LEASY V6.2A

Utility Routines

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

1.1 Brief product description

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

LEASY provides various security and backup features designed to ensure the consistency of data.

LEASY supports

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

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

LEASY can be used in timesharing mode (TIAM) and in transaction mode (openUTM, DCAM).

LEASY requires various utility routines for initialization and execution purposes; these utility routines are described in this manual.

1.2 Target group

This manual is intended primarily for organizers and administrators of LEASY applications.

The next page describes the target groups for which the various parts of the manual are intended. Generally speaking, the following requirements apply:

– In order to use LEASY you will need a general knowledge of BS2000, including its data management system (DMS).

– If you are working with LEASY via the COBOL interface, you will need to be familiar with the COBOL programming language; if you are using the Assembler interface, you will need to be familiar with the macro assembler and the BS2000 system macros.

– If you are operating LEASY in connection with openUTM, in a DCAM environment or in a multiprocessor system environment, you will need to be familiar with openUTM, DCAM and MRS/RFA respectively.

1.3 Summary of contents

The LEASY software product is described in three manuals:

- LEASY Program Interface and Strategies
- LEASY Utility Routines
- LEASY Ready Reference

The **LEASY Program Interface and Strategies** manual provides an overview of the LEASY software program and contains all the information a developer of LEASY application programs will need.

The **LEASY Ready Reference** manual is aimed at application developers and administrators of LEASY. By bringing together all the LEASY command and operands and various tables in one handy volume it should make working with LEASY much easier.

The present manual, **LEASY Utility Routines**, describes the LEASY utilities and contains all the information needed for using the LEASY utility routines.

- **Administrators** will find all the relevant specifications for startup, working with other software products and monitoring operation, as well as for logging and saving.
- **Organizers** are provided with a description of the individual utility routines, the operations necessary for implementing their objectives and the extent to which cyclic activities must be scheduled.

For “novices”

A concise introduction with a description of the most important features can be found in chapter ”Basic features of LEASY” of the manual “LEASY (BS2000/OSD) - Program Interface and Strategies”. It is essential that this section be read first, particularly for users who are using the LEASY software product for the first time. When you have gained a general idea of the layout of LEASY from this manual, you can move on to the “LEASY (BS2000/OSD) - Utility Routines” manual.

Once you have read the brief introduction in chapter “Overview of the LEASY utility routines” on page 13ff, you can read any of the other chapters in this manual independently of one another. When individual sections in other chapters are necessary for your understanding, you will be directed to the relevant passages in the manual.

For experienced LEASY users

For information on the basic innovations in Version 6.2A as they affect the utility routines, read the list of amendments on the next page. This list provides direct references to the sections where these new features are described in detail. For information on other new features, please refer to the preface of the partner manual, “LEASY (BS2000/OSD) - Program Interface and Strategies”.

1.4 Changes since the previous manual

The changes in the present manual as compared with the LEASY V6.1A edition derive in the main from the following new features of LEASY Version 6.2A:

- The connection modules LEAMAST and LEAMCON are no longer provided.
- Controlled release of AIM file generations

  AIM file generations are by default protected against being deleted. Before they can be deleted, they must be released. They are released either automatically by LEASY after the shadow files have been automatically reconstructed, or they must be released by the LEASY administrator using the *AIMA* function of the LEASY-MASTER utility routine when the files are no longer required, see page 166.

  The LEASY-MAINTASK statements **AGF*, **AIS* and **FAA* control the behavior with regard to releasing AIM file generations, see page 129, page 131 and page 139.
- Replacing original files by shadow files during ongoing operation

  The *REPO* function of the LEASY-MASTER utility routine enables files to be replaced by their shadow files without terminating LEASY operation, see page 196.
- Online save

  The *ROMS* function of the LEASY-MASTER utility routine enables a write lock to be assigned to files, see page 203. These files can then be saved during ongoing operation. The *ROMR* function (page 202) cancels this write lock. To permit the *ROMS* function to be used, the files concerned must be reserved using the LEASY-CATALOG utility routine (*ROM* operand in the **CAT* and **FIL* statements, see page 30 and page 45).

  Specifying **INF ,O* in the LEASY-CATALOG utility routine causes the files to be displayed which are reserved for setting the write lock, see page 30.

1.5 Notational conventions

In continuous text the names of commands, operands, files, paths and screen elements are shown in *italics*. Important terms and pairs of opposites appear in **bold**.

Message texts and examples of system outputs are shown in a `fixed-pitch font`.

Texts that the user must enter are shown in a **`bold fixed-pitch font`**.

