LEASY V6.2A

Ready Reference

Edition March 2007

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to manuals@fujitsu-siemens.com.

Certified documentation according to DIN EN ISO 9001:2000

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2000.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

1 Preface

LEASY (German acronym for linear input/output system) is a transaction-oriented data management and access system which is run under BS2000.

LEASY supports

- simple and uniform access to DMS files
- secondary keys
- transactions
- data security

Files can be accessed from COBOL, Assembler or RPG programs. The interface complies with KLDS, the standard for compatible interfaces to linear database systems.

LEASY can be used in timesharing mode (Batch/TIAM) and in transaction mode (UTM, DCAM)

The **LEASY Ready Reference** is intended to provide the user quickly with essential information on the syntax of the commands and macros required for LEASY application.

Notational conventions

The following conventions have been employed in the manual for the formal representation of the statements and their operands:

Formal representation	Explanation	Examples
UPPERCASE LETTERS and special characters	Uppercase letters and special characters indicate constants which must be entered by the user in exactly this form	*CAT file catalog
lowercase letters	Lowercase letters indicate variables which the user must replace by current values.	The user enters: *CAT TESTCAT
$\left\{ \quad \right\}$	Braces enclose alternatives, i.e. one of the specifications must be selected.	$\left\{\begin{array}{l}\text{file}\\ \text{file.suffix}\\ \text{file.}\end{array}\right\}$ The user enters: FILE1 or FILE1.Z1 or FILE1.
[ ]	Square brackets enclose optional specifications.	keyname[,iub] The user enters: KEY1 oder KEY1,X' 00'
...	Dots indicate a repetition; the preceding syntactical unit can be repeated several times in succession.	(pos,len),... The user enters: (12,4) or (12,4),(14,10),(25,2)
_	Underlining indicates the default value. This is the value set by the utility routine if no specification is made by the user.	$\text{INF=}\left\{\begin{array}{c}\text{Y}\\ \underline{\text{N}}\end{array}\right\}$ The user enters: INF=Y or INF=N or nothing (i.e. same as INF=N)

Table 1: Notational conventions

2 Locking strategy

The following table illustrates how various actions affect the lock log.

<table>
<tr><th>Action</th><th colspan="2">Meaning</th></tr>
<tr><td>LOCK, RHLD,
RNHD,RPHD</td><td colspan="2">Individual records of an ISAM, DAM or PAM file can be locked (explicit locking).</td></tr>
<tr><td>LOCK, RHLD</td><td colspan="2">Individual record ranges of an ISAM, DAM or PAM file can be locked.</td></tr>
<tr><td>LOCK</td><td colspan="2">The <i>LOCK</i> operation also serves to lock records/record ranges that do not (yet) exist (so-called phantoms).</td></tr>
<tr><td>INSR, STOR</td><td colspan="2">Inserted records are automatically locked (implicit locking).</td></tr>
<tr><td>INSR, STOR,
REWR, DLET</td><td colspan="2">Locks on updated, inserted or deleted records are automatically retained until the end of the transaction and cannot be canceled by <i>UNLK</i>.
If the record is contained in a lock range, LEASY generates an additional lock element for this record. Although the range can then be released by means of <i>UNLK</i>, the additional record locks are retained until the end of the transaction.</td></tr>
<tr><td>DLET, REWR</td><td colspan="2">Records to be deleted or updated must first be locked (implicitly or explicitly).</td></tr>
<tr><td>UNLK</td><td colspan="2">Locked but not updated records/record ranges are released and, if <i>OPE1=‘U‘</i> is specified, updated as well.</td></tr>
<tr><td>CLTR</td><td colspan="2">All locks are canceled automatically at the end of the transaction (except for foreign files).</td></tr>
<tr><td>“unprotected read”</td><td colspan="2">Locked and updated records can be read by other transactions ((<i>RDIR, RNXT, RPRI</i>).
This so-called "unprotected read" is authorized to permit a higher degree of parallel transaction processing.</td></tr>
<tr><td>Initialization of the main LEASY task with the operand <i>*TIME</i></td><td>The timeout for release of records locked by parallel transactions can be globally specified with this operand.</td><td rowspan="2">If timeout occurs without success, a return code informs the user. The locking attempt is repeated at one-second intervals.</td></tr>
<tr><td>OPE-WTIME field in the <i>RE</i> area</td><td>Timeouts for the release of locked records can be specified for each operation.</td></tr>
</table>

Table 2: Effects of different actions on the lock log

3 Secondary keys

The following table shows the main differences between LEASY and ISAM secondary keys

Criterion	**LEASY SK**	**ISAM SK**
File type	Master files	Master and foreign files
Storage of index	Own SI file	Primary file
Access method	DAM, PAM, ISAM	NK-ISAM
Number of SKs	255	30
Length of SK	PK + SK < 255	< 128
Multiple keys	YES	NO
Suppression	YES	NO
Sequence for identical SKs	Primary key	Time of entry
Definition	Utility routine LEASY-CATALOG	CREATE-ALTERNATE-INDEX command CREAIX macro
Display	Utility routine LEASY-CATALOG	Utility routine LEASY-CATALOG, SHOW-INDEX-ATTRIBUTES command, SHOWAIX macro
Manual setup	Utility routine LEASY-LOADSI	CREATE-ALTERNATE-INDEX command CREAIX macro
Automatic setup via LEASY runtime system	Controlled via LEASY-CATALOG	For each defined SK

Table 3: Differences between LEASY and ISAM secondary keys

4 Job variables

LEASY offers the following job variables for monitoring central resources:

*LEACMST	state of common memory
*LEAIOST	number of active I/O tasks

Before a job variable can be used, the following measures are required.

1. Catalog a job variable by means of a *CREATE-JV* command.

   ```
   /CREATE-JV jvname
   ```

   The job variable name (*jvname*) can be selected by the user.

2. Assign the link name (*LEACMST* or *LEAIOST*) of the job variable by means of the *SET-JV-LINK* command.

   ```
   /SET-JV-LINK LINK=LEAxxx,JV-NAME=jvname
   ```

The following table shows the values which LEASY assignes to the job variables.

Link name	Bytes	Contents	Meaning
*LEACMST	1-10	INIT NORMAL NOT ACTIVE	State of common memory
	11-20	IN ACTION IN ERROR END Blanks	State of AIM switching

	21-30	END ALL
		END VALID
		READY
			State of reconstruction

	31-40	ACTIVE FINISHED WAITING Blanks	State of PETR handling
	41-50	ACTIVE ERROR READY END	State of ROMS
*LEAIOST	1-10	*nnn*-ACTIVE	Number of active I/O tasks

Table 4: Job variables

5 LEASY interface

5.1 List of LEASY operands

Table 5 shows all the possible LEASY operands and their respective positions in the LEASY call.

Position	Name	Meaning	Type
1	OP	Operation code	U
2	RE	Reference area	U/R
3	DB	File allocation	U
	CI	Currency information	U/R
	CAT	Catalog information	U
4	AR	Input/output area	U/R
5	FA	Field selection	U
6	SI	Secondary index	U
7	KB	Key begin	U
8	KE	Key end	U
last	US	User area	U

Table 5: Overview of the LEASY operands

Key

U Information supplied by the user program to LEASY

R Information returned from LEASY to the user program

The operands listed under “3” (*DB/CI/CAT*) are alternatives.

Operation code OP

Table 6 lists the permissible LEASY operation.

LEASY operation	**Meaning**
CATD	Call LEASY catlog
OPFL	Open files
CLFL	Close files
OPTR	Open or extend transaction
CLTR	End transaction
MARK	Create checkpoint
BACK	Execute rollback
RDIR	Directly read record
RNXT	Read next record
RPRI	Read previous record
RHLD	Directly read and lock record
RNHD	Read and lock next record
RPHD	Read and lock previous record
SETL	Position file pointer
INSR	Insert new record
STOR	Insert record
REWR	Rewrite record
DLET	Delete record
LOCK	Lock record
UNLK	Unlock record
CINF	Transfer currency information

Table 6: LEASY operations

Reference area RE

Table 7 shows the structure of the reference area.

Field names (graduated)	Position (bytes)	Length	Type		Meaning	
RC-CC	1-3	3	A	R	Compatible return code	Compatible part of reference area to KLDS
RC-KZ	4	1	A	R	System identifier "L"	
RC-LC	5-8	4	A	R	LEASY return code	
PASS	9-16	8	A	-	Reserved for password	
OPE OPE-STX OPE-OM OPE-LOG -----	17-24 17 18 19 20-24	8 1 1 1 5	A A A A -	U U U U -	Operation extensions STXIT mode OPEN/USAGE mode BIM logging control Reserved	
INT SAMPTR PAMHPNR ----- SAMPTR	25-32 25-28 25-28 29-32 25-32	8 4 4 4 8	A A B A A	U/R U/R U/R - U/R	Internal key aspect SAM retrieval address (24-bit) or PAM block number Reserved SAM retrieval address (32-bit)	
NUM	33-40	8	N	R	Number of primary records	
IDE	41-48	8	A	U/R	Identification field for DCAM application	
REOP	49-52	4	A	R	Last operation code	LEASY extension of RE
REDB	53-68	16	A	R	Last file name (+ SI name)	
L-OPT	69	1	A	U	Version identifier "1"	
OPE1 OPE2	70 71	1 1	A A	U U	Operation extensions for OPTR/CLTR/RDIR/RHLD/RNHD /RPHD/LOCK/CINF	
OPE-WTIME	72-74	3	N		Waiting time for locks	
RC-LCE	75-79	5	A	R	LEASY return code extension	
U-PROT	80	1	A	U	User information	

Table 7: Structure of the reference area RE

A alphanumeric field

B numeric field (binary)

N numeric field (printable)

U information supplied by the user program to LEASY

R information returned by LEASY to the user program

Transfer and return in individual fields

The following table shows the transfer and return information in the individual fields of the reference area *RE*

Field	Type	Contents
RC-CC	R	Compatible return code from KLDS.
RC-KZ	R	LEASY identifier "L".
RC-LC	R	Error code internally generated by LEASY. This error code is more detailed than the compatible return code. The 4 bytes of RC-LC may be given the following format: A ddd AIM file B ddd BIM file C ddd DMS error in catalog file D ddd primary file J ddd processing job file(JV) S ddd secondary index file T ddd status file ddd For three-digit DMS message numbers, these are the rightmost 3 bytes of the DMS error code, which has the format 0ddd For four-digit DMS message numbers (first digit not 0), these are the string "DMS". The *RC-LCE* field then contains the 4-digit DMS message number. L eee LEASY-internal error code. An additional error code can be provided in the *RC-LCE* field as supplementary information. The compatible return codes together with the return information generated by LEASY are listed in detail with their meanings in the chapter “Return codes” on page 89ff.
PASS		Reserved
OPE-STX	U	Entries in the OPE-STX field are ignored as of LEASY V6.1, the STXIT routine in LEASY remains activated in any case

Table 8: Transfer and return in the fields of the RE area (part 1 of 7)

<table>
<tr><th>Field</th><th>Type</th><th>Contents</th></tr>
<tr><td>OPE-OM</td><td>U</td><td>An identifier indicating the method of opening files or file identifiers can be specified in the OPE-OM field for the OPFL and OPTR operations.
OPE-OM= X'FF' means for <b>both operations</b> that the DB4 format is selected in the 3rd operand of the LEASY call for the file allocation and that the associated OPEN mode is specified in the DB operand for each file.
n the <i>OPFL</i> operation the 1-byte OPEN mode can be specified in the <i>OPE-OM</i> field; this mode is then valid in the same way for all those files allocated with <i>DB1/DB2</i> format.
In the case of the OPTR operation it is possible to specify not only X' FF' in this field, but also a 1-byte long processing mode, which is then valid in the same way for all file identifiers that are allocated with DB1 or DB2 format. This processing mode is mapped to a particular LEASY USAGE mode according to the table below:
<table>
<tr><th>Processing mode</th><th>USAGE mode</th></tr>
<tr><td>_ (default value)</td><td>EXLD (SAM)
UPDT(ISAM/PAM/DAM)</td></tr>
<tr><td>A</td><td>RETR</td></tr>
<tr><td>E</td><td>PRUP</td></tr>
<tr><td>G</td><td>EXRT</td></tr>
<tr><td>L</td><td>EXLD</td></tr>
<tr><td>I</td><td>PRRT</td></tr>
<tr><td>O</td><td>EXLD</td></tr>
<tr><td>Q</td><td>EXLD</td></tr>
<tr><td>X</td><td>EXRT</td></tr>
<tr><td>B</td><td>EXUP</td></tr>
<tr><td>R</td><td>ULRT</td></tr>
<tr><td>U</td><td>ULUP</td></tr>
</table>
The specification of a processing mode has the same effect as the specification of the assigned USAGE mode for all declared file identifiers by means of the DB4 format.</td></tr>
</table>

