE SECTION | LINKAGE SECTION | |
| CICS | CICS | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-NAMES. | *01 TWA 03 FILLER PIC S9(8) COMP SYNC. 03 COPY IDMS SUBSCHEMA-CTRL. 03 COPY IDMS SUBSCHEMA-RECORDS. OR **COPY IDMS SUBSCHEMA-CTRL. COPY IDMS SUBSCHEMA-RECORDS. <u>(A CICS GETMAIN must be issued for the SUBSCHEMA-CTRL and for each record being copied.)</u> | COPY IDMS IDMS-WAIT. |
| | CICS-EXEC CICS-EXEC-AUTO | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-CTRL. COPY IDMS SUBSCHEMA-NAMES. COPY IDMS SUBSCHEMA-RECORDS. | | |
| INTERCOMM | INTERCOMM INTERCOMM-AUTO | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-NAMES. | COPY IDMS SUBSCHEMA-CTRL. COPY IDMS SUBSCHEMA-RECORDS. | |
| SHADOW | SHADOW SHAD-AUTOSTATUS | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-NAMES. | COPY IDMS SUBSCHEMA-CTRL. COPY IDMS SUBSCHEMA-RECORDS. | |
| UTM | UTM UTM-AUTOSTATUS | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-NAMES. | COPY KCKBC. 05 X PIC S9(8) COMP SYNC. 05 COPY IDMS SUBSCHEMA-CTRL. 05 COPY IDMS SUBSCHEMA-RECORDS. COPY KCPAC. | MOVE LOW-VALUES to SUBSCHEMA-CTRL before each BIND RUN-UNIT. |
| WESTI | WESTI-REENT WESTI-REENT-AUTO | IDMS-RECORDS MANUAL. | COPY IDMS SUBSCHEMA-NAMES. | COPY IDMS SUBSCHEMA-CTRL. COPY IDMS SUBSCHEMA-RECORDS. | |
| <p>2 * *If SUBSCHEMA-CTRL, SUBSCHEMA-RECORDS, and additional data does not exceed 4,096 bytes. **If SUBSCHEMA-CTRL, SUBSCHEMA-RECORDS, and additional data exceeds 4,096 bytes.</p> | | | | | |

Chapter 10. Record/set representation

Format

| | | | |
|-----------------------------|------------------|----------------------|----------------------|
| <i>record-name</i> | | | |
| <i>record-ID</i> | <i>stor mode</i> | <i>record length</i> | <i>location-mode</i> |
| <i>calc-key-or-set-name</i> | | | <i>dup opt</i> |
| <i>area-name</i> | | | |