⚠ This symbol indicates an important warning which you must heed in the interest of system security or operational reliability, otherwise there may be gaps in the security coverage, loss of data or blockage of computers or lines.

ℹ This symbol precedes an important message which you should heed in the interest of system security or operational reliability.

Where references are made to other publications, the titles are abbreviated. The complete title of each publication referred to is given under "Related publications" at the back of the manual.

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
{ }	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 or KEY1,X' 00'
...	Dots indicate a repetition; the preceding syntactical unit can be repeated several times in succession. The maximum number of repetitions is specified in the operand description.	(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.	$\text{INF=}\left\{\begin{array}{l}\text{Y}\\ \underline{\text{N}}\end{array}\right\}$ The user enters: INF=Y or INF=N or nothing (i.e. same as INF=N)

Table 1: Notational conventions

2 Overview of the LEASY utility routines

The LEASY software product incorporates the following utility routines:

- **LEASY-CATALOG**

  for defining, modifying and deleting a LEASY catalog and its files and attributes.

- **LEASY-CONVERT**

  for converting existing LEASY applications files from PAMKEY to NK format and vice versa.

- **LEASY-IOTASK**

  for starting an I/O task for central file access.

- **LEASY-LOADSI**

  for generating secondary key pointers corresponding to the secondary keys defined in the LEASY file catalog.

- **LEASY-MAINTASK**

  for generating the common memory CMMAIN required by the LEASY runtime system and LEASY-RECONST, for creating AIM and BIM files, and for initializing the system (cold/warm start).

- **LEASY-MASTER**

  for controlling and monitoring the common memory CMMAIN.

- **LEASY-RECONST**

  for restoring destroyed LEASY files.

- **LEASY-SAVE**

  for saving a LEASY catalog and its files

2.1 Using the utility routines

This section describes the rules for starting utility routines in interactive and batch mode as well as the rules that must be taken into account for inputs to the utility routines.

2.1.1 Inputs to the utility routines

All utility routines can be used in interactive or batch mode. The following notes apply to both modes.

i The LEASY-LOADSI and LEASY-SAVE utility routines provide user prompting (guided dialog) and therefore have slightly different syntax rules.

The LEASY-MASTER utility routine provides a mask-like input and output. Therefore, not all the syntax rules apply.

- The statements from all utility routines are read by SYSDTA.
- If the statements are not input at a terminal, they are automatically logged in the system file SYSOUT.
- Each statement must be in columns 1-4 of the input record for all utility routines except LEASY-MASTER.
- An asterisk (*) may prefix each statement. In the case of LEASY-CATALOG, LEASY-CONVERT and LEASY-RECONST, the statement name must be followed by at least one blank; with LEASY-MAINTASK it can be suffixed directly by the equals sign (=).
- Generally, any sequence of statements or keyword operands within the statements is permitted; special notification is given of exceptions.
- If a statement contains only keyword operands, and if all of these operands are optional, the comma preceding the first keyword operand is omitted.

- The maximum input record length is 30 KB.

  Record chaining must be adopted to obtain readable input. In this way the inputs can extend over several input records whose total length can be up to 30 KB long. The last character of a record to be continued must be a hyphen (-) to indicate that continuation records will follow.

  When input is via a terminal, the logical end-of-line (LZE) character (X'15') can be used to spread the record contents over several lines to make it easier to read. LZE is handled internally as a blank.

- The characters ( ), = _ and LZE (logical end-of-line, X'15') are separators, i.e. they may be preceded or followed by any number of blanks.

2.1.2 Starting utility routines

The LEASY utility routines are started with the following SDF commands:

/START-LEASY-CATALOG

/START-LEASY-CONVERT

/START-LEASY-IOTASK

/START-LEASY-LOADSI

/START-LEASY-MAINTASK

/START-LEASY-MASTER

/START-LEASY-RECONST

/START-LEASY-SAVE

In SDF guided dialog these commands are available in the **UTILITIES** domain.

The syntax conventions for SDF are not described in this manual. For a detailed description, see, for example, the “Introductory Guide to the SDF Dialog Interface”.

START-LEASY-...
VERSION = ***STD** / <product-version>
,**MONJV** = ***NONE** / <filename_1..54_without-generation_version>
,**CPU-LIM**IT = ***JOB-REST** / <integer_1..32767 seconds>

VERSION =
This operand specifies the utility routine version.
A version specified in the *START-LEASY-utility* command always has priority over a version specified in a possibly prior *SELECT-PRODUCT-VERSION* command. If no version is specified in a *START-LEASY-utility* command, the latest installed version is used, if no *SELECT-PRODUCT-VERSION* command was issued before the *START-LEASY-utility* command. Otherwise, the *START-LEASY-utility* command takes over the version specified in the *SELECT-PRODUCT-VERSION* command.