Table 8: Transfer and return in the fields of the RE area (part 2 of 7)

Field	Type	Contents
OPE-LOG	U	In the 1st *OPTR* operation of a transaction the BIM save method for this transaction can be canceled by specifying "N". The field is space-filled (X' 40') as standard, i.e. BIM saving is activated for the current transaction if the appropriate operand values are assigned in the LEASY-MAINTASK and LEASY-CATALOG utility routines. If openUTM and LEASY are linked, BIM saving may only be deactivated for read transactions.
SAMPTR	U/R	In the case of SAM files, the current retrieval address is returned in the *SAMPTR* field after each operation. This is specified in the format (24-bit or 31-bit) predefined with the *SETL* or *RDIR* operations (*ID1RPTR*=’bbbbbbrr’ or *ID1BLK#*=’bbbbbbbb’ and *ID1REC#*=’rrrrrrrr’). Unless defined otherwise, 24-bit format is used. With the *SETL* or *RDIR* operation, such a retrieval address must be stored in the *SAMPTR* field in either 24-bit (’bbbbbbrr’) or 31-bit format (’bbbbbbbbrrrrrrrr’). This allows for positioning within the file for a subsequent sequential read operation. With *RNXT/RNHD*, a switchover is made from 24 bit mode to 31 bit mode if the number of the record being read in the block exceeds 255. The 31 bit mode remains activated until it is reset back to 24 bit mode possibly by either *SETL* or *RDIR*. If the 24 bit format is used, the second word of the *SAMPTR* field must then be filled with zeros or blanks.
PAMHPNR	U/R	The PAM block number must be stored in this field in PAM write operations and for direct reading; in sequential read operations and read operations via secondary keys this is done by LEASY.
NUM	R	LEASY supplies the number of primary records belonging to a secondary index value in the *NUM* field for *RDIR/RHLD* operations. This is only possible if the identifier "*N*" is specified in the *OPE2* field, and if no range has been specified for access via a secondary index.
IDE	U/R	No entry is made to this field unless LEASY is called by a DCAM application. The DCAM application name must be supplied in the *IDE* field for the *CATD* operation. *IDE* must be erased prior to the 1st *OPTR* operation of each transaction. LEASY will then return the transaction identifier with the *OPTR* operation. This identifier must be supplied for all LEASY operations affecting this transaction. A *CLTR* operation causes the *IDE* field to be erased.

Table 8: Transfer and return in the fields of the RE area (part 3 of 7)

Field	Type	Contents			
REOP/ REDB	R	LEASY always enters the operation code and the file name (+ SI name) of the last call in the *REOP* and *REBD* fields. If an error occurs during the *OPFL* (open files) or the *OPTR* (open transaction) operations, the file causing the error (together with its OPEN or USAGE mode) is stored in the *REDB* field. In the *CATD* operation (call LEASY catalog) the first 16 bytes of the specified catalog name are stored in the *REDB* field. This allows the user to employ a common error routine when handling errors.			
L-OPT	U	LEASY interface identifier. This field must always be set to "*1*".			
OPE1/OPE2	U	Additional functions can be specified in the OPE1 and OPE2 fields for the following operations:			
		OP=OPTR:	OPE1=	'_'	normal transaction start (*DB* specification)
			OPE1=	'W'	transaction start and simultaneous file positioning (*CI* specification in 3rd operand)
		OP=CLTR:	OPE1=	'_'	normal end of transaction
			OPE1=	'R'	resetting of transaction
			OPE2=	'_'	transaction termination with cancellation of all file access requests
			OPE2=	'T’	transaction termination and simultaneous transaction start (restart point with release of record locks but retention of resources and file positions)
		OP=RDIR/RHLD:			
			OPE2=	'N'	LEASY must transfer the number of primary records to a secondary index value (in the *NUM* field).

Table 8: Transfer and return in the fields of the RE area (part 4 of 7)

Field	Type	Contents			
OPE1/OPE2 (continued)	U	OP=RHLD/RNHD/RPHD/LOCK:			
			OPE1=	'S'	READ-LOCK enforced on locking
			OPE1=	'_'	WRITE-LOCK enforced on locking
		OP=RNHD/RPHD:			
			OPE2=	L	If the required record is free, it is transferred to the *AR* area and locked. The pointer is positioned after or before the record that has been read, depending on the direction in which it was read. If the record is locked, LEASY sets the pointer in the same way as if it had been read.
			OPE2=	_	If the required record is free, it is transferred to the *AR* area and locked. The pointer is positioned after or before the record that has been read, depending on the direction in which it was read. If the record is locked, the return code (*99ALL006*) is transferred after the waiting time has elapsed. The record is not read and the pointer is not modified.
		OP=CINF:	OPE1=	'_'	Currency information on all the file identifiers open in the transaction and also their current file pointers. This currency information enables a transaction to be opened (with simultaneous file positioning) by means of *OPTR* and *OPE1=' W'* .
			OPE1=	'F'	Currency information on the files contained in the LEASY catalog and their secondary indices.

Table 8: Transfer and return in the fields of the RE area (part 5 of 7)

Field	Type	Contents			
OPE1/OPE2 (continued)	U		OPE2=	'_' 'C'	Currency information (type 1) on all the files in the LEASY catalog.
			OPE2=	'O'	Currency information (type 1) on all the files opened by means of *OPFL*.
			OPE2=	'T'	Currency information (type 1) on all the files involved in the transaction.
			OPE2=	'S'	Currency information (type 2) on the file specified in CI.
			OPE2=	'W'	The help function immediately preceding this field is to be continued.
			– The OPE2 entry is only practical if *OPE1='F'* is also specified. – Type 1 currency information only includes general information on the file. Type 2 currency information lists all the tables for the specified file which are for use within LEASY.		
OPE1	U	OP=UNLK	OPE1=	'_'	Normal record release
			OPE1=	'U'	In transactions without BIM saving, modified records are also released
OPE-WTIME	U	A waiting time in seconds for locked records or logical files can be specified individually for each operation in the *OPE-WTIME* field. If the field is not occupied *(X'40'* or *X'00'*), the global waiting time for the session applies (_*TIME_ operand in the LEASY-MAINTASK utility routine); the default value is 0 if there is no LEASY catalog. Even if an *OPTR* operation encounters a USAGE mode incompatibility with a parallel transaction for a file identifier of the file list specified, the specified or the global waiting time comes into force. If this waiting time expires without the locking transaction having been completed, the user program receives the return code *99ALL110*; otherwise it can continue within its *OPTR* operation.			

Table 8: Transfer and return in the fields of the RE area (part 6 of 7)

Field	Type	Contents
RC-LCE	R	The 5 bytes of *RC-LCE* can have the following format:
		1. 4-character message code for a DMS error in one of the following forms:
		Axxxx DMS error while processing an AIM file Bxxxx DMS error while processing a BIM file Cxxxx DMS error while processing a catalog file Dxxxx DMS error while processing a primary file Jxxxx DMS error while processing a job variable Sxxxx DMS error while processing a secondary index file Txxxx DMS error while processing a status file
		xxxx 4-digit DMS message number (see the "System Messages" manual)
		2. Error code extension for the internal LEASY error code stored in the *RC-LC* field in the following form:
		L_eee
		eee LEASY-internal error code
		3. NKISAM macro error code for the NKISAM macro error stored in the *RC-LC* field, in the form
		Iiiii
		iiii Main return code of NKISAM macro (see the "Introductory Guide to DMS").
		4. Other macro code for the macro error stored in the *RC-LC* field, in the form
		Mbbaa
		bbaa corresponds to the return code of the relevant macro in R15 (R15=X' bb0000aa')
U-PROT	U	When user information is specified, the value ' Y' must be set in this field in the case of the operations *BACK, CATD, CLFL, CLTR, DLET, INSR, OPFL, OPTR, REWR* and *STOR*. In the case of the other operations, this field is not evaluated.
		U-PROT= ' ' No user information specified.
		U-PROT= 'Y' User information specified. The last operand in the operand list is interpreted as user information.

Table 8: Transfer and return in the fields of the RE area (part 7 of 7)

File allocation DB

A file identifier consists of the logical name of the file, which can be up to 8 positions long, and an identification code for a sequence identifier (*fm*), which can be up to 3 positions long (optional). The file name and sequence identifier are separated from one another by a slash (/).

```
file-identifier : file[/fm]
```

Format DB1

Format for OPFL

```
file
```

file	logical file name (max. 8 characters)

Format for OPTR and all read and write operations

```
file[/fm]
```

file/fm	file identifier	
	file	logical file name (max. 8 characters)
	fm	sequence identifier (max. 3 characters)

Format DB2

Format for OPFL

```
(file1,file2,...)
```

file	logical file names (max. 8 characters)

Format for OPTR

```
(file1[/fm1],file2[/fm2],...)
```

file/fm	file identifiers	
	file	logical file names (max. 8 characters)
	fm	sequence identifier (max. 3 characters)

Blanks must not be entered in the parenthesized expression.

Format DB3

This format may only be used for *CLFL* and *UNLK* operations. *ALL* addresses **all** allocated files.

```
⎧(ALL)⎫
⎩ALL  ⎭
```

If *ALL* is specified without parentheses, the field must be 12 bytes in length.

Format DB4

Formats for OPFL

for *one* file	for *several* files
(file,mode)	((file1,mode1),(file2,mode2)...)

file	logical file name (max. 8 characters)
mode	OPEN mode (1 character)

Formats for OPTR

for *one* file identifier	for *several* file identifiers
(file[/fm],mode)	((file1[/fm1],mode1),(file2[/fm2],mode2)...)

file/fm file identifier

file/fm	file identifier	
	file	logical file name (max. 8 characters)
	fm	sequence identifiers (max. 3 characters)
mode	USAGE mode (4 characters)	

i Blanks must not be entered in the parenthesized expression.

Currency information CI

Format of the currency information CI

The CI takes the form of a variable-length record with a 4-byte length field at the beginning.

CI	ci-slf	ci-slr	ci-inf	
	1	3	5	n+4 bytes

Field name	Position (bytes)	Length	Type	Meaning
ci-slf	1 - 2	2	U/R	Length field; contains the value n+4
ci-slr	3 - 4	2	R	Length field; contains the necessary minimum length of CI
ci-inf	5 to n+4	n	R	Information field with length n

Return values

<table>
<tr><td rowspan="6"></td><td colspan="6">Is ci-inf large enough?</td></tr>
<tr><td colspan="2">yes</td><td colspan="4">no</td></tr>
<tr><td colspan="2">File available?</td><td colspan="4">len > 64K?</td></tr>
<tr><td rowspan="3">yes</td><td rowspan="3">no</td><td colspan="2">no</td><td colspan="2">yes</td></tr>
<tr><td colspan="2">Space for at least one file?</td><td colspan="2">Space for at least one file?</td></tr>
<tr><td>yes</td><td>no</td><td>yes</td><td>no</td></tr>
<tr><td>Error message</td><td>-</td><td>-</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>ci-slf</td><td>len</td><td>0</td><td>partlen</td><td>0</td><td>partlen</td><td>0</td></tr>
<tr><td>ci-slr</td><td>-</td><td>-</td><td>len</td><td>len</td><td>X' FFFF'</td><td>X' FFFF'</td></tr>
<tr><td>ci-inf</td><td>inf</td><td>-</td><td>partinf</td><td>-</td><td>partinf</td><td>-</td></tr>
</table>

len	length of all the file information
inf	all the file information
partlen	length of the transferred information section
partinf	part of the file information

Calculating the length of ci-slf

The length of *ci-slf* is calculated as follows:

For OPE1=_

$$ci\text{-}slf = 4 + n*16 + n_1*5 + \sum_{i=1}^{n_2}(KEYLEN_i + 1) + n_3*8 + \sum_{i=1}^{n_4}2*KEYLENINT_i$$

n number of file identifiers. n = n1 + n2

n_1 number of file identifiers of SAM files

n_2 number of file identifiers of ISAM, PAM and DAM files

$n_3 \leq n_1$
$n_4 \leq n_2$ number of file identifiers with current range limits (KB, KE)

$KEYLEN_i$ max (*KEYLEN-PRIMFILE,KEYLEN-SIFILE*) of the (i)th file identifier

$KEYLENINT_i$ *KEYLEN-PRIMFILE* or *KEYLEN-SIFILE* of the (i)th file identifier for which the range limits apply.