*set-name
pointers
membership
order*

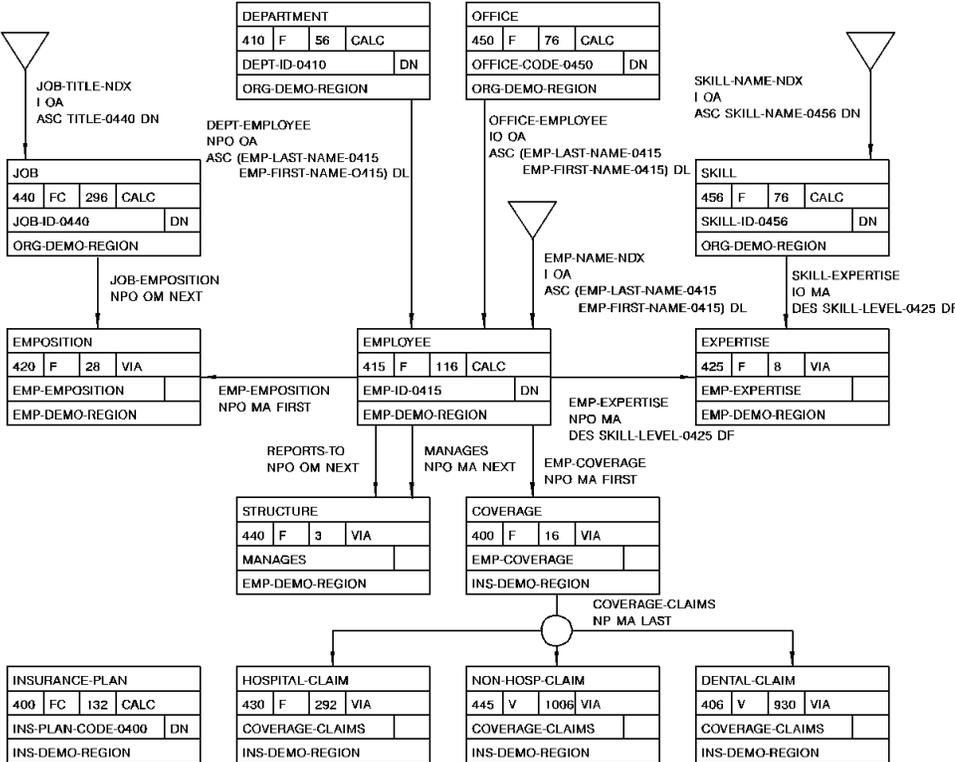
| | | | |
|--------------------|------------------|----------------------|----------------------|
| <i>record-name</i> | | | |
| <i>record-ID</i> | <i>stor mode</i> | <i>record length</i> | <i>location-mode</i> |

- stor mode* = F, FC, V, or VC
- location-mode* = CALC, VIA, DIRECT, VSAM, or VSAM CALC
- dup opt* = DN, DF, DL, or DU
- pointers* = N, NP, NO, NPO, I, or IO
- membership* = MA, MM, OA, or OM
- order*
 - for unsorted sets* = FIRST, LAST, NEXT, or PRIOR
 - for sorted sets* = ASC or DES/ *sort-key/dup opt for sort-key*

Example

| | | | |
|---|----|-----|------|
| STUDENT | | | |
| 108 | FC | 452 | CALC |
| STUD-ID | | | DN |
| STUDENT-REGION | | | |
| STUDENT-SCHEDULE NP MA ASC SCHED-PERIOD DN | | | |
| SCHEDULE | | | |
| 107 | F | 24 | VIA |
| STUDENT-SCHEDULE | | | |
| STUDENT-REGION | | | |

Chapter 11. EMPLOYEE database structure diagram



Chapter 12. Online debugger syntax

12.1 Debugger symbols

General registers and the PSW: **General registers** include the registers used by the program at the time of execution and the registers used by the DC/UCF system. The program status word (PSW) and register definitions are always preceded by a colon (:), and are specified by these symbols:

- **:PSW** for the current program status word
- **:R n** for the user program register at the time of interrupt, where n represents the number of the register and can have a value of 0 through 15
- **:REGS** for all user program registers at the time of interrupt
- **:SR n** for a DC/UCF system register at the time of interrupt, where n represents the number of the register and can have a value of 0 through 15
- **:SREGS** for all DC/UCF system registers at the time of interrupt

DC/UCF system symbols

| | |
|-------|---|
| :BAT | Base address table for session |
| :CSA | DC/UCF common storage area |
| :DLB | Debug local block, control block required for debugging session |
| :LTE | Current logical terminal element |
| :PTE | Current physical terminal element |
| :TCE | Current task control element |
| :VECT | Vector table for debugger |

Address symbols and markers

| Symbol | Symbol name | Designated location |
|--------|-------------|-----------------------------------|
| @ | At sign | Absolute address |
| \$ | Dollar sign | Load address |
| ¢ | Cent sign | Address of current dialog process |

12.2 User symbols

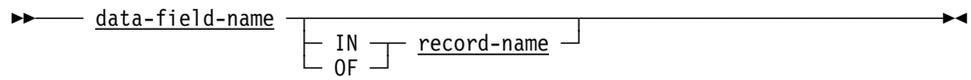
- **:DR n** for a debugger general register, where n represents the number of the register and can have a value of 0 through 15
- **:DREGS** for all debugger registers
- **:H1** and **:H2** for halfword 1 and halfword 2
- **:F1** and **:F2** for fullword 1 and fullword 2
- **:UCHR** for a 48-byte character area