VERSION = <product-version>
Explicit specification of the product version in the form mm.n[a[so]], e.g. 6.2 or 6.2A or 6.2A00.

MONJV =
Assignment of a job variable (JV) for program monitoring. There is no point in specifying this operand unless the JV software product is installed.

MONJV = *NONE
The program is not monitored with a JV.

MONJV = <filename_1..54_without-generation-version>
Name of the JV that is to monitor the program. While the program is running, the system sets the JV to the appropriate value:

$R	Program is running
$T	Program was successfully terminated
$A	Program was aborted with an error

CPU-LIMIT =
This operand assigns the utility routine a CPU time specified in seconds.
If the program exceeds this limit during execution, the user is informed by the system in interactive mode.
When the run is continued, the program is processed to completion. In batch mode, the program is terminated.

CPU-LIMIT = *JOB-REST
The remaining CPU time for the task is defined as a maximum program runtime.

CPU-LIMIT = <integer_1..32767 seconds>
Explicit specification of the CPU time (in seconds).

2.2 Optimization

For assistance with regard to improving the performance of LEASY applications, have a look at the instructions in the section “Planning a LEASY operation” of the “LEASY (BS2000/OSD) - Program Interface and Strategies” manual.

2.2.1 Optimization of LEASY-CATALOG

The following two operands of the **FIL* statement are of prime importance for optimizing the program runtime:

- The BLKSIZE operand

  The ideal block length for primarily sequential access is approximately from (*STD,3*) to (*STD,5*), and for direct access (*STD,1*). Longer data blocks should be used in exceptional cases only.

- The SPACE operand

  The size of the values for primary and secondary allocation depends on the block size on the one hand, and on the anticipated amount of data on the other.

Depending on the range of primary key numbers, a check should also be carried out as to whether the PAM access method can be used for part of the primary files.

In the case of large ISAM files, the option of splitting into data and index sections should be considered (operands *VOLUME*, *DEVICE-TYPE*, *DATA-VOLUME* and *DATA-DEVICE-TYPE* of the *CREATE-FILE* command).

2.2.2 Optimization of LEASY-LOADSI

In order to optimize the size of the SI files and the number of SI record splitting operations, the *PAD* operand in the file specification can be used.

The *SIZE* operand serves solely to optimize the runtime of the LEASY-LOADSI utility routine for large primary files.

2.2.3 Optimization of LEASY-MAINTASK

The **AIB* statement (define the size of the AIM file buffer) can be used to optimize the runtime of the user program.

The AIM buffer must be at least twice as long as the longest user record (for the precise minimum length see the description of the **AIB* statement). It should be at least long enough to accommodate, on average, all the AIM entries of a transaction in half of the buffer. This produces an average of one write operation to an AIM file for each transaction.

The paging rate for user programs is influenced by, amongst other things, the **MEM* statement (overall size of common memory CMMAIN). Normally **MEM=2* or *3* is sufficient. If higher values are required, check to make sure that the transaction facility is being used efficiently.

2.2.4 Optimization of LEASY-MASTER

This utility routine does not offer any direct facility for reducing the runtime of a LEASY application.

An indirect facility for doing so is offered by the shadow memory concept (switching to the next AIM generation), so that the difference between the original file and the shadow file does not become too great and so that any reconstruction run that takes place does not last too long. Another indirect possibility is to remove seriously delayed transactions from the current session.

2.2.5 Optimization of LEASY-RECONST

With the aid of shadow files it is possible to restart an application promptly after a volume has been destroyed.

The reconstruction run itself can be shortened by the following measures:

- the operand *PRI=M* in the **MOD* statement (shortest possible AIM list);
- the operand *SIU=N* in the **MOD* statement (suppression of the SI file update, after which the SI files must be updated by means of LEASY-LOADSI. As LEASY-LOADSI can write block by block, this method saves more time in the case of large volumes of data in the AIM file generation(s));
- the operand *UNL=N* in the **MOD* statement (avoiding *CLOSE/OPEN* cycles for the user files to be restored for each session entry of the AIM file);
- use of the *FILE* filter (**FIL* statement) to ensure that only those files which were actually destroyed are restored.

3 LEASY-CATALOG

The LEASY-CATALOG utility routine is used to set up, update and erase the LEASY catalog and its files.