KEYLEN-PRIMFILE=3 is mandatory for PAM files

KEYLEN-PRIMFILE=4 is mandatory for DAM files

For OPE1=F and OPE2=C, O, T or _

$$ci\text{-}slf = 4 + n*88 + v$$

n number of files

v 16 or 0
space for internal LEASY administrative information if only a section of the file information is to be retrieved and additional sections are to be requested with the aid of *CINF* and *OPE2=W*. The value *v=16* should be used in this case.

For OPE1=F and OPE2=S

$$\boxed{\text{ci-slf} = 4 + 111 + s*22 + \sum_{j=1}^{s}(st_j*5 + \sum_{k=1}^{r_j}(rid_{jk}+1))}$$

rounded up to a multiple of 4

s number of secondary index definitions in the file

st_j number of code sections of secondary index definition j

r_j number of record type definitions of secondary index definition j

rid_{jk} length of record type definition k in secondary index definition j

Catalog information CAT

This operand must be specified in the *CATD* operation.

CAT	catname	suffix	
	1	25	44 bytes

Field name	Position (bytes)	Length	Type	Meaning
catname	1 - 24	24	U	Name of LEASY catalog
suffix	25 - 44	20	U	Suffix for model files

Input/output area AR

The operand *AR* refers to a transfer or return area. This area has a variable length.

AR	reczone	
	1	n bytes

In DAM files the *AR* area has the following format:

AR	recno	reczone
	1	5 n+4 bytes

Field name	Position (bytes)	Length	Type	Meaning
recno	1 - 4	4	U/R	Relative record number (binary)
reczone	5 to n+4	n	U/R	Record zone with length n

Field selection FA

FA	string
	1 8 bytes

Secondary index SI

SI	secondary-index-name
	1 8 bytes

Key begin KB and key end KE

KB	key-value	KE	key-value
	1 m bytes		1 m bytes

m key length

User area US

US	us-sl	filler	us-inf	
	1	3	5	n+4

Field name	Position (bytes)	Length	Type	Meaning
us-sl	1 - 2	2	U	Length of USER information (=n+4)
filler	3 - 4	2	U	Filler characters
us-inf	5 to n+4	n	U	USER information of variable length

5.2 LEASY operations

Overview of LEASY operations and permissible operands

Operation	RE	OPE1	OPE2	INT	DB1/2/3/4/CI/CAT	AR	FA	SI	KB	KE	US
OPFL	x				DB1/DB2/DB4						[x]
OPTR	x	x			DB1/DB2/DB4/CI						[x]
CLFL	x				[DB1/DB2/DB3]						[x]
CLTR	x	x	x								[x]
MARK	x										[x]
BACK	x										[x]
RDIR	x		x	P+S	DB1	x	[x	[x	[x	[x]]]]	
RHLD	x	x	x	P+S	DB1	x	[x	[x	[x	[x]]]]	
SETL	x			P+S	DB1	[I+D	[x	x	[x	[x]]]]	
LOCK	x	x		P	DB1	[I+D	[I+P+D	I+P+D	I+P+D	[I+P+D]]]	
RNXT	x				DB1	x	[x]				
RNHD	x	x			DB1	x	[x]				
RPRI	x				DB1	x	[x]				
RPHD	x	x			DB1	x	[x]				
INSR	x			P	DB1	x					[x]
STOR	x			P	DB1	x					[x]
REWR	x			P	DB1	x					[x]
DLET	x			P	DB1	[I+D	[I+P+D	I+P+D	I+P+D]]		[x]
UNLK	x	x		P	[DB3]/[DB1	[I+D	[I+P+D	I+P+D	I+P+D	[I+P+D]]]]	
CINF	x	x	x		CI						
CATD	x				CAT						[x]

Table 9: LEASY operations and their operands

x	Operands are mandatory.
I	Operands are mandatory for ISAM.
P	Operands are mandatory for PAM.
D	Operands are mandatory for DAM.
S	Operands are mandatory for SAM.
[ ]	Operands are optional.
/	One of the listed operands must be specified.

BACK Execute rollback

Operands in the LEASY call:

```
OP,RE[,US]
```

CATD Call LEASY catalog

Operands in the LEASY call:

```
OP,RE,CAT[,US]
```

CINF Transfer currency information

Operands in the LEASY call:

```
OP,RE,CI
```

CLFL Close files

Operands in the LEASY call:

$$\text{OP,RE,}\left\{\begin{array}{l}\text{DB1}\\ \text{DB2}\\ \text{DB4}\end{array}\right\}\text{[,US]}$$

CLTR Close transaction

Operands in the LEASY call:

```
OP,RE[,US]
```

DLET Delete record

Operands in the LEASY call:

```
OP,RE,DB1[,AR[,FA,SI,KB]][,US]
```

The table below shows the various methods of transferring the key values as a function of the file type and the number of operands in the LEASY call.

File type	**No. of operands**	**Supplied:**		**Deleted:**
ISAM, DAM	7	DB1 KB	with file name with primary key	Record with the primary key from *KB*
	4	DB1 AR	with file name with the primary key at the defined position for ISAM; in the first 4 bytes for DAM	Record with the primary key from the *AR* area
	3	DB1 AR	with file name not supplied	Last record read successfully via the same file identifier
PAM	7	DB1 KB	with file name with PAM block number	Block with the PAM block number from *KB*
	3	DB1 PAMHPNR (reference area *RE*)	with file name with PAM block number	Block with the PAM block number from *PAMHPNR*. With *PAMHPNR=0:* last block read successfully via the same file identifier

Table 10: Transfer of key values for the DLET operation

INSR Insert new record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,US]
```

LOCK Set record lock

Operands in the LEASY call:

```
OP,RE,DB1[,AR[,FA,SI,KB[,KE]]]
```

The table below shows the various methods of transferring the primary key values as a function of the file type and the number of operands in the LEASY call.

File type	**No. of operands**	**Supplied:**		**Deleted:**
ISAM, DAM	8	DB1	with file name	File section delimited by the primary keys *KB/KE*
		SI	with blanks or “MAINITEM”	
		KB, KE	Primary keys of range limits. The contents of *KB* may be greater than, less than or equal to those of *KE*	
	7	DB1	with file name	Record with the primary key from *KB*
		SI	with blanks or “MAINITEM”	
		KB	with primary key	
	4	DB1	with file name	Record with the primary key from the *AR* area
		AR	with the primary key at the defined position for ISAM; in the first 4 bytes for DAM	
PAM	8	DB1	with file name	File section delimited by the PAM block numbers *KB/KE*
		SI	with blanks or “MAINITEM”	
		KB, KE	PAM block numbers of the range limits	
	7	DB1	with file name	Block with the PAM block number from *KB*
		SI	with blanks or “MAINITEM”	
		KB	PAM block number	
	3	DB1	with file name	Block with the PAM block number from *PAMHPNR*
		PAMHPNR (reference area *RE*) with PAM block number		

Table 11: Transfer of key values for the LOCK operation

MARK Create checkpoint

Operands in the LEASY call:

```
OP,RE[,US]
```

OPFL Open files

Operands in the LEASY call:

```
       ⎧DB1 ⎫
OP,RE,⎨DB2 ⎬[,US]
       ⎩DB4 ⎭
```

OPTR Open or extend transaction

1) Defining the start of a transaction or extending a transaction-oriented file list

Operands in the LEASY call:

```
       ⎧DB1 ⎫
OP,RE,⎨DB2 ⎬[,US]
       ⎩DB4 ⎭
```

2) Opening a transaction, and opening and positioning file identifiers in accordance with CI

Operands in the LEASY call:

```
OP,RE(mit OPE1=W),CI[,US]
```

RDIR/RHLD Directly read record / Directly read and lock record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,FA[,SI[,KB[,KE]]]]
```

The table below shows the various methods of transferring the key values as a function of the file type and the number of operands.

File type	**No. of operands**	**Supplied:**	**Returned by LEASY:**
ISAM PAM, DAM, SAM	8	DB1 with file name FA with $\left\{\begin{array}{l}\text{ALL}\\\text{MAINITEM}\end{array}\right\}$ SI with $\left\{\begin{array}{l}\text{Secondary index}\\\text{Blanks}\\\text{MAINITEM}\end{array}\right\}$ $\left.\begin{array}{l}\text{KB}\\\text{KE}\end{array}\right\}$ Range limit keys (primary or secondary key)	FA: ALL If KB<KE: record with the lowest key value in the range in the *AR* area If KB> KE: record with the highest key value in the range in the *AR* area If KB=KE: record with the key value *KB* in the *AR* area FA: MAINITEM If KB<KE: lowest key in the range in the *AR* area or *RE* area (PAM, SAM) If KB> KE: highest key in the range in the *AR* area or *RE* area (PAM, SAM) If KB=KE: key from *KB* in the *AR* area or RE area (PAM, SAM)
	7	DB1 with file name FA with $\left\{\begin{array}{l}\text{ALL}\\\text{MAINITEM}\end{array}\right\}$ SI with $\left\{\begin{array}{l}\text{Secondary index}\\\text{Blanks}\\\text{MAINITEM}\end{array}\right\}$ KB with key (primary or secondary key)	FA: ALL Record with the key from *KB* in the *AR* area FA: MAINITEM Key from *KB* in the *AR* area or *RE* area (PAM, SAM)

Table 12: Transfer of key values for the RDIR/RHLD operation (part 1 of 2)

File type	**No. of operands**	**Supplied:**	**Returned by LEASY:**
ISAM PAM, DAM, SAM	6	DB1 with file name AR with primary key: with ISAM at the defined position; with DAM in the first 4 bytes (binary) PAMHPNR (RE area) with PAM: FA with $\left\{\begin{array}{l}\text{ALL}\\ \text{MAINITEM}\end{array}\right\}$ SI with $\left\{\begin{array}{l}\text{Secondary index}\\ \text{Blanks}\\ \text{MAINITEM}\end{array}\right\}$	FA: ALL With ISAM and DAM: record with the primary key specified in the *AR* area in the *AR* area With PAM: block with the primary key specified in *PAMHPNR* in the *AR* area FA: MAINITEM With ISAM and DAM: primary key in the *AR* area With PAM: primary key from *PAMHPNR* in the *RE* area
	5	DB1 with file name AR with primary key: with ISAM at the defined position; with DAM in the first 4 bytes (binary) PAMHPNR (RE area) with PAM: FA with $\left\{\begin{array}{l}\text{ALL}\\ \text{MAINITEM}\end{array}\right\}$	FA: ALL With ISAM and DAM: record with the primary key specified in the *AR* area in the AR area With PAM: block with the primary key specified in *PAMHPNR* in the *AR* area FA: MAINTEM With ISAM and DAM: primary key in the AR area With PAM: primary key from *PAMHPNR* in the RE area
	4	DB1 with file name AR with primary key: with ISAM at the defined position; with DAM in the first 4 bytes (binary) PAMHPNR (RE area) with PAM: primary key SAMPTR (RE area) with SAM: primary key	With ISAM and DAM: record with the primary key specified in the *AR* area in the *AR* area With PAM: block with the primary key specified in *PAMHPNR* in the *AR* area With SAM: record with the retrieval address specified in *SAMPTR*

Table 12: Transfer of key values for the RDIR/RHLD operation (part 2 of 2)

REWR Rewrite record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,US]
```

RNXT/RNHD Read next record / Read and lock next record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,FA]
```

RPRI/RPHD Read previous record / Read and lock previous record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,FA]
```

SETL Position file pointer

Operands in the LEASY call:

```
OP,RE,DB1[,AR[,FA,SI[,KB[,KE]]]]
```

STOR Insert record

Operands in the LEASY call:

```
OP,RE,DB1,AR[,US]
```

UNLK Cancel record lock

Operands in the LEASY call:

