

Cincom

SUPRA SERVER PDM

DBA Utilities Quick Reference
(OS/390 & VSE)

P26-6270-63



SUPRA[®] Server PDM DBA Utilities Quick Reference (OS/390 & VSE)

Publication Number P26-6270-63

© 1986, 1989, 1993, 1994, 1996, 1998, 2000, 2002 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage [®]	iD CinDoc [™]	MANTIS [®]
C+A-RE [™]	iD CinDoc Web [™]	Socrates [®]
CINCOM [®]	iD Consulting [™]	Socrates [®] XML
Cincom Encompass [®]	iD Correspondence [™]	SPECTRA [™]
Cincom Smalltalk [™]	iD Correspondence Express [™]	SUPRA [®]
Cincom SupportWeb [®]	iD Environment [™]	SUPRA [®] Server
CINCOM SYSTEMS [®]	iD Solutions [™]	Visual Smalltalk [®]
 gOOj [™]	intelligent Document Solutions [™]	VisualWorks [®]
	Intermax [™]	

UniSQL[™] is a trademark of UniSQL, Inc.
ObjectStudio[®] is a registered trademark of CinMark Systems, Inc.

All other trademarks are trademarks or registered trademarks of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U.S.A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: <http://www.cincom.com>

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *SUPRA Server PDM DBA Utilities Quick Reference (OS/390 & VSE)*, P26-6270-63, is dated January 15, 2002. This document supports Release 2.7 of SUPRA Server PDM in IBM mainframe environments.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the [survey](#) provided with the online documentation at your convenience.

Cincom Technical Support for SUPRA Server PDM

FAX: (513) 612-2000
Attn: SUPRA Server Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: SUPRA Server Support
55 Merchant Street
Cincinnati, OH 45246-3732
U.S.A.



Contents

Conventions	vii
UCL command language for control section	9
Matrix of control statements and functions	13
UCL functions and applications	14
Depopulate	15
Expand	16
File Statistics	17
Format	18
Load (Version 1)	19
Log-Print	20
Modify	21
Print	22
Recover and Restore	23
Reorganize	24
Review	25
Sorted-Populate	26
Unload (Version 1)	27
Unlock	28
Non-UCL utilities, functions, and applications	29
Execution Statistics Utility	30
Insert Linkpath Function, Version 2	31
File definitions	31
Control statements	32
Inter-Directory Copy Utility	32
Load Function, Version 2	33
File definitions	33
Run control statements	35
File control statements	38

- PDM Termination Utility 39
 - Input statements 39
- Unload Function, Version 2 40
 - File definitions 40
 - Run control statements 41
 - File control statements 44

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example
Slashed b (<i>b</i>)	Indicates a space (blank). The example indicates that four spaces appear between the keywords.	<code>BEGINbbbbSERIAL</code>
Brackets []	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations. A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a WHERE clause.	<code>[WHERE <i>search-condition</i>]</code>
	Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter either WAIT or NOWAIT. (WAIT is underlined to signify that it is the default.)	<code>[<u>(WAIT)</u> (NOWAIT)]</code>
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter ON or OFF when using the MONITOR statement.	<code>MONITOR {ON OFF}</code>

Convention	Description	Example
<p><u>Underlining</u></p>	<p>Indicates the default value supplied when you omit a parameter.</p> <p>The example indicates that if you do not choose a parameter, the system defaults to WAIT.</p> <p>Underlining also indicates an allowable abbreviation or the shortest truncation allowed.</p> <p>The example indicates that you can enter either STAT or STATISTICS.</p>	<p>[WAIT] [NOWAIT]</p> <hr/> <p><u>STATISTICS</u></p>
<p>Ellipsis points...</p>	<p>Indicate that the preceding item can be repeated.</p> <p>The example indicates that you can enter multiple host variables and associated indicator variables.</p>	<p>INTO :host-variable [:ind-variable],...</p>
<p>UPPERCASE</p>	<p>Indicates literals which you must enter exactly as shown.</p>	<p>KEYWORD</p>
<p><i>Italics</i></p>	<p>Indicate variables you replace with a value, a column name, a file name, and so on.</p> <p>The example indicates that you must substitute the name of a table.</p>	<p>FROM <i>table-name</i></p>
<p>Punctuation marks</p>	<p>Indicate required syntax that you must code exactly as presented.</p> <p>() parentheses . period , comma : colon ' ' single quotation marks</p>	<p>(<i>user-id</i>, <i>password</i>, <i>db-name</i>)</p> <p>INFILE 'Cust.Memo' CONTROL LEN4</p>