The user ID under which the utility routine is run serves as the common catalog identifier for all files generated by LEASY, including the LEASY catalog. Existing files which are transferred to the LEASY file catalog can also be cataloged under another user ID.

Notes on input to the LEASY-CATALOG utility routine

The **CAT* statement must always be entered as the first statement of a statement sequence. If this statement is not entered, or is rejected due to an error, all subsequent statements (except **COM* and **END*) are rejected until the next **CAT* statement.

3.1 LEASY-CATALOG statements

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 2: Overview of statements (LEASY-CATALOG)

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

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

The statements are described below in alphabetical order.

*CAT Access LEASY catalog

The _*CAT_ statement accesses a LEASY catalog. It must always be the first statement specified. Two or more LEASY catalogs may be processed in one LEASY-CATALOG run. Each _*CAT_ statement accesses the LEASY catalog specified therein.

A LEASY catalog is an ISAM file bearing the name

```
:catid:$userid.file-catalog.LEASYCAT
```

The file is protected by an internally assigned DMS write password.

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

catid

Catalog identifier of the public volume set of the LEASY catalog, consisting of up to four characters enclosed in colons.

If *catid* is omitted, LEASY uses the catalog identifier of the default public volume set with the user ID under which LEASY-CATALOG was started.

file-catalog

Logical LEASY catalog name (≤ 8 characters, comprising letters, the special characters $, #, @ and digits. The first character must not be a "$"). This catalog name is part of the DMS file name of the LEASY catalog and of all files defined in that catalog.

TYP[E]

This operand specifies what is to happen to the LEASY catalog.

=N[EW]

A new LEASY catalog is created.

=O[LD]

An existing LEASY catalog is updated.

=E[RASE]

An existing LEASY catalog is to be erased. A check is made to see if there are still files entered in the catalog.

- The catalog is erased if it is empty.
- Message *LEA2118* is issued if the catalog still contains entries:

```
LEASY DIRECTORY NOT EMPTY
DO YOU REALLY WANT TO ERASE NON EMPTY
LEASY DIRECTORY (&00)? (Y=YES;N=NO)
```

Response: Y[ ] The catalog is erased.
Any user files are retained.

N[ ] The catalog is not erased.

Any BIM files that may exist and the LEASY status file are erased.

PAS[SWORD]

Specification of a catalog password when creating a LEASY catalog (*TYP=N*). This password must then be specified under *PAS* each time the catalog (*TYP=O* or (*TYPE=E*) is used in this utility routine. This password may also be interrogated when the LEASY-MASTER utility routine is used.

Passwords may be up to 4 bytes long:

C’c...c’ where *c* is 1 to 4 printable characters.

X’h...h’ where *h* is 1 to 8 hexadecimal digits.

NEW[PASSWORD]

Changes the password entry for an existing LEASY catalog (*TYP=O*). If a password has already been assigned, this should be specified under *PAS*. If an old password is to be deleted, then *NONE* should be specified.

INF[ORMATION] Specifies whether or not information on the respective LEASY catalog is to be output.

This operand may only be specified together with *TYP=O* (Default value).

=Y[ES] Date and number of the last LEASY session generated by LEASY-MAINTASK, the current values of *CPC* and *CPS* and also the entries in the *CID* operand. In addition, the value of the *ROM* operand which is currently preset is displayed.

=N[O] No information will be output by the *CAT statement.

The operands *VOL* and *DEV* are permitted only in conjunction with the operand *TYP=N*. They allow a LEASY catalog to be created on a private disk. Both operands must be specified if this is required.

VOL[UME]=vsn Volume serial number of the disk required.

DEV[ICE]=device Specification of disk storage unit type *device* (see the *CREATE-FILE* command in the "Commands, Volume 1 - 5" manuals).

CID[ ] Specifies whether or not the catalog identifier *:catid:* is to be included in the file name supplied to DMS for all user files. This operand is effective both in the LEASY runtime system and in the utility routines.

=Y[ES] Complete file names, including the catalog identifier, are always supplied to DMS in the form

```
:catid:$userid.file-catalog.file
```

LEASY can access user files in various DMS catalogs.

=N[O] The file names are supplied to DMS without a catalog identifier. This also applies if BIM or AIM entries containing the catalog identifier are processed during a warm start or a restore run. The following should be noted:

All LEASY system and user files must be linked to the DMS catalog whose catalog identifier *:catid:* is the default value for DMS. It is thereby possible to transfer LEASY applications independently of the catalog identifier.

CPC[ ] Defines names for copies of those files (shadow files) whose DMS name conforms to the LEASY name structure (`:catid:$userid.file-catalog.file`).