```
       ⎧{DB1[,AR[,FA,SI,KB[,K ⎫
OP,RE[,⎨                      ⎬
       ⎩E]]]                  ⎭
```

The table below shows the various methods of transferring key values in accordance with the file type and the number of operands.

File type	**No. of operands**	**Supplied:**		**Deleted:**
ISAM, DAM	8	DB1 SI KB KE	with file name with blanks or “MAINITEM” Primary keys of range limits The contents of *KB* may be greater than, less than or equal to the contents of *KE*	File section delimited by the primary keys *KB/KE*
	7	DB1 SI KB	with file name with blanks or “MAINITEM” with primary key	Record with the primary key from *KB*
	4	DB1 AR	with file name with primary key: at the defined position for ISAM; in the first 4 bytes for DAM	Record with the primary key from the *AR* area
	3	DB1 AR	with file name no specification	All locked (but unmodified) records in this file

Table 13: Transfer of key values for the UNLK operation (part 1 of 2)

File type	No. of operands	Supplied:		Deleted:
PAM	8	DB1 SI KB KE	with file name with blanks or “MAINITEM” PAM block numbers of range limits	File selection delimited by the PAM block numbers *KB/KE*
	7	DB1 SI KB	with file name with blanks or “MAINITEM” PAM block number	Block with the PAM block number from *KB*
	3	DB1 PAMHPNR	with file name (*RE* reference area) with PAM block number	Block with the PAM block number from *PAMHPNR*
	3	DB1 PAMHPNR	with file name (*RE* reference area) with value 0	All locked (but unmodified) blocks in this file
ISAM, PAM, DAM	3	DB3	with “(ALL)”	All locked (but unmodified) records or blocks of all files involved in the transaction
	2	DB1	no specification	

Table 13: Transfer of key values for the UNLK operation (part 2 of 2)

5.3 OPEN and USAGE modes

LEASY OPEN mode for the OPFL operation

The OPEN mode is specified in the *OPE-OM* field of the *RE* area.

LEASY OPEN mode	**S=SAM** **I=ISAM** **P=PAM** **D=DAM**	**DMS OPEN mode**	**USAGE modes permitted for OPTR**
1	I+P+D+S	INPUT	*PRRT, EXRT*
2	I+P+D	INPUT, SHARUPD	*RETR, PRRT, EXRT, ULRT*
3	I+P+D	INOUT	all ISAM, DAM and PAM USAGE modes except *ULRT* and *ULUP*
4	I+P+D	INOUT, SHARUPD	all ISAM, DAM and PAM USAGE modes
5	S	REVERSE	*PRRR, EXRR*
6		(reserved)	---
7	S	UPDATE	*EXUP*
8	S	OUTPUT	*EXLD*
9	S	EXTEND	*EXLD*
A	I+P+D	OUTIN	all ISAM, DAM and PAM USAGE modes except *ULRT* and *ULUP*
B	I+P+D	OUTIN, SHARUPD	all ISAM, DAM and PAM USAGE modes except *ULRT*

Table 14: LEASY OPEN modes

LEASY USAGE mode for the OPTR operation

The USAGE mode is specified in the *DB* area.

The table below is valid for master and model files. Foreign and temporary files use the DMS OPEN mode mentioned, but are always opened with *SHARED-UPDATE=NO* if the *OPFL* operation is not specified.

USAGE mode	**SAM**	**ISAM PAM DAM**	**Current transaction**	**Permitted access by parallel transactions**	**DMS OPEN mode**
RETR retrieval	-	+	read	read write	INPUT SHARUPD
PRRT protected retrieval	+	+	read	read	INPUT
PRRR protected retrieval reverse	+	-	read backwards	read	REVERSE
EXRT exclusive retrieval	+	+	read	no access	INPUT
EXRR exclusive retrieval reverse	+	-	read backwards	no access	REVERSE
UPDT update	-	+	read write	read write	INOUT SHARUPD
PRUP protected update	-	+	read write	read	INOUT SHARUPD
EXUP exclusive update	+	+	read write	no access	INOUT or UPDATE
EXLD exclusive load	+	+	write in load mode	no access	INOUT or EXTEND

Table 15: Defined USAGE modes for the OPTR operation (part 1 of 2)

USAGE mode	**SAM**	**ISAM** **PAM** **DAM**	**Current transaction**	**Permitted access by parallel transactions**	**DMS OPEN mode**
LOAD share load	-	+	read write (the ascending record key is allocated by LEASY)	read write	INOUT SHARUPD
PLOD protected load	-	+		read	INOUT SHARUPD
ELOD exclusive load	-	+		no access	INOUT
LDUP load + update	-	+		read write	INOUT SHARUPD
PLUP protected load +update	-	+		read	INOUT SHARUPD
ELUP exclusive load + update	-	+		no access	INOUT
ULRT unlocked retrieval	-	+	read	read write	INPUT SHARUPD
ULUP unlocked update	-	+	read write	read	INOUT SHARUPD

Table 15: Defined USAGE modes for the OPTR operation (part 2 of 2)

LEASY operations and compatible USAGE modes

The following table shows the operations permitted according to the USAGE mode of the file identifier and the DMS file type (*ACCESS-METHOD*).

Operation → **USAGE mode ↓**	RDIR	RHLD	SETL	RNXT	RNHD	RPRI	RPHD	INSR	STOR	REWR	DLET	LOCK	UNLK
RETR + PRRT + EXRT (ISAM + PAM + DAM)	x	x	x	x	x	x	x	-	-	-	-	x	x
UPDT + PRUP + EXUP+ ULUP (ISAM + PAM + DAM)	x	x	x	x	x	x	x	x	x	x	x	x	x
LDUP + PLUP + ELUP (ISAM + PAM + DAM)	x	x	x	x	x	x	x	x	-	x	x	x	x
LOAD + PLOD + ELOD (ISAM + PAM + DAM)	x	x	x	x	x	x	x	x	-	-	-	x	x
ULRT (ISAM + PAM + DAM)	x	-	x	x	-	x	-	-	-	-	-	-	-
EXLD (ISAM + PAM + DAM)	-	-	-	-	-	-	-	x	-	-	-	x	x
PRRT + EXRT (SAM)	x	x	x	x	x	-	-	-	-	-	-	-	x
PRRR + EXRR (SAM)	x	x	x	-	-	x	x	-	-	-	-	-	x
EXUP (SAM)	x	x	x	x	x	-	-	-	-	x	-	-	x
EXLD (SAM)	-	-	-	-	-	-	-	x	-	-	-	-	x

Table 16: LEASY operations as a function of the USAGE mode

Possible combinations of USAGE modes

The following table indicates the permitted USAGE modes for user *U2*, after user *U1* has opened a file with the USAGE mode specified.

B2 → **B1 ↓**	**RETR**	**UPDT**	**PRRT**	**PRRR**	**PRUP**	**EXRT**	**EXRR**	**EXUP**	**LOAD**	**LDUP**	**EXLD**	**PLOD**	**ELDO**	**PLUP**	**ELUP**	**ULRT**	**ULUP**
RETR	x	x	x	-	x	-	-	-	x	x	-	x	-	x	-	-	-
UPDT	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PRRT	x	-	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-
PRRR	-	-	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-
PRUP	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EXRT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EXRR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EXUP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LOAD	x	-	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-
LDUP	x	-	-	-	-	-	-	-	-	x	-	-	-	-	-	-	-
EXLD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PLOD	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ELOD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PLUP	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ELUP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ULRT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
ULUP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	-

Table 17: Compatibility of LEASY USAGE modes

The table below shows where compatibility exists between USAGE modes specified by a user within a transaction for the same logical file but for different sequence identifiers.

U2 → **U1 ↓**	RETR	UPDT	PRRT	PRUP	EXRT	EXUP	LOAD	PLOD	ELOD	LDUP	PLUP	ELUP	EXLD	PRRR	EXRR	ULRT	ULUP
RETR	RETR	UPDT	PRRT	PRUP	EXRT	EXUP	LOAD	PLOD	ELOD	LDUP	PLUP	ELUP	-	-	-	-	-
UPDT	UPDT	UPDT	PRUP	PRUP	EXUP	EXUP	-	-	-	-	-	-	-	-	-	-	-
PRRT	PRRT	PRUP	PRRT	PRUP	EXRT	EXUP	PLOD	PLOD	ELOD	PLUP	PLUP	ELUP	-	-	-	-	-
PRUP	PRUP	PRUP	PRUP	PRUP	EXUP	EXUP	-	-	-	-	-	-	-	-	-	-	-
EXRT	EXRT	EXUP	EXRT	EXUP	EXRT	EXUP	ELOD	ELOD	ELOD	ELUP	ELUP	ELUP	-	-	-	-	-
EXUP	EXUP	EXUP	EXUP	EXUP	EXUP	EXUP	-	-	-	-	-	-	-	-	-	-	-
LOAD	LOAD	-	PLOD	-	ELOD	-	LOAD	PLOD	ELOD	PLUP	PLUP	ELUP	-	-	-	-	-
PLOD	PLOD	-	PLOD	-	ELOD	-	PLOD	PLOD	ELOD	PLUP	PLUP	ELUP	-	-	-	-	-
ELOD	ELOD	-	ELOD	-	ELOD	-	ELOD	ELOD	ELOD	ELUP	ELUP	ELUP	-	-	-	-	-
LDUP	LDUP	-	PLUP	-	ELUP	-	PLUP	PLUP	ELUP	LDUP	PLUP	ELUP	-	-	-	-	-
PLUP	PLUP	-	PLUP	-	ELUP	-	PLUP	PLUP	ELUP	PLUP	PLUP	ELUP	-	-	-	-	-
ELUP	ELUP	-	ELUP	-	ELUP	-	ELUP	ELUP	ELUP	ELUP	ELUP	ELUP	-	-	-	-	-
EXLD	-	-	-	-	-	-	-	-	-	-	-	-	EXLD	-	-	-	-
PRRR	-	-	-	-	-	-	-	-	-	-	-	-	-	PRRR	EXRR	-	-
EXRR	-	-	-	-	-	-	-	-	-	-	-	-	-	EXRR	EXRR	-	-
ULRT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ULRT	-
ULUP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ULUP

Table 18: Rules for combining the USAGE modes of a logical file

6 COBOL interface

The user program calls LEASY by means of *CALL*s via subroutine linkage as is common in high-level languages.

The following standard registers are used:

R1	Address of the operand list
R14	Return address
R15	Branch destination

With a few exceptions, the operand definition is the same as for KLDS.

Calls from the main program