UCL command language for control section

CONTROL { **BEGIN** }
 { **END** }

ENV-DESC (*environment-description-name*)

SCHEMA (*schema-name*)

[FORMAT (**NO**)
 { **YES** }]

[DIAGNOSTICS (**ABEND**
 { **SIMPLE** }
 { **EXTENDED** })]

[LIST ([**ALL**] [**NONE**] [**AFTER**] [**BEFORE**] [**BLOCK**] [**SYSTEM**]
 [**FUNCTION**] [**DESCRIPTION**] [**APPLIED-IMAGES**])
 [**HEADER** (**NO**)
 { **YES** }]
 [**EXTENSION** ('*string*')]
 [**SUPPRESS** ([**ELEMENT**] [**SPACE**] [**REFER**])]]

 [**LINES** (**1**)
 { *nnn* }]

 [**DATA-FORMAT** (**HEX**)
 { **CHAR** }]]

```

[ SORT ( [ SORT
          [ program - name ] ] )
  [ MEMORY ( [ 120k
              [ nnnnnK ] ] ) ] ]

[ CONSOLE ( [ NO
             [ YES ] ] )
  [ [ NOTIFY ]
    [ REPLY ] ] ('operator - msg - text') ] ]

[ DATA - FILE ( [ CSUDATA
                 [ ddname ] ] )
  [ LABEL ( [ NO
            [ YES ] ] ) ]
  [ RECORD - FORMAT ( [ F
                      [ V
                      [ FB
                      [ VB ] ] ] ) ]
  [ RECORD - SIZE ( [ b
                    [ nnnnn ] ] ) ]
  [ BLOCK - SIZE ( [ b
                   [ nnnnn ] ] ) ]
  [ DEVICE ( [ DISK
             [ TAPE ] ] ) ] ]

[ SUMMARY - DATA ( [ ALL ] [ FILE ] [ NONE ] [ FUNCTION ]
                   [ CUMULATIVE ] ) ] ]

```

**LOG - FILE ([ddname]
[LOGFILE])**

**[ACCESS - METHOD ([BSAM]
[BDAM])
[ESDS]]**

**[DEVICE ([DISK]
[TAPE])
[VSAM]]**

**[DEVICE - ADDRESS ([SYS010]
[SYSnnn])]**

**[BLOCK - SIZE ([b]
[nnnnn])]**

**[SEQ - ERROR ([EOF]
[ERROR]
[IGNORE])
[WARNING]
[INFORMATION]]**

**[PDM - ID - ERROR ([EOF]
[ERROR]
[IGNORE])
[WARNING]
[INFORMATION]]**

**[LOG - ID - ERROR ([EOF]
[ERROR]
[IGNORE])
[WARNING]
[INFORMATION]]**

FUNCTION ({
 FORMAT
 SORTED - POPULATE
 DEPOPULATE
 REORGANIZE
 FILE - STATS
 EXPAND
 UNLOAD
 LOAD
 RECOVER
 RESTORE
 LOG - PRINT
 REVIEW
 UNLOCK
 PRINT
 MODIFY
}) ... **1+**

1+

You can code one or more functions.