=:catid1: Catalog identifier of the public volume set that is to replace the catalog ID in the DMS name. The catalog identifier comprises up to 4 characters enclosed in colons. This operand is permitted only for users working with MPVS (multiple public volume sets).

$userid1.

User ID that is to replace the user ID in the DMS name.

copycat Name of the catalog in the name of the copy file; ≤ 8 characters; character set as for *file-catalog*.

=(<u>NO</u>) No name is defined or a predefined name is deleted.

Depending on whether *catid1* and/or *userid1* and/or *copycat* are specified, the *CPC* values replace the corresponding parts of the DMS name of the original file in order to form the DMS names of the shadow files.

CPS[ ] Defines names for copies of files (shadow files) whose DMS name does not conform to the LEASY name structure.
These are all foreign files, all temporary files and all master files which were given an explicit DMS name with the **FIL* statement.

CPS applies to all files if no *CPC* operand has been specified.

=:catid2: Catalog identifier of the public volume set. The catalog identifier comprises up to 4 characters enclosed in colons.

$userid2. User ID.

suffix Character string; ≤ 8 characters, character set as for *file-catalog*, hyphen also permitted.

=(<u>NO</u>) No name is defined or a predefined name is deleted.

The name of the shadow file is formed in such a way that if *catid2* and/or *userid2* is specified, the relevant parts of the DMS name of the original file are replaced. If specified, *suffix* is also attached to the DMS file names.

Examples of CPC and CPS:

LEASY catalog	LEACAT
User ID	USERID
CAT statement	CAT LEACAT,CPC=COPYCAT,CPS=$USER2.SAVE

Original file name	Shadow file name
$USERID.LEACAT.FILE	$USERID.COPYCAT.FILE
$USERID.FILENAME	$USER2.FILENAME.SAVE
$USER2.A.B.C	$USER2.A.B.C.SAVE

Two cases are supported in the multiprocessor environment, depending on the name structure of the copy file names:

- Original and shadow file are on the same processor (although the individual original files can be on different processors).
- All shadow files are on a particular processor (which must be defined with *CPC/CPS* by specifying *catid1/catid2*), regardless of where the original files are located.

OLDL[OCATION] / NEWL[OCATION]

The *OLDL* and *NEWL* operands of the **CAT* statement can be used to replace a catalog ID and/or user ID in the file name entries in the LEASY catalog with another *CATID/USERID*. Any *CATIDs* and/or *USERIDs* contained in the user data are not converted, they remain unchanged.

This function can be used if the LEASY catalog “**moves**“ together with the corresponding user files to another PVS and/or to another ID, i.e. the catalog file is copied to another pubset and/or another ID.

The file format (BLKCTRL property) of the file concerned must thereby remain unchanged, i.e. it is possible, for example, to copy from NK2 to NK2 or from keyed to keyed. Otherwise, the LEA.CONVERT utility routine must be used.

A prerequisite for this is that the catalog move, together with all files concerned is already **completed**. LEASY itself does **not** create the catalog and user files on the new PVS/user ID, the **user** must do this himself.

Contents are changed in the catalog file (ISAM). However, since the file name entries already have the maximum length (54 characters) in the files, lengthening or shortening a path name will not cause any moves.

Path names of shadow files are not converted because they are not in the catalog. This only contains the rules for forming names, that can be changed with *CAT ...CPC* or *CAT ..CPS*.

Path names of AIM files are not converted, which means that after the move nothing can be reconstructed from the former AIM generations. Therefore, the original files must be saved or the shadow files must be updated immediately before the move.

CATID and/or *USERID* are only changed if the BIM files belonging to the catalog are all **empty**, i.e. nullified or **non-existent**.

The changes made by the function are logged to SYSLST. It is recommended to make a **backup** of the catalog before changing the path components, in order to be able to reverse any unwanted changes after evaluating the SYSLST log.

OLDL[OCATION] Specifies the path components to be changed.

=:catid1: Is the catalog ID that is to be changed.
If *catid1* **is not** specified and a value *catid2* is specified in *NEWL*, the *catid2* specified under *NEWL* is inserted in all path names that contain **no** catalog ID.

$userid1 Is the user ID that is to be changed.
If *userid1* **is not** specified and a value *userid2* is specified in *NEWL*, the *userid2* specified under *NEWL* is inserted in all path names that contain **no** user ID.

NEWL[OCATION] Specifies the new path components

=:catid2: Is the catalog ID that is to replace the *catid1* specified in *OLDL*.
If *catid2* **is not** specified and a value *catid1* is specified in *OLDL*, the catalog ID is removed from all path names that **contain** the *catid1* specified in *OLDL*.