```
                          ⎧CAT⎫
CALL "LEASY" USING OP,RE, ⎨DB ⎬,AR,FA,SI,KB,KE,US.
                          ⎩CI ⎭
```

The required COPY elements are contained in the library *SYSLIB.LEASY.061*.

COPY element	Contents
LEASYPAR	Parameter for interface
LEASYKON	Constant for interface
LEASYRE	LEASY-RE area for the WORKING-STORAGE-SECTION
LEASYREL	LEASY-RE area for the LINKAGE-SECTION

7 Assembler interface

The LEASY Assembler macros are contained in the macro library *SYSLIB.LEASY.061*.

7.1 Definition macros

Operation	Operands
LEA@AR	[LEN=length] [,FOR=$\left\{\begin{array}{c}\underline{V}\\F\\D\end{array}\right\}$]
LEA@CALL	[[op],[re],[db],[ar],[fa],[si],[kb],[ke],[us]]
LEA@CAT	[catalog] [,catalog-suffix]
LEA@CI	[LEN=length]
LEA@DB	[string] [,LEN=length]
LEA@DB1	[file-identifier]
LEA@FA	[string]
LEA@OP	[operation]
LEA@OPS	
LEA@RE	
LEA@SI	[index]
LEA@US	[LEN=length]

Table 19: Definition macros

7.2 Action macros

Operation	Operands
LEA@BACK	[[re],[us]] [,SAVE=address2] [,PIDE=ide][,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@CAL	[[op],[re],[db],[ar],[fa],[si],[kb],[ke],[us]] [,SAVE=address2] [,POPE1=ext1] [,TOPE1=ext1] [,POPE2=ext2] [,TOPE2=ext2] [,POPEOM=openmode] [,TOPEOM=openmode] [,POPELOG=log] [,TOPELOG=log] [,POPEWTM=waiting-time] [,TOPEWTM=waiting-time] [,PSAMPTR=X'sam-pointer'] [,TSAMPTR=X'sam-pointer'] [,PPAMHP=X'pam-block-number'] [,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 1 of 12)

Operation	Operands
LEA@CATD	[[re],[cat],[us]] [,SAVE=address2] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@CINF	[[re],[ci]] [,SAVE=address2] [,POPE1=$\left\{\begin{array}{l}\texttt{F}\\ \\ \texttt{BLANK}\end{array}\right\}$][,TOPE1=$\left\{\begin{array}{l}\texttt{F}\\ \\ \texttt{BLANK}\end{array}\right\}$] [,POPE2=$\left\{\begin{array}{l}\texttt{C}\\ \texttt{O}\\ \texttt{T}\\ \\ \texttt{S}\\ \texttt{W}\\ \texttt{BLANK}\end{array}\right\}$][,TOPE2=$\left\{\begin{array}{l}\texttt{C}\\ \texttt{O}\\ \texttt{T}\\ \\ \texttt{S}\\ \texttt{W}\\ \texttt{BLANK}\end{array}\right\}$] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 2 of 12)

Operation	Operands
LEA@CLFL	[[re],[db],[us]] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@CLTR	[[re],[us]] [,SAVE=address2] [,POPE1=$\left\{\begin{array}{l}\text{R}\\ \text{BLANK}\end{array}\right\}$][,TOPE1=$\left\{\begin{array}{l}\text{R}\\ \text{BLANK}\end{array}\right\}$] [,POPE2=$\left\{\begin{array}{l}\text{T}\\ \text{BLANK}\end{array}\right\}$][,TOPE2=$\left\{\begin{array}{l}\text{T}\\ \text{BLANK}\end{array}\right\}$] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@DLET	[[re],[db],[ar],[fa],[si],[kb],[us]] [,SAVE=address2] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 3 of 12)

Operation	Operands
LEA@INSR	[[re],[db],[ar],[us]] [,SAVE=address2] [,POPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\\texttt{BLANK}\end{array}\right\}$] [,TOPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\\texttt{BLANK}\end{array}\right\}$] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\\texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@LOCK	[[re],[db],[ar],[fa],[si],[kb],[ke]] [,SAVE=address2] [,POPE1=$\left\{\begin{array}{l}\texttt{S}\\\texttt{BLANK}\end{array}\right\}$] [,TOPE1=$\left\{\begin{array}{l}\texttt{S}\\\texttt{BLANK}\end{array}\right\}$] [,POPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\\texttt{BLANK}\end{array}\right\}$] [,TOPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\\texttt{BLANK}\end{array}\right\}$] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\\texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 4 of 12)

Operation	Operands
LEA@MARK	[[re],[us]] [,SAVE=address2] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@OPFL	[[re],[db],[us]] [,SAVE=address2] [,POPEOM=openmode][,TOPEOM=openmode] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 5 of 12)

Operation	Operands
LEA@OPTR	[[re],[$\left\{\begin{matrix}\text{db}\\ \text{ci}\end{matrix}\right\}$],[us]] [,SAVE=address2] [,POPE1=$\left\{\begin{matrix}\text{W}\\ \text{BLANK}\end{matrix}\right\}$] [,TOPE1=$\left\{\begin{matrix}\text{W}\\ \text{BLANK}\end{matrix}\right\}$] [,POPEOM=openmode] [,TOPEOM=openmode] [,POPELOG=$\left\{\begin{matrix}\text{N}\\ \text{BLANK}\end{matrix}\right\}$] [,TOPELOG=$\left\{\begin{matrix}\text{N}\\ \text{BLANK}\end{matrix}\right\}$] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{matrix}\text{(error-code,...)}\\ \text{address3}\end{matrix}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 6 of 12)

Operation	Operands
LEA@PARC	[[op],[re],[db],[ar],[fa],[si],[kb],[ke]] [,ADDRLST=address1] [,LASTPAR=value] [,POPE1=ext1] [,POPE2=ext2] [,POPEOM=openmode] [,POPELOG=log] [,POPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\ \texttt{BLANK}\end{array}\right\}$] [,PSAMPTR=X'sam-pointer'] [,PPAMHP=X'pam-block-number'] [,PIDE=ide]
LEA@RDIR	[[re],[db],[ar],[fa],[si],[kb],[ke]] [,SAVE=address2] [,POPE2=ext2] [,TOPE2=ext2] [,PPAMHP=X'pam-block-number'] [,TPAMHP=X'pam-block-number'] [,PSAMPTR=X'sam-pointer'] [,TSAMPTR=X'sam-pointer'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 7 of 12)

Operation	**Operands**
LEA@REWR	[[re],[db],[ar],[us]] [,SAVE=address2] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{matrix}\text{(error-code,...)}\\ \text{address3}\end{matrix}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@RHLD	[[re],[db],[ar],[fa],[si],[kb],[ke]] [,SAVE=address2] [,POPE1=ext1] [,TOPE1=ext1] [,POPE2=ext2] [,TOPE2=ext2] [,POPEWTM=$\left\{\begin{matrix}\text{waiting-time}\\ \text{BLANK}\end{matrix}\right\}$] [,TOPEWTM=$\left\{\begin{matrix}\text{waiting-time}\\ \text{BLANK}\end{matrix}\right\}$] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{matrix}\text{(error-code,...)}\\ \text{address3}\end{matrix}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 8 of 12)

Operation	Operands
LEA@RNHD	[[re],[db],[ar],[fa]] [,SAVE=address2] [,POPE1=ext1] [,TOPE1=ext1] [,POPE2=ext2] [,TOPE2=ext2] [,POPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\ \texttt{BLANK}\end{array}\right\}$] [,TOPEWTM=$\left\{\begin{array}{l}\texttt{waiting-time}\\ \texttt{BLANK}\end{array}\right\}$] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@RNXT	[[re],[db],[ar],[fa]] [,SAVE=address2] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 9 of 12)

Operation	Operands
LEA@RPHD	[[re],[db],[ar],[fa]] [,SAVE=address2] [,POPE1=ext1] [,TOPE1=ext1] [,POPE2=ext2] [,TOPE2=ext2] [,POPEWTM=$\left\{\begin{array}{l}\text{waiting-time}\\\text{BLANK}\end{array}\right\}$] [,TOPEWTM=$\left\{\begin{array}{l}\text{waiting-time}\\\text{BLANK}\end{array}\right\}$] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\\text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@RPRI	[[re],[db],[ar],[fa]] [,SAVE=address2] [,PIDE=ide][,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\\text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 10 of 12)

Operation	Operands
LEA@SETL	[[re],[db],[ar],[fa],[si],[kb],[ke]] [,SAVE=address2] [,PPAMHP=X'pam-block-number'] [,TPAMHP=X'pam-block-number'] [,PSAMPTR=X'sam-pointer'] [,TSAMPTR=X'sam-pointer'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@STOR	[[re],[db],[ar],[us]] [,SAVE=address2] [,POPEWTM=$\left\{\begin{array}{l}\text{waiting-time}\\ \text{BLANK}\end{array}\right\}$] [,TOPEWTM=$\left\{\begin{array}{l}\text{waiting-time}\\ \text{BLANK}\end{array}\right\}$] [,PPAMHP=X'pam-block-number'] [,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\text{(error-code,...)}\\ \text{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]
LEA@TOLR	ERRCODE=(error-code,...)

Table 20: Action macros (part 11 of 12)

Operation	Operands
LEA@UNLK	[[re],[db],[ar],[fa],[si],[kb],[ke]] [,SAVE=address2] [,POPE1=ext1] [,TOPE2=ext2] [,PPAMHP=X'pam-block-number'][,TPAMHP=X'pam-block-number'] [,PIDE=ide] [,TIDE=ide] [,ERRCODE=$\left\{\begin{array}{l}\texttt{(error-code,...)}\\ \texttt{address3}\end{array}\right\}$,ERRADDR=address4] [,ERRADDR=address4]

Table 20: Action macros (part 12 of 12)

7.3 Macros for the evaluation of currency information (CI)

Operation	Operands
LEA@@DDL	[PRE=prefix]
LEA@@DSI	[PRE=prefix]
LEA@@DPL	[PRE=prefix]
LEA@@DRI	[PRE=prefix]

Table 21: Macros for the evaluation of currency information

8 Utility Routines

8.1 LEASY-CATALOG

Overview of statements

Statement	Meaning
[*]CAT[1]	Accesses the LEASY catalog
[*]COM	Comment text; inserts comment text, e.g. in procedures
[*]END[2]	Program termination; terminates the LEASY-CATALOG utility routine
[*]ERA	Erase request; erases a file or all instances of a model file from the LEASY catalog and the DMS catalog
[*]FIL	File specification; enters a new file in the LEASY catalog or updates the attributes of a file
[*]INF	Information request; supplies an information block from the DMS and/or LEASY catalogs
[*]PIN	Output of information on ISAM pools; outputs information on the ISAM pools defined in the LEASY catalog to SYSOUT
[*]POO	Definition of ISAM pool; defines the attributes of a LEASY-specific ISAM pool

Table 22: Overview of statements (LEASY-CATALOG)

[1] Mandatory; must be entered as the first statement

[2] Mandatory; must be entered as the last statement

Statements

Operation	Operands
[*]CAT[ALOG]	[:catid:]file-catalog N[] [,TYP[]={O[]}] E[] C'password' [,PAS[]={X'password'}] NONE C'newpassword' [,NEW[]={X'newpassword'}] NONE Y[] [,INF[]={ }] N[] [,VOL[]=vsn,DEV[]=device] Y[] [,CID[]={ }] N[] [:catid1:][$userid1.][copycat] [,CPC[]={ }] (NO) [:catid2:][$userid2.][suffix] [,CPS[]={ }] (NO) [,OLDL[OCATION]=[:catid1:] [$userid1]] [,NEWL[OCATION]=[:catid2:] [$userid2]] Y[] [,ROM[]={ }] N[]
[*]COM[MENT]	[text]
[*]END[]	
[*]ERA[SE]	file {file.} file.suffix X'writepassword' [,PAS[]={C'writepassword'}] NONE R[] [,CLE[]={ }] C[]

Table 23: LEASY-CATALOG statements (part 1 of 4)

Operation	Operands
[*]FIL[E]	{file {file.suffix} {file. [,NAM[]=[:catid:][$userid.]filename] {S[] {M[] [,LEA[]={ }] {T[] {F[] {N[] {Y[] [,AIM[]={(Y[],A[])}] {R[] {(R[],A[]) {Y[] [,BIM[]={ }] {N[] {VERY-HIGH {HIGH [,IOPERF={ }] {STD {USER-MAX {RDWRT [,IOUSAGE={WRITE}] {READ {VERY-HIGH {HIGH [,SIOPERF[]={ }] {STD {USER-MAX {C'writepassword' [,WRP[]={X'writepassword'}] {NONE {C'readpassword' [,RDP[]={X'readpassword'}] {NONE {C'writepassword' [,PAS[]={X'writepassword'}] {NONE

Table 23: LEASY-CATALOG statements (part 2 of 4)

Operation	Operands
	[,MOD[]={ N[] U[] C[] L[] }]
	[,LOC[]={ N[] Y[] }]
	[,RTF[]=(rpos,rlen)]

	[,KEY[]={
	(keyname,-)
	([rep*]keyname,{
	[iub,]pos,len[,dist]
	[iub,](pos,len[,dist]),...
	(iub,pos,len[,dist]),...
	}
	[,[dup][,[upd]]][,RTP={
	recordtype
	(recordtype,...)
	NONE
	}])
	}]...

	[,FCBTYPE={ ISAM SAM PAM DAM }]
	[,BLKCTRL={ PAMKEY DATA DATA2K DATA4K NO }]
	[,PLK[]={ *STD poollinkname }]
	[,SLK[]={ *POOL *STD poollinkname }]
	[,RECSIZE=recsize]
	[,SHA[]={ N[] Y[] }]

Table 23: LEASY-CATALOG statements (part 3 of 4)

Operation	Operands
	Y[] [,ROM[]=] N[]
[*]INF[ORMATION]	file [file.suffix] file. A[] S[] C[] [, T[]] P[] R[] L[]
[*]PIN[FORMATION]	[poolname] [,A[LL]]
[*]POO[L]	poollinkname *DEF[AULT] [,CAT[ID]=] catid *STD [,SIZ[E]=] size N[EW] [,MOD[E]= U[PDATE]] E[RASE] R[UNTIME] [,PCR[EATION]=] M[AINTASK]

Table 23: LEASY-CATALOG statements (part 4 of 4)

8.2 LEASY-CONVERT

Overview of statements

Statement	Function
CON	Specify the conversion direction, catalog ID and file protection
CAT	Specify the LEASY catalog to be converted
FIL	Specify the files to be converted
END	Initiate conversion and terminate LEASY-CONVERT

Table 24: Overview of statements (LEASY-CONVERT)

Statements

<table>
<tr><th>Operation</th><th>Operands</th></tr>
<tr><td>CAT[ALOG]</td><td>catalog [,C[ATID]= catid]
[,VOL[UME]= vsn]
[,DEV[ICE]= device]</td></tr>
<tr><td>CON[VERT]</td><td>P[AMKEY]
[TO= 4K-N[ONKEY]][,C[ATID]=catid]
N[ONKEY]
Y[ES]
[O[VERWRITE]=]
N[O]
Y[ES]
[T[RUNCATE]=]
N[O]</td></tr>
<tr><td>END</td><td></td></tr>
<tr><td>FIL[E]</td><td>file
[,C[ATID] = catid]
*ALL [,VOL[UME] = vsn|(vsn1,...)]
[,DEV[ICE] = device]
[,SIV[OL] = vsn|(vsn1,...)]
[,SID[EV] = device]
S[AME]
(STD,blksize)
[,BLKS[IZE]=]
blksize
A[DAPT]
F
[,T[YPE] =]
C
DATA
[,BLKC[TRL] =]
NO</td></tr>
</table>

Table 25: LEA.CONVERT statements

8.3 LEASY-IOTASK

Overview of statements

Statement	Meaning
[*]ARL	Define the maximum length of the AR area
[*]CAT[1]	Access the LEASY catalog
[*]DBL	Define the length of the DB file allocation
[*]END[2]	Terminate statement input
[*]IOT	Define the maximum number of I/O tasks
[*]KBL	Define the length of the KB area
[*]KEL	Define the length of the KE area
[*]OPF[3]	Define LEASY files
[*]QUE	Define queue processing for I/O tasks
[*]USE	Define the maximum number of application programs
[*]WAI	Define the maximum waiting time for a response from the I/O task

Table 26: Overview of statements (LEASY-IOTASK)

[1] Mandatory; this statement must be specified first.

[2] Mandatory; this statement must be specified last.

[3] Mandatory; this statement can be specified several times.

Statements

Operation	Operands
[*]ARL[EN]	=arlen
[*]CAT[ALOG]	=[:catid:][$userid.]file-catalog[,suffixname]
[*]DBL[EN]	=dblen
[*]END[]	
[*]IOT[TASK]	=maxiotask
[*]KBL[EN]	=kblen
[*]KEL[EN]	=kelen
[*]OPF[L]	=file,mode mode: permissible LEASY OPEN modes: 1 INPUT 2 INPUT,SHARUPD 3 INOUT 4 INOUT,SHARUPD 5 REVERSE A OUTIN B OUTIN,SHARUPD.
[*]QUE[UING]	$=\left\{\begin{array}{l}\underline{\text{LIFO}}\\ \text{FIFO}\end{array}\right\}$
[*]USE[R]	=maxuser
[*]WAI[TING]	=time

Table 27: LEASY-IOTASK statements

8.4 LEASY-LOADSI

Assigning the LEASY catalog

The first prompt issued by the LEA.LOADSI utility routine after being started is an input request for the LEASY catalog whose files are to be processed.