Matrix of control statements and functions

CONTROL STATEMENTS	FUNCTION														
	Format	Sorted-Populate	Depopulate	Reorganize	File Statistics	Expand	Unload	Load	Print	Modify	Recover	Restore	Log-Print	Review	Unlock
CONTROL	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
ENV-DESC	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SCHEMA	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
FORMAT								O							
DIAGNOSTICS	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
LIST							O	O	O	O	O	O	O		
HEADER	O				O	O	O	O	O	O	O	O	O		
EXTENSION	O				O	O	O	O	O	O	O	O	O		
SUPPRESS									O	O					
LINES											O	O	O		
DATA-FORMAT							O	O	O	O	O	O	O		
SORT		O			O	O		O							
MEMORY		O			O			O							
CONSOLE											O	O	O		
NOTIFY											O	O	O		
REPLY											O	O	O		
DATA-FILE								O	O						
LABEL								O	O						
RECORD-FORMAT								O							
RECORD-SIZE								O							
BLOCK-SIZE								O							
DEVICE								O	O						
LOG-FILE											O	O	O		
ACCESS-METHOD											O	O	O		
DEVICE											O	O	O		
DEVICE-ADDRESS											O	O	O		
BLOCK-SIZE											O	O	O		
SEQ-ERROR											O	O	O		
PDM-ID-ERROR											O	O	O		
LOG-ID-ERROR											O	O	O		
SUMMARY-DATA		O			O	O	O	O	O	O					
FUNCTION	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

UCL functions and applications

Function	Application
Depopulate	Deletes secondary keys.
Expand	Expands the capacity of an existing related file. Use this function only with SUPRA native files.
File Statistics	Reports the physical and logical characteristics of the database file being examined. Use this function only with SUPRA native files.
Format	Formats a database file. Sets all records in the file to blanks and writes a file control record on the file. Use this function only with SUPRA native files.
Load (Version 1)	Formats a database file or files and writes data records from a sequential medium to the files. Use this function with SUPRA native files and to convert Series/80 or SUPRA converted files to the SUPRA native format.
Log-Print	Prints selected information from the System Log File without updating the database.
Modify	Updates records in a database file. This function works with SUPRA native, SUPRA converted, and Series/80 files.
Print	Prints records from a database file. This function works with SUPRA native, SUPRA converted, and Series/80 files.
Recover	Backs off updates to the database after an abend.
Reorganize	Corrects the deterioration of the tree structure that can result from updating the database.
Restore	Reapplies updates to the database after an abend.
Review	Determines whether database files are locked.
Sorted-Populate	Puts secondary key information from primary and related files into index files.
Unload (Version 1)	Extracts all or selected records from a database file and writes them to a sequential output medium. Use this function with SUPRA native files and to convert Series/80 or SUPRA converted files to the SUPRA native format.
Unlock	Unlocks database files after an abend.

Depopulate

FUNCTION (DEPOPULATE)

```

[ STATISTICS ( [ ALL
                BASE
                NONE ] ) ]

```

[STANDARD-EXIT (*exit-name*)]

```

FILE ( { ALL
        file-name-list } ) 1+

```

```

[ SECONDARY-KEY ( [ ALL
                   key-name-list ] ) ]
  PURGE ( [ NO
           YES ] ) O+

```

1+

You can code the FILE statement and its subordinate statements one or more times.

O+

You can code the SECONDARY-KEY statement and its subordinate statement more than once, or not at all.

Expand

FUNCTION (EXPAND)

FILE ({ ALL
file-name-list })

1+

1+

You can code the FILE statement and its subordinate statements one or more times.

File Statistics

FUNCTION (FILE-STATS)

FILE ({ ALL
file-name })

1+

[CLOSE ([NO
YES])]

[LINKPATH ([b
linkpath - list])]

[STATISTICS ([ALL] [BASE] [SIZE] [LINK] [CHAIN] [CODE])]

1+

You can code the FILE statement and its subordinate statements one or more times.

Format

FUNCTION (FORMAT)

FILE {
 ALL
 file-name-list
}

1+

1+

You can code the FILE statement and its subordinate statements one or more times.

Load (Version 1)

FUNCTION (LOAD)

[STANDARD-EXIT (*exit-name*)]

FILE ({ ALL
file-name }) 1+

[LINKPATH ({ b
access - linkpath })]

[CLEARLINKS (*linkpath - list*)]

[SEQUENCE (*sort - list*)]

[DIRECTION ({ ASCENT
DESCEND } - *list*)]

[DATA - TYPE ({ HEX
CHAR
ZONED - DEC
PACKED - DEC } - *list*)]

[RECORD ({ ALL
record-code })] 0+

[ELEMENT ({ ALL
element-list })]

1+

You can code the FILE statement and its subordinate statements one or more times.