You can also refer to specified sections of this area:

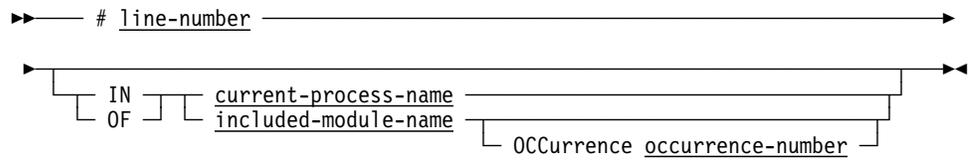
- **:UC0**, the first 16 bytes
- **:UC16**, the next 16 bytes
- **:UC32**, the last 16 bytes

12.3 Program symbols

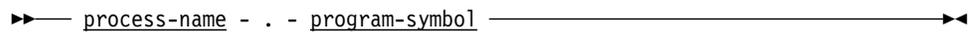
Data Field names



Line numbers



Qualifying program symbols



12.4 Expression operators

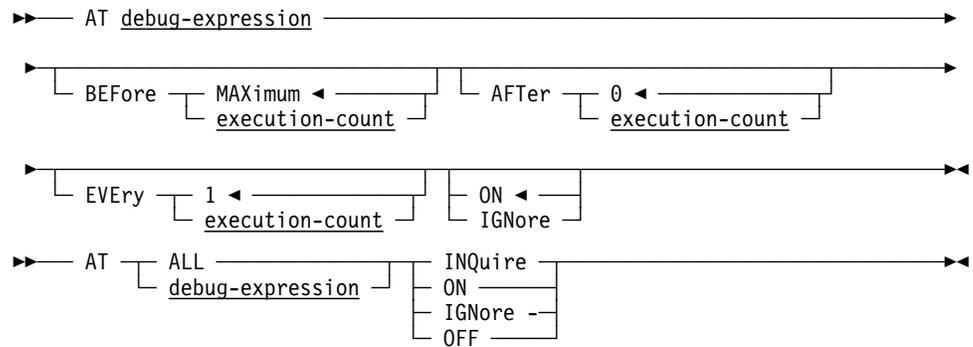
| Operator | Meaning |
|-----------------|----------------|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |

12.5 Delimiters

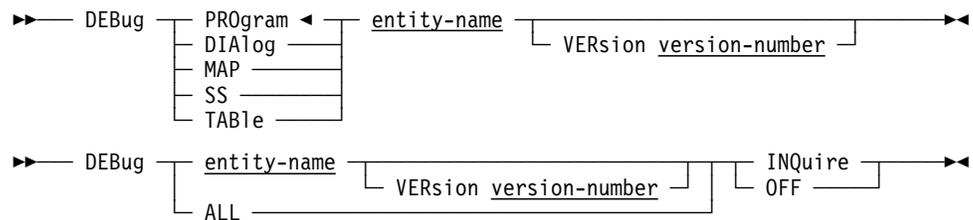
| | |
|---|-------------------|
| * | Asterisk |
| | Blank |
| , | Comma |
| = | Equal sign |
| ! | Exclamation point |
| - | Hyphen |
| % | Percent sign |
| . | Period |
| + | Plus sign |
| / | Slash |