```
LEA0001 PLEASE TYPE IN NAME OF LEASY DIRECTORY
```

[:catid:][$userid.]file-catalog
Logical name of the LEASY catalog.

*END End of the LEASY-LOADSI run.

*HALT End of the LEASY-LOADSI run.

Assigning the primary file

```
LEA0203 PLEASE TYPE IN FILE SPECIFICATION
```

file[.suffix] [PAD=pad][,SIZE=size]
Specifies the primary file
where: $0 \leq \mathit{pad} < 99$
Default: PAD = 15
$1 \leq \mathit{size} \leq 2147483639$.

*END or blank or DUE
Handling of the last primary file processed in a LEASY catalog is terminated. The prompt which requests the input of a LEASY catalog is then repeated.

*HALT Termination of the LEASY-LOADSI utility routine.

Specifications for secondary index management

```
LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
```

NEW[ ] [keyname,...] Creates secondary index pointers. Any existing contents are first deleted.

ADD[ ] keyname,... Adds secondary pointers to those already existing.

DEL[ ] keyname,... Deletes secondary pointers from the SI file.

[*]END[ ] or blank or DUE
Terminates input sequence for a file. The request is then made to specify the next primary file.

[*]HALT[ ] Terminates the LEASY-LOADSI utility routine.

8.5 LEASY-MAINTASK

Overview of statements

Statement	Meaning
[*]ACA=acatid	Specify public volume set for AIM file
[*]ADE=device	Specify device type for AIM file
[*]AGE=gen	Specify number of generations of AIM file
[*]AGF=gen	Specify number of AIM file generations which are to be released
[*]AIB=page	Define AIM buffer
[*]AIO={VERY-HIGH HIGH STD USER-MAX}	Specify performance attributes for AIM file
[*]AIS={pamblocknummer (pamblocknummer,inkrement)}	Define AIM file size
[*]APP=anoita	Specify size of inquiry-and-transaction mode
[*]ASP={primary (primary,secondary) TAPE}	Specify memory space of a new AIM file generation
[*]AUT={Y N}	Specify automatic reconstruction
[*]AVO=vsn	Specify VSN of volume containing AIM file
[*]BCA=bcatid	Specify public volume set for BIM files
[*]BDE=device	Specify disk storage type for BIM file
[*]BIO={VERY-HIGH HIGH STD USER-MAX}	Specify performance attribute for BIM files
[*]BVO=vsn	Specify volume serial number of disk containing BIM files
[*]CAT=file-catalog	Access LEASY catalog
[*]COM[text]	Insert comment

Table 28: LEASY-MAINTASK statements (part 1 of 2)

Statement	Meaning
Y [*]DES= N	Specify processing of memory space on deletion
[*]END	Terminate statement input
Y[] [*]FAA= N[]	Always release AIM file generations
[*]FIL=files	Specify number of files
[*]KEY=keylen	Specify key length
B[] M[] A[][,][,K[]] [,P[]] R[] [*]LOG= M[] Y[][,] [,P[]] R[] N[][,K[]]	Declare backup
[*]MEM=mem	Define size of common memory
[*]MFB=mfact	Specify multiplication factor for defining the maximum number of buckets to be held free of lock element list requests from batch or TIAM tasks
[*]MTT=wtime	Define wait time with main task termination
[*]MUS=unitsize	Define size of memory units
[*]REN=enter-command	Define ENTER command for the RECONST task
[*]PAS=password	Pass password(s) for the RECONST task
C [*]STA= W[,R]	Determine cold/warm start
[*]TIM=time	Define maximum waiting time for canceling a lock
[*]TRA=maxtrans	Define maximum number of transactions
[*]TSK=notask	Specify size of task table
N [*]USE= C R	Define state of common memory

Table 28: LEASY-MAINTASK statements (part 2 of 2)

8.6 LEASY-MASTER

Overview of functions

The following overview of the functions of the LEASY-MASTER utility routine is arranged by content.

Action functions

- MAINTASK termination

TERM
Termination of the main task. Running transactions are not impeded; new transactions are accepted as long as there are still tasks linked to the common memory CMMAIN.
The general information mask is displayed subsequently.

CLOS
Normal termination of the LEASY session. The LEASY runtime system is to accept no more new transactions; transactions still running are not affected, however. The main LEASY task is terminated.
The general information mask then appears.

SHUT
Emergency stop. All current transactions are rolled back when next called; new transactions are no longer accepted. The main LEASY task is terminated.
The general information mask then appears.

- Lock functions

QUIE
New transactions are rejected; no current transactions are impeded.
The general information mask then appears.

HOLD
The session is suspended.
The general information mask then appears.

LOCT
A transaction is locked.
The transactions to be locked are selected in a separate mask.

LOCF
A file is locked.
The file is selected in a separate mask.

RLBT
A special transaction is to roll itself back when called again. The transaction is selected in a separate mask.

- Release functions

CONT *QUIE*, *HOLD* and *LOCT* are canceled.
The general information mask then appears.

UNLT A transaction lock is canceled.
The transaction is selected in a separate mask.

UNLF The lock on a file is canceled.
The file is selected in a separate mask.

- AIM file management

AIMA AIM file administration

AIMI Immediate AIM file generation switching, followed by messages on the screen indicating whether or not switching was successful.

AIMC AIM file generation switching after all current transactions have been terminated, followed by messages at the terminal indicating whether or not switching was successful.

AIMW Switching of AIM file generation after waiting for the end of the transaction.

AIME AIM file generation groups or AIM file generations are erased.
The generation is selected in a separate mask.

AIMS The status of the AIM file generations is displayed.

- SYSLST functions

ONPR A SYSLST listing is started containing all specified activities of the LEASY-MASTER utility routine from the present time.
The general information mask then appears.

OFFP The SYSLST listing is deactivated.
The general information mask then appears.

- Dump generation

DPRC Specification of a return code which will trigger a memory dump.

- Maintenance functions

  REPO The original files are replaced by shadow files.

  ROMS READ-ONLY mode is set.

  ROMR READ-ONLY mode is reset.

Display functions

- General

  GENT General information of the common memory CMMAIN is displayed in a separate mask.

  GENC The general counters of a LEASY session are displayed in a separate mask.

- Task-specific

  TSKT The task table is displayed in a separate mask.

  TSKC The counters of a particular task are displayed in a separate mask.

- Transaction-specific

  TRAT The transaction table is displayed in a separate mask.

  TRAC The transaction-specific counters are displayed in a separate mask.

- Application-specific

  FILT The table of all files of the common memory CMMAIN is displayed in a separate mask.

  OPFT The table of open master and model files is displayed in a separate mask.

  THAT The table of inquiry-and-transaction mode applications is displayed in a separate mask.
  UTMA may also be specified instead of *THAT* for reasons of compatibility.

- Lock elements

  SHLE The *SHLE* function is used to display the lock elements.

- ISAM pool

  PINF ISAM pool information is displayed in a separate mask.

- Others

CYCI Cyclic display of a table.
The control data (repetition factor, screen dwell time and table selection) is entered in a separate mask.

IO task functions

- IO task termination

IOTE Termination of an I/O task.

- IO task displays

IOGT The general I/O task table is displayed in a separate mask.

IOUT The I/O task user table is displayed in a separate mask.

IOTT The I/O task task table is displayed in a separate mask.

IOQA The contents of the queue are displayed in a separate mask.

LEASY-MASTER management functions:

HELP All possible functions of the LEASY-MASTER utility routine are displayed in a separate mask.

_ Termination of activities in conjunction with a particular common memory CMMAIN and return to the main task menu.
The general information mask then appears.

*END Immediate termination of the LEASY-MASTER utility routine without having to exit via the catalog menu.

8.7 LEASY-RECONST

Overview of statements

Statement	Meaning
[*]CAT[1]	Accesses the LEASY catalog
[*]COM	Comment text
[*]DAT	Date filter; restricts reconstruction to a particular date range
[*]END[2]	End of statement input; terminates input for the LEASY-RECONST utility routine and starts the reconstruction run.
[*]FIL	File selection; selects the files to be reconstructed
[*]MOD	Function selection; controls the sequence of operations of the LEASY-RECONST utility routine
[*]RAN	Control of range to be listed; restricts the range to be listed
[*]REP	Control of listing; controls the extent of the listing (SYSLST)
[*]SES	Session filter; restricts reconstruction to a particular session number range

Table 29: Overview of statements (LEASY-RECONST)

[1] Mandatory; must be specified prior to a **FIL* statement

[2] Mandatory; must be the last statement specified

The following table shows the permitted LEASY-RECONST control statements for user-defined ENTER files:

LEASY-RECONST control statement	**must be**	**can be**	**cannot be**
	specified		
CAT[ALOG]	X *)		
COM[MENT]		X	
DAT[E]			X
END[ ]	X		
FIL[ELIST]			X
MOD[E]		X *)	
RAN[GE]			X
REP[ORT]		X	
SES[SION]			X

*) Not all parameters are permitted for control statements **CAT* and **MOD*. The permitted parameters are given in the following table.

	must be	**can be**	**cannot be**
	specified		
CAT	file-catalog COP=(Y,A)	-	GEN FRO TOG
MOD	-	UPD=Y PRI	UPD=N SIU UNL TRA FRE

Statements

Operation	Operands
[*]CAT[ALOG]	[:catid:][$userid.]file-catalog [,GEN[]=[-]generation] [,FRO[]=[-]generation1] [,TOG[]=[-]generation2] $\texttt{[,COP[]=}\left\{\begin{array}{l}\texttt{Y[]}\\\texttt{(Y[],A[])}\\\underline{\texttt{N[]}}\end{array}\right\}\texttt{]}$
[*]COM[MENT]	[text]
[*]DAT[E]	$\left\{\begin{array}{l}\texttt{FRO[]=date1[,TOD[]=date2]}\\\texttt{TOD[]=date2}\end{array}\right\}$
[*]END[]	
[*]FIL[ELIST]	$\left\{\begin{array}{l}\underline{\texttt{A[]}}\\\texttt{[-](dvsname[,...])}\end{array}\right\}$
[*]MOD[E]	$\texttt{[PRI[]=}\left\{\begin{array}{l}\underline{\texttt{N}}\texttt{[ ]}\\\texttt{M[ ]}\end{array}\right\}\texttt{]}$ $\texttt{[,UPD[]=}\left\{\begin{array}{l}\underline{\texttt{Y}}\texttt{[ ]}\\\texttt{N[ ]}\end{array}\right\}\texttt{]}$ $\texttt{[,SIU[]=}\left\{\begin{array}{l}\texttt{Y[ ]}\\\texttt{N[ ]}\end{array}\right\}\texttt{]}$ $\texttt{[,UNL[]=}\left\{\begin{array}{l}\underline{\texttt{Y}}\texttt{[ ]}\\\texttt{N[ ]}\end{array}\right\}\texttt{]}$ $\texttt{[,TRA[]=}\left\{\begin{array}{l}\underline{\texttt{A}}\texttt{[ ]}\\\texttt{C[ ]}\\\texttt{V[ ]}\end{array}\right\}\texttt{]}$ $\texttt{[,FRE[]=}\left\{\begin{array}{l}\underline{\texttt{Y}}\texttt{[ ]}\\\texttt{N[ ]}\end{array}\right\}\texttt{]}$

Table 30: LEA.RECONST statements (part 1 of 2)

Operation	Operands
[*]RAN[GE]	[FRO[]=time1][,TOT[]=time2] [,FIR[]=tsn1][,LAS[]=tsn2]
[*]REP[ORT]	[LEN[]= { S[ ] M[ ] L[ ] A[ ][,EXT[]=(pos,len)]... }] [,LIS[]= { Y[ ] N[ ] }] [,REC[]= { A[ ] M[ ] [-](sa[,...]) }] [,USE[RINFORMATION]= { N[ ] Y[ ] }] [,PRO[]= { N[ ] Y[ ] }]
[*]SES[SION]	{ FRO[]=sessno1[,TOS[]=sessno2[,LAS[]=transno]] TOS[]=sessno2[,LAS[]=transno] }

Table 30: LEA.RECONST statements (part 2 of 2)

Reconstruction log

Field	Meaning
OP	Initiating LEASY action AIM element)
X	Flagged with "*" if this record has been declared invalid by a previous run with *MOD TRA=C.
Y	Flagged with "<" if the file cannot be processed due to a DMS error. The flag appears from the occurrence of the error until the file is closed.
S	Flagged with "s" (shortened) if the record was stored in truncated form in the AIM file.
POS	Byte position in relation to the PAM block number which is output in conjunction with each read operation in the AIM file.
SESSION	Session number
TRANS	Transaction number within the session
ITR	Internal LEASY transaction number

Table 31: Reconstruction log (part 1 of 2)

Field	Meaning		
TSN	TSN of the AIM record		
FILE	File name for file access operations		

TIME	Time counter printable in the form hh:mm:ss-t, ssssss
	t=S: daylight saving time (S = summer)
	t=W: standard time (W = winter)		

RECORD	AIM-record-specific information is output in this field:		
	MTSK *D* and SESS *D*	*D*	Date in the format *yyyy-mm-dd*
	CATD	T U P I	TSN User ID User program name "TSN-tsno" for programs running in timesharing mode; openUTM application name for openUTM programs running in inquiry and transaction mode; "UTM-tsno" for openUTM programs in test mode; DCAM application name for DCAM applications
	OPTR	B H A #	openUTM user ID for openUTM programs; user ID for DCAM applications, otherwise left blank. Processor name for openUTM programs, "DCA-tsno" for DCAM applications. "$DIALOG" for timesharing programs; openUTM application name for openUTM programs; DCAM application name for DCAM applications; openUTM session number for openUTM programs. "000" for DCAM applications, otherwise left blank.
		The file names or record key (up to the first 35 bytes) are given here, if possible, for record types which access LEASY files.	

Table 31: Reconstruction log (part 2 of 2)

AIM elements

Name	Action	Meaning
MTSK	Main task entry	LEASY-MAINTASK start
SESS	Session entry	Start of new session (LEASY-MAINTASK)
CATD	CATD entry	Connection to common memory
OPEN	OPEN entry	Physical opening of files
CLOS	CLOSE entry	Physical closing of files
OPTR	OPTR entry	Start of new transaction

Table 32: AIM elements (part 1 of 2)

Name	Action	Meaning
CLTR	CLTR entry	End of a transaction
RLBK	Rollback entry	Start of a rollback
STOR	STORE/PUT entry	Addition of a record (ISAM, DAM or SAM) or a block (PAM)
DLET	DLET entry	Deletion of an ISAM or DAM record or a block of a PAM file
PUTX	PUTX entry	Overwriting an ISAM or DAM record or a block of a PAM file
PUTS	PUTXSAM entry	Overwriting of a SAM record
ELIF	ELIMFILE entry	Deletion of a whole ISAM, DAM or PAM file
ELIR	ELIMREC entry	Delete an ISAM, DAM or PAM file beginning at a specified key
SETS	SETLSAM entry	SETL is used to truncate a SAM file
ENDA	End of AIM file entry	Main task has switched AIM file during the session
CSES	Continue session entry	Main task continues session in newly created AIM file
OLDB	Old buffer entry	Main task was unable to write the *ENDA* entry to the old AIM file after switching over to the new AIM file because of an I/O error. For this reason, the main task saves the contents (not yet written) of the AIM buffer in CMMAIN by entering them in the new AIM file and terminating with the *OLDB* element.
CTSK	Continue task entry	A LEASY task has linked itself to the newly created AIM file
FILS	Files list entry	A LEASY task has, at the moment of linkage to the new AIM file, physically opened the files specified in the FILS entry.
PETR	PETR entry	Suspended transaction
STOD	Store DAM buffer entry	Addition or overwriting of a block of a DAM file
CINF	CINF entry	Transfer currency information
LOCK	LOCK entry	Set record lock
RDIR	RDIR entry	Read record directly
RHLD	RHLD entry	Read record directly with record lock
RNXT	RNXT entry	Read next record
RNHD	RNHD entry	Read next record with record lock
RPRI	RPRI entry	Read preceding record
RPHD	RPHD entry	Read preceding record with record lock
UNLK	UNLK entry	Cancel record lock

Table 32: AIM elements (part 2 of 2)

8.8 LEASY-SAVE

Specify volume