0+

You can code the RECORD statement and its subordinate statements more than once, or not at all.

Log-Print

FUNCTION (LOG-PRINT)

[STANDARD-EXIT (*exit-name*)]

[STATISTICS ([ALL
BASE]
NONE)]

FILE ({ ALL
file-name })

1+

[RRN-RANGE ({ *low-rrn*
- *high-rrn*
low-rrn-high-rrn })]

KEY-RANGE
[D'*low-dec-key*] [-D'*hi-dec-key*]
([X'*low-hex-key*] [-X'*hi-hex-key*])
[C'*low-chr-key*] [-C'*hi-chr-key*]]

1+

You can code the FILE statement and its subordinate statements one or more times.

Modify

FUNCTION (MODIFY)

[STANDARD-EXIT] (*exit-name*)

FILE (*file-name*)

1+

[OPEN-MODE ($\left[\begin{array}{c} \text{SUPD} \\ \text{EUPD} \end{array} \right]$)]

[CLOSE ($\left[\begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right]$)]

QUALIFIER ($\left\{ \begin{array}{c} \text{DIRECT} \\ \text{SERIAL} \\ \text{SEQUENTIAL} \end{array} \right\}$)

[RRN (*record - rrn*)]

[LINKPATH (*access - linkpath*)]

[KEY ($\left\{ \begin{array}{c} \text{D' } dec - string' \\ \text{X' } hex - string' \\ \text{C' } char - string' \end{array} \right\}$)]

[MAXIMUM ($\left[\begin{array}{c} \text{b} \\ record - count \end{array} \right]$)]

[CRITERIA (*element*₁ [, *element*₂, ..., *element*_n]
 .*operator*.*datavalue*₁
 [.*datavalue*₂...*datavalue*_n]END.)]

[RECORD ($\left\{ \begin{array}{c} \text{ALL} \\ record - code \end{array} \right\}$)
 ELEMENT (*element - list*)
 DATA (.*data - string* END.)]

O+

1+

You can code the FILE statement and its subordinate statements one or more times.

O+

You can code the RECORD statement and its subordinate statements more than once, or not at all.

Print

FUNCTION (PRINT)
[STANDARD-EXIT (*exit-name*)]

FILE ({ ALL
file-name })

1+

[OPEN-MODE ([READ
IUPD
SUPD
EUPD])]

[CLOSE ([NO
YES])]

QUALIFIER ({ DIRECT
SERIAL
SEQUENTIAL })

[LINKPATH (*access-linkpath*)]

[KEY ({ D'*dec-string*'
X'*hex-string*'
C'*char-string*' })]

[RRN (*record-rrn*)]

[RRN-RANGE ([*low-rrn*
- *high-rrn*
low-rrn-high-rrn])]

[MAXIMUM ([b
record-count])]

[CRITERIA (*element*₁,*element*₂,...,*element*_n)
.operator.*datavalue*₁
[.*datavalue*₂...*datavalue*_n]END.)]

[RECORD ({ ALL
record-code })]

O+

[ELEMENT ({ ALL
element-list })]

1+

You can code FILE and its subordinate statements one or more times.

O+

You can code the RECORD statement and its subordinate statements more than once, or not at all.

Recover and Restore

```

FUNCTION ( { RECOVER
           }
          { RESTORE
          } )

STATE ( { LOG - END
        }
       { LOG - BEGIN
        }
       { LAST - COMMIT
        } )

[ STANDARD - EXIT (exit - name) ]

OPEN - FILE ( { INITIAL
              }
            { DYNAMIC
              } )

[ STATISTICS ( { ALL
               }
              { BASE
               }
              { NONE
               } ) ]

FILE ( { ALL
       }
      { file - name
       } ) 1+

[ RRN - RANGE ( { low - rrn
                 }
               { - high - rrn
                 }
               { low - rrn - high - rrn
                 } )
  KEY - RANGE
  [ D' low - dec - key' ][ -D' hi - dec - key' ]
  ( [ X' low - hex - key' ][ -X' hi - hex - key' ] )
  [ C' low - chr - key' ][ -C' hi - chr - key' ] ]

```

1+

You can code the FILE statement and its subordinate statements one or more times.