$userid2 Is the user ID that is to replace the *userid1* specified in *OLDL*.
If *userid2* **is not** specified and a value *userid1* is specified for *OLDL*, the user ID is removed from all path names that **contain** the *userid1* specified in *OLDL*.

If no catalog D is to be changed, *catid1* and *catid2* can be omitted. The same applies for the user ID. Once all entries have been made, the *catid1/userid1* pair is changed into the *catid2/userid2* pair. Other combinations in the file entries remain unchanged.

Syntax errors cause the statement processing to be stopped and the statement is not executed. Entering spaces between *catid* and *userid* is interpreted as a syntax error.

Examples:

1. Pubset *:A:* is to be changed to pubset *:B:* in all path names:

   ```
   CAT <catalogname>,OLDL=:A:,NEWL=:B:
   ```

   Catid *:A:* is replaced with catid *:B:* in all path names in the LEASY catalog. All other catids remain unchanged.

2. Pubset *:C:* is to be inserted into all path names without a catid:

   ```
   CAT <catalogname>,NEWL=:C:
   ```

   Catid *:C:* is to be inserted into all path names in the LEASY catalog that do not contain a catid. Path names that do contain catids remain unchanged.

Once the function has been successfully executed, the changed entries are logged to SYSLST. The log contains the following information:

- path name before the change
- path name after the change
- location of the change (catalog name)
- date and time

Log layout example:

```
CHANGES  OF CAT-COMMAND BY  LEASY-CATALOG  AT   2001-01-31  12:00:45

FROM                         TO                          IN CATALOG

:5ONK:$XDL1DSAT.DATEI1      :B:$XDL1SLEA.DATEI1        LEATEST

....
```

If the conversion makes the file name too long, the following note is written in the “TO“ column:
“`>>>NO CONVERSION BECAUSE NEW FILENAME > 54 CHARACTERS`“

If an error occurs during output to SYSLST, this is reported but processing is not stopped.

ROM[]

This operand is used to reserve files for setting READ-ONLY mode using the *ROMS* function of the LEASY-MASTER utility routine. This mode is required to perform an online save.

i This specification applies for all existing and future files of the LEASY catalog concerned. If the *ROM* operand is not specified, the *ROM* specification of the last **CAT* statement applies for this LEASY catalog. If no **CAT* statement with a *ROM* specification has as yet been executed for this LEASY catalog, *ROM=NO* applies. The current value for the specified LEASY catalog can be determined by specifying *INF=Y*.

The **FIL* statement following the *ROM* operand enables settings which differ from this default to be specified for individual files.

=Y[ES]

All master files of the LEASY catalog with the ISAM or PAM access method (including any existing SI files) are reserved for setting READ-ONLY mode.

For other files contained in the LEASY catalog, *ROM=NO* is also set when *ROM=YES* is specified, and a corresponding message (*LEA2166*) is output.

=N[O]

All master files of the LEASY catalog with the ISAM or PAM access method (including any existing SI files) are not (no longer) reserved for setting READ-ONLY mode.

*COM Comment text

The **COM* statement enables comment lines to be inserted at any position in the statement sequence. It is used primarily with LEASY-CATALOG in procedures.

Operation	Operands
`[*]COM[MENT]`	`[text]`

text

Any text.

The text string specified in the *COM statement can be spread over several input lines by chaining the records. It has no syntactical or semantic significance.

*END Program termination

The **END* statement terminates the LEASY-CATALOG utility routine.

Operation	Operands
`[*]END[]`	---

*ERA Erase request

The **ERA* statement erases a file (*file* or *file.suffix*) or all instances of a model file group (*file.*). They are removed from the LEASY catalog and also from the DMS catalog if the *CLE* operand is not specified.

The model in a model file group can only be erased if no instances for this model exist, therefore existing instances must be explicitly erased beforehand.

If the **ERA* statement is used to erase files for which secondary indices are defined, the corresponding secondary index files are also erased. The *CLE* operand is also effective for secondary index files.

Operation	Operands

`[*]ERA[SE]`	`{ file`
	`file.`
	`file.suffix }`
	`[,PAS[]={ X'writepassword'`
	`C'writepassword'`
	`NONE }]`
	`[,CLE[]={ R[]`
	`C[] }]`

file

Logical file name of an existing master, foreign or temporary file, or of the model in a model file group.

> The name of the public volume set does not have to be specified explicitly in the case of MPVS.

file.