12.6 Debugger commands

AT



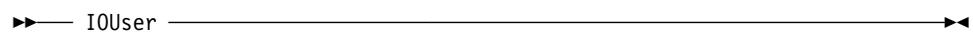
DEBUG



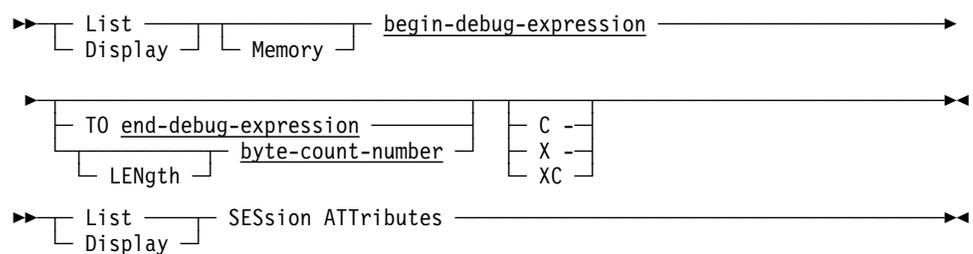
EXIT



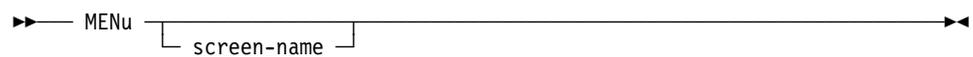
IOUSER



LIST



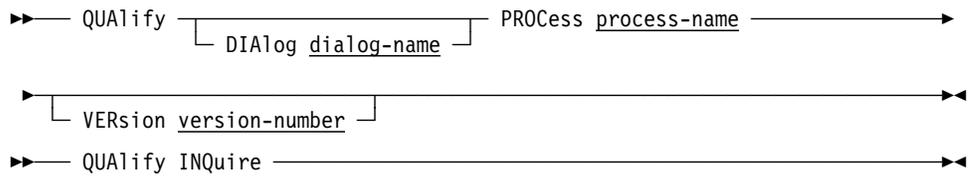
MENU



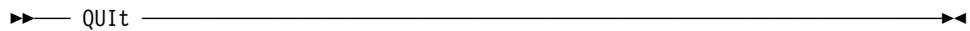
PROMPT



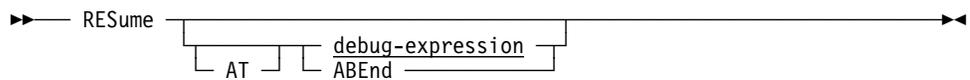
QUALIFY



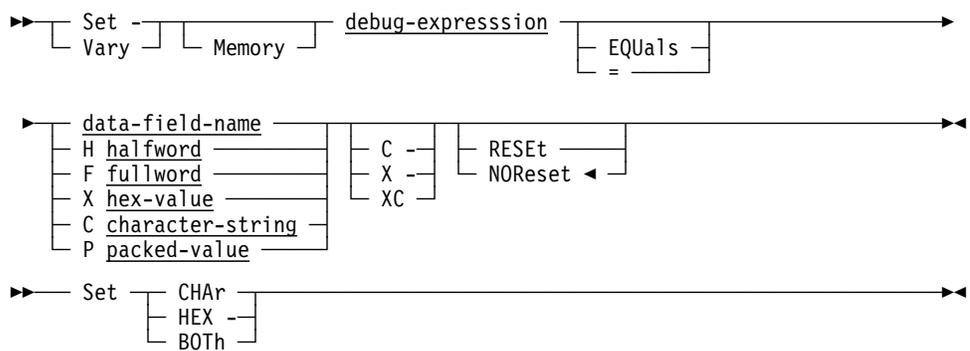
QUIT



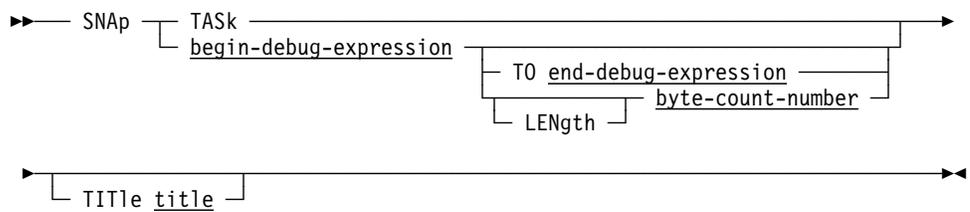
RESUME



SET



SNAP



WHERE