Reorganize

FUNCTION (REORGANIZE)

```
[ STATISTICS ( [ ALL
                BASE
                NONE ] ) ]
```

[STANDARD-EXIT (*exit-name*)]

```
FILE ( { ALL
        file-name-list } )
```

1+

```
[ SECONDARY-KEY ( [ ALL
                   key-name-list ] )
  [LOAD-DENSITY (0-99)] ]
```

0+

1+

You can code the FILE statement and its subordinate statements one or more times.

0+

You can code the SECONDARY-KEY statement and its subordinate statements more than once, or not at all.

Review

FUNCTION (REVIEW)

FILE ({ ALL
file-name-list })

1+

1+

You can code the FILE statement and its subordinate statements one or more times.

Sorted-Populate

FUNCTION (SORTED-POPULATE)

[STATISTICS ([ALL
BASE
NONE])]

[STANDARD-EXIT (*exit-name*)]

FILE ({ ALL
file-name-list })

1+

[SECONDARY-KEY ([ALL
key-name-list])
[LOAD-DENSITY (0-99)]]

O+

1+

You can code the FILE statement and its subordinate statements one or more times.

O+

You can code the SECONDARY-KEY statement and its subordinate statements more than once, or not at all.

Unload (Version 1)

FUNCTION (UNLOAD)

[STANDARD-EXIT (*exit-name*)]

FILE ({ ALL
file-name-list })

1+

[LINKPATH (b
access - linkpath)]
[PRESERVE (NO
YES)]

[CLEAR - LINKS (*linkpath - list*)]

[RRN - RANGE (*low - rrn*
- high - rrn
low - rrn - high - rrn)]

[CRITERIA (*element*₁, *element*₂, ..., *element*_n)
. *operator* . *datavalue*₁
[. *datavalue*₂ ... *datavalue*_n] END.)]

[RECORD (ALL
record-code)]
[ELEMENT ({ ALL
element-list })]

O+

1+

You can code the FILE statement and its subordinate statements one or more times.

O+

You can code the RECORD statement and its subordinate statements more than once, or not at all.

Unlock

FUNCTION (UNLOCK)

FILE { **ALL**
file-name-list }

Non-UCL utilities, functions, and applications

Function	Application
Execution Statistics Utility	Generates system statistics.
Insert Linkpath Function (Version 2)	Inserts linkpath data into Series/80, SUPRA converted, SUPRA native, or SUPRA Directory files without having to unload and reload the primary files. Input for this function must be the output of the Version 2 Load function.
Inter-Directory Copy Utility	Copies information from one SUPRA Directory to another.
Load Function (Version 2)	Formats Series/80, SUPRA converted, or SUPRA native database files. Writes data records at a high speed from a sequential medium (output from Version 2 Unload function) to the files.
PDM Termination Utility	Shuts down the PDM
Unload Function (Version 2)	Extracts records from Series/80, SUPRA converted, SUPRA native, or SUPRA Directory files at high speed and writes them to a sequential output medium appropriate for the Version 2 Load function.

Execution Statistics Utility

The Execution Statistics utility generates a statistics report. The report shows the contents of the statistics file and calculations based on the contents.

To execute this utility in OS/390 or VSE, see sample JCL member TXJPSTAT.