This partially qualified logical file name addresses all instances of a LEASY model file group.

file.suffix

Logical file name of an instance of a LEASY model file group.

PAS[SWORD]

A write password *writepassword* defined for this file must be specified here. The password can be up to 4 bytes long:

C’c...c’ where *c* represents 1 to 4 printable characters

X’h...h’ where *h* represents 1 to 8 hexadecimal digits.

CLE[AR]

This operand specifies whether the file as a whole or only its contents are to be erased from the LEASY and DMS catalogs.

If omitted, the file is erased both from the LEASY catalog and from the DMS catalog.

=R[ECORD] Only the contents of the specified files are to be erased; the catalog entries are retained.

ℹ LEASY-CATALOG does not make entries in any existing AIM files.

=C[ATALOG] The file, and an SI file if one already exists, is erased from the LEASY catalog. The files themselves are retained as DMS files. This export function is possible for all data files.

*FIL File specification

The **FIL* statement is used either to add a new file to the LEASY catalog, or to modify one or more files and their attributes. For new files a file's DMS attributes can also be transferred from a TFT entry using *MOD=L*.

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

(part 1 of 3)

```
           ⎧C'writepassword'⎫
[,PAS[]=⎨X'writepassword'⎬]
           ⎩NONE             ⎭

           ⎧N[]⎫
           ⎪U[]⎪
[,MOD[]=⎨   ⎬]
           ⎪C[]⎪
           ⎩L[]⎭

           ⎧N[]⎫
[,LOC[]=⎨   ⎬]
           ⎩Y[]⎭

[,RTF[]=(rpos,rlen)]

           ⎧(keyname,-)
           ⎪
[,KEY[]=⎨                   ⎧[iub,]pos,len[,dist]         ⎫
           ⎪([rep*]keyname,⎨[iub,](pos,len[,dist]),...⎬
           ⎩                   ⎩(iub,pos,len[,dist]),...    ⎭
                                                                ⎫
                                   ⎧recordtype       ⎫        ⎬]...
       [,[dup][,[upd]]][,RTP=⎨(recordtype,...)⎬])    ⎪
                                   ⎩NONE              ⎭        ⎭

               ⎧ISAM⎫
               ⎪SAM ⎪
[,FCBTYPE=⎨     ⎬]
               ⎪PAM ⎪
               ⎩DAM ⎭

               ⎧PAMKEY⎫
               ⎪DATA   ⎪
[,BLKCTRL=⎨DATA2K⎬]
               ⎪DATA4K⎪
               ⎩NO     ⎭

           ⎧*STD        ⎫
[,PLK[]=⎨             ⎬]
           ⎩poollinkname⎭

           ⎧*POOL       ⎫
[,SLK[]=⎨*STD        ⎬]
           ⎩poollinkname⎭

[,RECSIZE=recsize]

           ⎧N[]⎫
[,SHA[]=⎨   ⎬]
           ⎩Y[]⎭
```

(part 2 of 3)

	⎧Y[]⎫ [,ROM[]=⎨ ⎬] ⎩N[]⎭

(part 3 of 3)

When _MOD=NEW_ is specified, in addition to the operands listed in this syntax, all of the other operands of the DMS macro _FILE_ which are required can be specified in the _*FIL_ statement.

These operands are transferred to a _FILE_ macro issued internally, but are otherwise ignored by the LEASY-CATALOG utility routine.
Incorrect operand specifications are not detected by the LEASY-CATALOG utility routine but are passed on to the BS2000 Data Management System; thus it is DMS which first marks them as being errored.

If the desired file attributes deviate from the DMS default value, appropriate operand specifications are required. This does not apply to the _LINK, OVERLAP, WROUT_, _SHARUPD_ and _DUPEKY_ operands, which would have no effect. In the case of foreign files, no additional DMS operands are permitted.

When _MOD=LINK_, only the LEASY attributes _LEA, AIM, BIM, SIO, WRP, RDP, PAS, SHA, MOD, LOC, RTF, KEY_ and _SLK_ may be specified in the _*FIL_ statement. All DMS attributes must be entered in a TFT. This separation into LEASY attributes and DMS attributes is necessary when DMS operands are used that can not be specified in the _*FIL_ statement (also see section “Creating a new file” on page 51f).

Description of the operands

file

Logical file name of a master, foreign or temporary file or of the model in a model file group, which must also be specified in the _DB_ operand of the LEASY call (≤ 8 characters comprising letters, the special characters #, @, $ and digits).
Wholly numeric file names are not permitted because this name is also used internally as the file link name and DMS rejects file link names which are numeric.

file.suffix

Logical file name of an instance of a model file group (≤ 29 characters). The character set is the same as for _file_ except for periods and hyphens, which are permitted in the suffix. The suffix length must not exceed 20 characters.