```
LEA0605  ENTER DEVICE TYPE AND RETENTION PERIOD.
         REPLY ('DISK <,RETPD=DAYS>' OR 'TAPE <,RETPD=DAYS>'
         OR '*END')
```

DISK The files and LEASY catalog(s) are saved to disk.

TAPE The files and LEASY catalog(s) are saved to tape.

RETPD The save files cannot be deleted during this period (entered in days); range of values: 0 to 32767.

*END[ ] The LEASY-SAVE utility routine is terminated.

Assign volume serial numbers

```
LEA0606  ENTER VSN. REPLY (VSN OR 'PO' OR 'OP' OR NO SPECIFICATION OR '*END')
```

PO[OL] For saving to tape only.
The volume serial numbers are taken from the tape pool of the associated directory file.

OP[ERATOR] For saving to tape only.
This entry requests a VSN at the console during the save operation.

vsn For saving to disk or tape.
The volume with the VSN *vsn* is requested.

(vsn,...) Several volume serial numbers can be specified.

Blank or DUE Both these entries have the same effect as the *POOL* specification.

*END[ ] The LEASY-SAVE utility routine is terminated.

Assign device

```
LEA0607  ENTER DEVICE. REPLY (NAME OF DEVICE OR NO SPECIFICATION OR '*END')
```

device defines the device type.

Blank or DUE For saving to tape only.
Both these entries have the same effect as the *TAPE* specification.

*END[ ] The LEASY-SAVE utility routine is terminated.

Output REPORT list

```
LEA0602  ENTER LIST OPTION. REPLY ('SYSO' OR 'SYSL' OR 'BOTH'
         OR 'NONE' OR '*END')
```

SYSO[UT] In the case of an interactive task, the REPORT list is output via the terminal.

SYSL[ST] The REPORT list is output via the printer.

BOTH The REPORT list is output via SYSLST and SYSOUT.

NONE No REPORT list is generated.

*END[ ] The LEASY-SAVE utility routine is terminated.

Define type of save operation

```
LEA0603  ENTER CHANGED OPTION. REPLY ('YES' OR 'YES,LARGE' OR 'YES,NUMBER OF
         PAGES' OR 'NO' OR '*END')
```

YES All those files are saved which have been changed since the last full save or which are not in the ARCHIVE directory file.

YES,LARGE Only those files which are marked as *LARGE* in the catalog are handled as large files, and only the changed pages in these files are saved.

YES,number of pages

Only those files occupying more pages than are specified here are treated as large files, and only the changed pages in these files are saved.

NO The specified files are fully saved.

*END[ ] The LEASY-SAVE utility routine is terminated.

i For PAM and DAM files in the format *BLOCK-CONTROL-INFO=NO* partial saving with *YES,LARGE* or *YES,number of pages* is not possible. These files must always be saved in full with *YES* or *NO*.

Assign LEASY catalog

```
LEA0604  ENTER NAME OF LEASY DIRECTORY AND ASK FOR CATID.
         REPLY ('DIRECTORY <,CATID=YES>')
```

```
[:catid:][$userid.]file-catalog[,CATID=YES]
```

Specify user IDs for temporary files

```
LEA0608 ENTER USERIDS OF TEMPORARY FILES. REPLY (USERIDS OR NO SPECIFICATION)
```

```
[:catid:]$userid1[,[:catid:]$userid2,...,[:catid:]$userid5]
```

Blank or DUE	Only occurrences of temporary files under the ID of the catalog are saved.

Continue save operation