Insert Linkpath Function, Version 2

File definitions

File	Description
CSIPARM	Identifies the CSIPARM which contains control information needed for the PDM.
CSUAUX	<p>Holds the auxiliary information for the files not in native format.</p> <p>FILE Name of file to which parameters on the CSUAUX record apply.</p> <p>RCYL Number of records per logical cylinder.</p> <p>CYLL Cylinder load limit - maximum percentage of each logical cylinder to be filled with data records during the load function.</p> <p>LOAD Format in which the file is to be loaded.</p> <p><u>CSUAUX File Record Format</u></p> $\text{FILE} = \text{ffffRCYL} = \text{nnnnCYLL} = \left\{ \begin{array}{l} \mathbf{80} \\ \text{nnnn} \end{array} \right\}$ $\text{LOAD} = \left[\begin{array}{l} \mathbf{COMPATIBILITY} \\ \mathbf{CONVERTED} \\ \mathbf{NATIVE} \end{array} \right]$
LINKWK01 (OS/390) LNKWRK1 (VSE)	Specifies the first linkage work file which contains linkpath data generated in the related file load for insertion into associated primary files.
LINKWK02 (OS/390) LNKWRK2 (VSE)	Specifies the second linkage work file for synonym processing.
SYSIN (OS/390) SYSIPT (VSE)	Holds the run control statements.
SYSPRINT (OS/390) SYSLST (VSE)	Specifies the file for the printed listing of all control statements, diagnostics, and so on.
SYSDUMP (OS/390 only)	Specifies a dump file.
ffffff	Specifies the SUPRA primary files to be processed by the Insert Linkpath function.

Control statements

INSERT Statement

Identifies primary files in which information is to be inserted.

```

INSERT FILES={ { ALL.
                ffff 1 [ ffff 2... ffff n ]. } }

[,CLEARLKS=(ppppLK xx1[ppppLK xx2...ppppLK xxn].)]

[,DBMOD= schemaname]

,END.
    
```

LINKWKnn

VSE only. Defines the characteristics of the LNKWRK1 and LNKWRK2 files.

```

LINKWK nn: [ DEVICE = { DISK
                       TAPE [ ,FILABL = { NO
                                       STD } ] } ]

[,BLKSIZE=n][,RECSIZE=n] ,DEVADDR=SYSnnn
    
```

Inter-Directory Copy Utility

The Inter-Directory Copy utility copies information from one SUPRA Directory to another.

To execute this utility in OS/390 or VSE, submit sample JCL member TXJDRCPY.

Load Function, Version 2

File definitions

File	Description
CSI#WK nn (OS/390 only)	Identifies sort work files.
CSIPARM	Identifies the CSIPARM file, which contains control information needed by the PDM.
CSU#REC	Holds the number of records unloaded for each file.
CSUAUX	Holds the auxiliary information for the files not in native format. FILE Name of the file to which the parameters on this CSUAUX record apply RCYL Number of records per logical cylinder. CYLL Cylinder load limit—maximum percentage of each logical cylinder to be filled with data records during the load function. LOAD Format in which the file is to be loaded. <u>CSUAUX File Record Format</u> FILE=ffff RCYL=nnnn CYLL= $\left. \begin{array}{l} 80 \\ nnnn \end{array} \right\}$ LOAD= $\left[\begin{array}{l} \text{COMPATIBILITY} \\ \text{CONVERTED} \\ \text{NATIVE} \end{array} \right]$
INPUT (VSE only)	Holds the data for all files to be loaded.
LINKWK01 (OS/390) LNKWRK1 (VSE)	Indicates the first linkage work file.
LINKWK02 (OS/390) LNKWRK2 (VSE)	Indicates the second linkage work file.
SORTLIB	Specifies the library containing the standard sort program.
SORTWK n (VSE only)	Identifies work files for sorting.

File	Description
SYSIN (OS/390) SYSIPT (VSE)	Specifies the file containing the run control and file control statements, and so on.
SYSOUT	Identifies the file the standard sort program uses
SYSPRINT (OS/390) SYSLST (VSE)	Identifies the output file for the printed listing of all control statements, diagnostics, etc.
SYSUDUMP (OS/390 only)	Specifies the file to which a dump is written if an abend occurs.
SYSUT1 (OS/390 only)	Contains the data for all files to be loaded.
ffffff (OS/390) ffffff and Zffff (VSE)	Specifies the files to be loaded.