```
LEA0609 NEXT LEASY DIRECTORY. REPLY ('Y' OR 'N' OR '*END')
```

Y[ES]	A further LEASY catalog is to be saved.
N[O]	New save operands are requested.
*END[ ]	The LEASY-SAVE utility routine is terminated.

9 Return codes

LEASY-internal error code RC-LC arranged in ascending order

RC-LC	Meaning
L000	Function correctly executed (all operations)
L001	Record with key not located (RDIR, RHLD, REWR, DLET)
L002	Duplicate (RNXT, INSR for primary or secondary key, REWR, STOR for secondary key where DUPEKY = NO)
L003	EOF for sequential reading (at file end for RNXT and RNHD, at file beginning for RPRI and RPHD) or positioning error: sequential read instruction without current range (RNXT, RNHD, RPRI, RPHD) or EOF for INSR in the case of ISAM (USAGE modes LOAD/PLOD/ELOD and LDUP/PLUP/ELUP)
L004	Sequence error in load mode (INSR)
L005	Record not locked (DLET, REWR)
L006	Timeout for locking attempt expired (LOCK, RHLD, RNHD, RPHD, INSR, STOR)
L007	Deadlock during locking attempt (LOCK, RHLD, RNHD, RPHD, INSR, STOR)
L008	Record cannot be unlocked because it was updated in the transaction (UNLK)
L009	Warning: record to be unlocked has not been locked (UNLK)
L010	Length error in variable-length record (INSR, REWR, STOR)
L011	Warning: more than 255 records per block (RNXT, RPRI; SAM) when using a SAM retrieval address in 24-bit format
L012	No current record exists (REWR; SAM) or no valid read instruction for the file identifier (before DLET without key specification)
L013	Key outside permitted range; highest PAM block number of block to be written must be ≤ (FILESIZE + SECONDARY ALLOCATION) (INSR, STOR; PAM and DAM)
L014	Rollback not possible as transaction without BIM saving
L015	openUTM: task deadlock
L016	Writing of a DAM file record or PAM file record with BLOCK-CONTROL-INFO= WITHIN-DATA-BLOCK or BLOCK-CONTROL-INFO=NO is not possible since X'FF' is set in the first byte of the record (erase identifier for DAM) (INSR, STOR, REWR)
L017	No /ADD-FILE-LINK command issued for the specified link name.

Table 33: LEASY-internal error code RC-LC in ascending order (part 1 of 7)

RC-LC	Meaning
L018	In terms of syntax, the name of the file assigned via the /ADD-FILE-LINK command is not a LEASY catalog
L019	During a sequential read operation via an ISAM secondary key the record read immediately beforehand cannot be found.
L101	File not specified in OPTR of this transaction (all operations whose 3rd operand specifies a file identifier)
L102	Operation not permitted - contrary to FCBTYPE and/or USAGE mode (all operations whose 3rd operand specifies a file identifier)
L103	No transaction open (CLTR for all operations whose 3rd operand specifies a file identifier)
L104	Transaction opened with CATD or DISCONNECT/openUTM
L105	File name or suffix not defined in LEASY catalog (OPFL, OPTR)
L106	USAGE mode incompatible with OPEN mode (OPTR after OPFL)
L107	Additional specification for model file missing (OPFL, OPTR)
L108	FILE table overflow (OPTR) – increase *FILE statement in LEASY-MAINTASK
L109	Secondary index name not defined in LEASY catalog (RDIR, RHLD, SETL) or ISAM secondary index specified for SETL.
L110	File/file identifier cannot be opened with USAGE mode or result USAGE mode requested, as it has already been opened by another transaction with a higher USAGE mode (OPTR)
L111	USAGE mode incompatible with already opened file/file identifier in the same transaction
L112	KEYLEN (ISAM file) > *KEY statement for LEASY-MAINTASK (OPTR)
L113	KEYLEN > 4 for USAGE modes LOAD, ELOD, PLOD, LDUP, PLUP, ELUP (OPTR; ISAM)
L114	Record length incompatible with block length or invalid BLKSIZE (OPFL, OPTR)
L115	The required sequence identifier was not specified for this file in earlier OPTR operations of this transaction (all operations specifying a file identifier in the 3rd operand)
L116	No CLFL executed (CATD after OPFL) or the file has already been opened (OPFL)
L117	No CLTR executed (OPFL after OPTR)
L118	CLFL: at least one of the specified files has not been opened by OPFL
L119	No CLTR executed (CLFL after OPTR)
L120	File (OPTR) not specified in previous file list (OPFL) (OPTR after OPFL)
L122	File identifier already open
L123	AIM buffer too small (*AIB in LEASY-MAINTASK) in relation to maximum RECSIZE (OPFL, OPTR) or warm start with LEASY-MAINTASK without AIM saving, although this was activated for the transaction to be rolled back
L124	2nd OPTR call without using OPFL
L125	Entries in the LEASY catalog and those in the DMS catalog are inconsistent

Table 33: LEASY-internal error code RC-LC in ascending order (part 2 of 7)

RC-LC	Meaning
L126	Incorrect file format (BLKCTRL=NO)
L130	File size exceeds 32 GB
LI01	CATD call is missing (foreign files are not permitted)
LI02	No transaction is active (DCAM LU80)
LI03	Overflow in transfer area; maximum number of application programs has been exceeded
LI04	Internal IOH error: waiting time for the I/O task has expired (*WAI statement)
LI05	Internal IOH error: I/O task has been terminated with errors when processing a LEASY call; the transaction is reset
LI06	Internal IOH error: I/O task has been terminated with errors when processing a LEASY call; the transaction is not reset
LI07	Internal IOH error: initialization error; common memory is not released
LI08	Version error; the internal version is incompatible with I/O task
LI09	Internal IOH error: semaphore (protected variable) cannot be accessed; error in internal synchronization
LI10	Internal IOH error: the record length in the CINF area is greater than the length specified in the DBL statement
LI11	File not specified in the OPF statement
LI12	Record length greater than 0 or greater than the value in the ARL statement
LI20	Versions of runtime system and I/O task do not match
LI26	Version of link module < V5.1
LP01	Operation code is incorrect (all operations)
LP02	Too few operands (all operations)
LP04	OPE1/OPE2 incorrect (CLTR)
LP06	USAGE mode incorrect or invalid (OPTR)
LP07	OPEN mode incorrect or invalid OPFL: foreign file, SHAREUP=YES, BIM=YES, OPEN mode for write. OPTR: USAGE mode not compatible with OPEN mode.
LP08	Field selection incorrect, "(ALL)" (SETL, RDIR, RHLD)
LP09	Syntax error in file list (OPFL, OPTR, CLFL)
LP10	Syntax error in catalog name (CATD)
LP11	CI area too small for currency information (CINF) or no information in the CI area (ci-slf=0
LP12	L-OPT incorrect, ≠'1' (all operations)
LP14	PAMHPNR/SAMPTR invalid (in all operations in which these fields are evaluated)
LP15	OPE-WTIME non-numeric (all operations)

Table 33: LEASY-internal error code RC-LC in ascending order (part 3 of 7)

RC-LC	Meaning
LP16	OPE-OM in RE area is set incorrectly
LP17	Invalid combination of (KB, KE) (SETL, RDIR for SAM file)
LP18	Syntax error in file identifier (for all operations with specification of DB1)
LP19	OPE-STX incorrect (CATD)
LP20	The length of the USER area is not in the range 5 < len < 1024
LS01	Common memory CMMAIN of main task not created for specified LEASY catalog (CATD, OPTR)
LS02	Operation is rejected because of CLOS or SHUT function (CATD, OPFL, OPTR)
LS03	Too many transactions - transaction table overflow (OPTR); increase *TRANS statement in LEASY-MAINTASK
LS04	Common memory CMMAIN is locked for the runtime system (*USE=R in LEASY-MAINTASK)
LS05	No operation at all possible at the moment because of HOLD function
LS06	No new transaction possible at the moment because of QUIE function
LS07	No operation for this transaction possible at the moment because of LOCT or QUIE function
LS08	Rollback due to second LS12
LS09	OPE2=T is ignored in CLTR operations because of SHUT, CLOS, RLBT or REPO function
LS10	Operation is converted to CLTR with OPE1=R because of RLBT or SHUT function
LS11	Virtual memory exhausted (REQM, ENAMP macros)
LS12	Overflow of the transaction element area (in the case of OPTR) or the lock protocol element area while attempting to enforce a new lock element; increase *MEM statement in LEASY-MAINTASK
LS13	The file is locked by the LEASY-MASTER utility routine (OPFL and OPTR)
LS14	The file is locked against opening in write mode by the LEASY-MASTER utility routine (OPFL and OPTR)
LS15	Task table overflow, increase *TSK operand in LEASY-MAINTASK utility routine
LS17	Error in job variable function
LS18	DVS error with CATALOG file
LS19	DVS error with SI file
LS20	General DVS error
LS21	DVS error with BIM file
LS22	DVS error with AIM file
LS23	Error during rollback (CLTR,OPE1=R)

Table 33: LEASY-internal error code RC-LC in ascending order (part 4 of 7)

RC-LC	Meaning
LS26	Version of link module < V5.1
LS30	STXIT macro error in LEASY module
LS31	Error in dynamic loading of a module
LS32	ENASI macro error
LS33	RELM macro error
LS34	DISSI macro error
LS35	ENAMP macro error
LS36	Version of LEACON module is incompatible with version of LEASY module
LS37	ENQAR macro error
LS38	DEQAR macro error
LS40	LEASY system error: enforced lock element not located
LS41	LEASY system error: internal lock for record splitting frozen in secondary file
LS42	LEASY system error: duplicate in secondary file when splitting record
LS43	Inconsistency between primary and secondary index files: no primary record exists for SI entry, or it contains an incorrect secondary key value. Record with primary key not found Record found, but record-type field is invalid Record found, but does not contain an SI key.
LS44	Format error in BIM file (during rollback)
LS45	LEASY system error: inconsistency in common memory (internal secondary index number not located)
LS47	LEASY system error: logic error in LEAWRAIM
LS48	LEASY system error: MVC lock frozen in LEAWRAIM
LS49	LEASY system error: WRT lock frozen in LEAWRAIM
LS51	LEASY system error: AIM buffer is full and cannot be cleared because of an error in PAM-WRITE
LS52	Format error in PAM file
LS53	LEASY system error: AIMSWITCH lock frozen in LEALAISW
LS54	LEASY system error: open file table frozen in LEASPERR
LS55	LEASY system error: transaction table lock frozen in LEASPERR
LS56	LEASY system error: free chain lock frozen in LEASPERR
LS57	LEASY system error: release lock frozen in LEASPERR
LS58	LEASY system error: file table lock frozen in LEAFTIN

Table 33: LEASY-internal error code RC-LC in ascending order (part 5 of 7)

RC-LC	Meaning
LS59	Error when writing a DAM data block: error in S1 or AIM processing has forced an automatic rollback of the transaction (CLTR, all operations whose 3rd operand specifies a file identifier)
LS60	LEASY system error: lock of deadlock bit matrix is frozen
LS61	Error in ENAEI macro
LS62	Error in ENACO macro
LS63	Error in SOLSIG macro
LS64	Error in POSSIG macro
LS65	Main task has been terminated with errors (e.g. when writing the AIM buffer to tape)
LS66	LEASY system error: error in the truncation of AIM records
LS67	LEASY system error: incorrect call for LEAKMP module
LS68	Version of the runtime system is not identical with the version of CMMAIN common memory
LS69	Error in the DISMP macro
LS70	Error in the DISEI macro
LS71	Error in CREPOOL macro (for NK-ISAM)
LS72	Error in DELPOOL macro (for NK-ISAM)
LS73	Error in ADDPLNK macro (for NK-ISAM)
LS74	Error in REMPLNK macro (for NK-ISAM)
LS75	The LEASY statement cannot be processed. The AIM file generation has reached the maximum size or it cannot be switched over (for system reasons, e.g. pubspace limit reached or because no AIM file generation is free and the value 0 was specified as an increment in the AIS statement of LEASY-MAINTASK).
LS76	Transaction semaphore could not be obtained.
LS77	Because of ROMS function, currently no LEASY statements which modify the data set (DLET, INSR, REWR, STOR) are possible.
LS78	No new transactions permitted because of REPO.
LS79	Transaction already reset because of READ-ONLY mode (LEASY-MASTER, ROMS) or copying of shadow files (LEASY-MASTER, REPO).
LS80	No statements expect CLTR permitted because of REPO.
LS81	AIM file can no longer be written because of an error, no further LEASY request permitted, transaction was reset by LEASY.
LU01	openUTM: invalid start operand
LU02	openUTM: syntax error in start operand

Table 33: LEASY-internal error code RC-LC in ascending order (part 6 of 7)

RC-LC	Meaning
LU10	openUTM: missing or insufficient start operands DCAM: error in start operation sequence (CATD and/or OPFL omitted)
LU11	openUTM/DCAM: less than 2 LEASY operands
LU12	openUTM/DCAM: OPEN mode not permitted for foreign or SAM files (file is read-only) (OPFL)
LU13	openUTM/DCAM: LEASY temporary file not permitted (OPFL)
LU14	openUTM: after a delayed CLTR a CALL-LEASY is not permitted in the same dialog step (all operations)
LU15	openUTM: file must not be opened for writing in transactions without BIM saving (OPTR)
LU16	openUTM/DCAM: error in intertask synchronization for OPFL or CLFL, or different sequence for OPFL
LU17	DCAM: error; open transaction within DCAM application for OPFL or CLFL
LU18	DCAM: error; transaction cannot be active in more than one task at the same time
LU50	openUTM/DCAM: application table overflow
LU51	openUTM/DCAM: inconsistency in the application table
LU52	openUTM/DCAM: internal lock of the application table is frozen
LU53	DVS error with STATUS file
LU54	openUTM: status inquiry for the current LEASY session with openUTM application transactions still open
LU80	openUTM: error in openUTM call sequence at openUTM database interface DCAM: DCAM application name missing (CATD); transaction identifier missing or errored (all operations within a LEASY transaction)
LU81	openUTM: OPFL call missing (OPTR)
LU82	openUTM: start operand does not start with ".LEASY_"
LU83	openUTM: incorrect operation code
LU84	openUTM: status call: operation code neither "inquiry" nor "delete"
LU85	openUTM: error in processing of suspended transactions
Addd	DMS error during processing of an AIM file
Bddd	DMS error during processing of a BIM file
Cddd	DMS error during processing of a catalog file
Dddd	DMS error during processing of a primary file
Jddd	Error during processing of a job variable
Sddd	DMS error during processing of a secondary file
Tddd	DMS error during processing of a LEASY status file

Table 33: LEASY-internal error code RC-LC in ascending order (part 7 of 7)

Index

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *…@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at
http://ts.fujitsu.com/...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *…@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.