Run control statements

LINKWK nn statement

VSE only. Defines the characteristics of the LNKWRK1 and LNKWRK2 files.

$$\text{LINKWK } nn: \left[\text{DEVICE} = \left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \left[\text{,FILABL} = \left\{ \begin{array}{l} \text{NO} \\ \text{STD} \end{array} \right\} \right] \end{array} \right\} \right]$$

$$\left[\text{,BLKSIZE} = \left\{ \frac{1000}{n} \right\} \right]$$

$$\left[\text{,RECSIZE} = \left\{ \frac{1000}{n} \right\} \right]$$

$$\left[\text{,DEVADDR} = \left\{ \begin{array}{l} \text{SYS030} \\ \text{SYS } nnn \end{array} \right\} \right]$$

RECFORM statement

VSE only. Defines the record format of the INPUT file.

$$\left[\text{RECFORM} = \left\{ \begin{array}{l} \text{FIXBLK} \\ \text{FIXUNB} \end{array} \right\} \right]$$

$$\left[\text{,DEVICE} = \left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \left[\text{,FILABL} = \left\{ \begin{array}{l} \text{NO} \\ \text{STD} \end{array} \right\} \right] \end{array} \right\} \right]$$

$$\left[\text{,BLKSIZE} = \left\{ \frac{1000}{n} \right\} \right]$$

$$\left[\text{,RECSIZE} = \left\{ \frac{100}{n} \right\} \right]$$

$$\left[\text{,DEVADDR} = \left\{ \begin{array}{l} \text{SYS030} \\ \text{SYS } nnn \end{array} \right\} \right]$$

MAXKEY statement

Indicates the length of the longest primary file control key in the primary and related files to be loaded.

$$\text{MAXKEY} = \left\{ \begin{array}{l} 256 \\ n \end{array} \right\}$$

SCHEMA statement

Names the schema to be used for loading database files.

$$\text{SCHEMA} = \left\{ \begin{array}{l} \textit{schemaname} \\ \textit{bootschema} \end{array} \right\}$$

PRIMARY: statement

Identifies the primary files to be loaded.

PRIMARY:*pppp*₁[*pppp*₂...*pppp*_{*n*}]**END.**

or

S-E:*mmmm*₁[*mmmm*₂...*mmmm*_{*n*}]**END.**

SORTCORE statement

Identifies the number of bytes of virtual storage the sort program can use.

$$\text{SORTCORE} = \left\{ \begin{array}{l} 12000 \\ 24000 \\ n \end{array} \right\} \quad \begin{array}{l} \boxed{\text{OS/390}} \\ \boxed{\text{VSE}} \end{array}$$

SORTNAME statement

Identifies the sort program.

$$\text{SORTNAME} = \left\{ \begin{array}{l} \text{IERRCO00} \\ \text{SORT} \\ \text{progname} \end{array} \right\} \quad \begin{array}{l} \text{OS/390} \\ \text{VSE} \end{array}$$

RELATED: statement

Identifies the related files to be loaded.

RELATED:rrrr 1[rrrr 2...rrrr n]END.

or

V-E:vvvv 1[vvvv 2...vvvv n]END.

WORK statement

VSE only. Indicates the number of tape devices or disk extents available for intermediate storage while sorting.

$$\text{WORK} = \left\{ \begin{array}{l} 4 \\ n \end{array} \right\}$$

File control statements

LINKPATH statement

Specifies the name of the linkpath to be used as the primary linkpath for loading a related file.

rrrr LINKPATH = *pppp*LK *xx*

Element List statement

Specifies the data elements to be loaded for a particular file.

ffff { *element*₁ [*element*₂ ... *element*_{*n*}] } END.
ALL.

Unload Function, Version 2

File definitions

File	Description
CSI#WK <i>nn</i> (OS/390 only)	Identifies sort work files.
CSIPARM	Identifies the CSIPARM file, which contains control information that the PDM needs.
CSU#REC	Holds the number of records unloaded for each file.
CSUAUX	Holds the auxiliary information to define files not in native format. FILE Name of the file to which the parameters on this CSUAUX record apply. RCYL Number of records per logical cylinder. CYLL Cylinder load limit - maximum percentage of each logical cylinder to be filled with data records during the load function. LOAD Format in which the file is to be loaded. <u>CSUAUX File Record Format</u> $\text{FILE}=\text{ffff} \quad \text{RCYL}=\text{nnnn} \quad \text{CYLL}=\left\{ \begin{array}{l} 80 \\ \text{nnnn} \end{array} \right\}$ $\text{LOAD}=\left[\begin{array}{l} \text{COMPATIBILITY} \\ \text{CONVERTED} \\ \text{NATIVE} \end{array} \right]$
OUTFILE	Specifies the file where data from all unloaded files will be written.
PARM (OS/390 only)	Holds file control statements.
SORTLIB (OS/390 only)	Specifies the library holding the standard sort program.
SORTWK <i>n</i> (VSE only)	Specifies work files to be used for sorting.
SYSIN (OS/390 only)	Holds run control statements.
SYSIPT (VSE only)	Holds run control and file control statements.
SYSOUT (OS/390 only)	Specifies the file to be used by the standard sort program.
SYSPRINT (OS/390) SYSLST (VSE)	Specifies the output file for the printed listing of all control statements, diagnostics, etc.
SYSUDUMP (OS/390 only)	Specifies a dump file.
##### (OS/390) ##### and Z##### (VSE)	Used to define the file to be unloaded.

Run control statements

DUMP statement

Controls the printing of a storage dump when certain user errors occur.

DUMP= $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

NEW-SCHEMA/NEW-ENVDESC statement

Identifies the new schema and new environment description containing the file definitions to be used when the files are reloaded.

NEW-SCHEMA=*schemaname*,

NEW-ENVDESC=*envdescname*

PRIMARY: statement

Identifies the primary files to be unloaded.

PRIMARY:*pppp*₁[*pppp*₂...*pppp*_n]**END.**

or

S-E:*mmmm*₁[*mmmm*₂...*mmmm*_n]**END.**

RECFORM statement

VSE only. Indicates the format of the records in the OUTPUT file.

$$\left[\text{RECFORM} = \left\{ \begin{array}{l} \text{FIXBLK} \\ \text{FIXUNB} \end{array} \right\} \right]$$
$$\left[,\text{DEVICE} = \left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \left[,\text{FILABL} = \left\{ \begin{array}{l} \text{NO} \\ \text{STD} \end{array} \right\} \right] \end{array} \right\} \right]$$
$$\left[,\text{BLKSIZE} = \left\{ \begin{array}{l} 1000 \\ n \end{array} \right\} \right]$$
$$\left[,\text{RECSIZE} = \left\{ \begin{array}{l} 100 \\ n \end{array} \right\} \right]$$
$$\left[,\text{DEVADDR} = \left\{ \begin{array}{l} \text{SYS030} \\ \text{SYS}nnn \end{array} \right\} \right]$$

RELATED: statement

Identifies the related files to be unloaded.

RELATED:rrrr1[rrrr2...rrrrn]END.

or

V-E:vvvv1 [vvvv2...vvvvn]END.

SORTNAME statement

Identifies the sort program.

$$\text{SORTNAME} = \left\{ \begin{array}{l} \text{IERRCO00} \\ \text{SORT} \\ \text{progname} \end{array} \right\} \quad \begin{array}{|c|} \hline \text{OS/390} \\ \hline \text{VSE} \\ \hline \end{array}$$

TEST statement

Indicates whether the Unload function is to check and validate all Run Control statements without actually unloading the files.

$$\text{TEST} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$$

Work statement

VSE only. Indicates the number of tape drives or disk extents available for intermediate sort storage.

$$\text{WORK} = \left\{ \begin{array}{l} 1 \\ n \end{array} \right\}$$

File control statements

LINKPATH statement

Identifies the access linkpath to be used to unload a related file.

[rrrrLINKPATH=ppppLK xx]

[,PRESERVE={ $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$ }

[,RC=yy]

Element List statement

Specifies the data elements to be unloaded from a particular file.

ffff $\left\{ \begin{array}{l} \text{element}_1[*\text{FILL}=\text{nn}] \\ \text{element}_2[*\text{FILL}=\text{nn}...] \\ \text{ALL.} \end{array} \right\}$ **END.**

BLANK-LINKS statement

Specifies primary file linkpaths to be blanked during unloading of the primary file.

ppppBLANK-LINKS=LK xx₁[LK xx₂...LK xx_n]END.