If secondary keys are used, the DMS name of the associated SI file must not exceed 54 characters.

file.

This partially qualified logical file name (≤ 9 characters including the period) addresses all instances of a model file group. This specification is permitted only when *MOD=U* is specified.

NAM[E]=[:catid:][$userid.]filename

Freely selectable DMS file name. This operand may only be used for master files.

With *MOD=N*, the file is created with this name. With *MOD=C*, an existing file of this name is transferred to the LEASY catalog.

If a DMS file name is specified explicitly with user ID (*userid*) and/or catalog identifier (*:catid:*), the DMS file name including user ID and/or catalog identifier is stored in the LEASY catalog.

If a new file is to be generated under a foreign user ID, a *CREATE-FILE* command must previously have been given in this user ID so that the file is known to the DMS file catalog.

MPVS support:

The catalog identifier (:*catid*:) of the appropriate public volume set must be specified if files are to be distributed over various public volume sets.

Using LEASY in MRS:

If a catalog identifier (:*catid*:) is specified, the DMS name can be up to 51 characters long.

LEA[SYTYPE]

LEASY file type.

=S[ ]
Stands for master file.

=M[ ]
Stands for model file.

=T[ ]
Stands for temporary file.

=F[ ]
Stands for foreign file.

AIM[ ]

Specification for maintaining the file in the AIM file.

ℹ In the case of temporary files or foreign files only *AIM=N* (default) is permitted.

=N[O]
The file is not to be maintained in the AIM file.

=Y[ES]
The file is to be maintained in the AIM file. This specification is only effective if the value *A* or *Y* is specified in the **LOG* operand when generating the main LEASY task with the LEASY-MAINTASK utility routine.

=(Y[ES],A[UTOMATIC])

The file is to be maintained in the AIM file. The associated shadow file is to be kept automatically. This specification is only effective if the value _Y_ is specified in the _*AUT_ operand when generating the main LEASY task with the LEASY-MAINTASK utility routine.

=R[EDUCED] The file is to be maintained in the AIM file. The modified records are to be stored in the AIM file in reduced form. This specification is only effective if the value _A_ or _Y_ is specified in the *LOG operand when generating the main LEASY task with the LEASY-MAINTASK utility routine.

=(R[EDUCED],A[UTOMATIC])

The file is to be maintained in the AIM file. The modified records are to be stored in the AIM file in reduced form. The associated shadow file is to be kept automatically. This specification is only effective if the value _Y_ is specified in the _*AUT_ operand when generating the main LEASY task with the LEASY-MAINTASK utility routine.

BIM[ ] Specification for maintaining the BIM files.

=Y[ES] Before-image saving is required. This specification is only effective if the value _B_ or _Y_ is specified in the _*LOG_ operand when generating the main LEASY task with the LEASY-MAINTASK utility routine.

=N[O] Before-image saving is not required.

IOPERF Defines the performance attribute of the primary file for I/O processing.

=VERY-HIGH Highest performance priority. If possible, the entire file is held permanently in a cache.

=HIGH If possible, the file is processed via a cache.

=ST The file is not processed via a cache.

=USER-MAX The file is assigned the highest performance attribute permitted for the user ID.

IOUSAGE Specifies the I/O operation to which the performance attribute relates.

=RDWRT The performance attribute relates to read and write operations.

=WRITE The performance attribute relates to write operations.

=READ The performance attribute relates to read operations.

SIOPERF[ ] Defines the performance attribute of the SI file for I/O processing. The attributes apply to read and write operations.

=VERY-HIGH — Highest performance priority. If possible, the entire SI file is held permanently in a cache.

=HIGH — If possible, the SI file is processed via a cache.

=STD — The SI file is not processed via a cache. New SI files are assigned *STD* as the default value. Existing SI files retain their assigned performance attributes.

=USER-MAX — The SI file is assigned the highest performance attribute permitted for the user ID.

WRP[ ], RDP[ ] — Specify a new write or read password or delete an old write or read password if *NONE* is entered.

Passwords may be up to 4 bytes long:

C’c...c’	where *c* represents 1-4 printable characters
X’h...h’	where *h* represents 1-8 hexadecimal digits.

The read and write passwords (*RDP*, *WRP*) are maintained in the LEASY catalog, but are at present not interpreted by the LEASY runtime system (*LEACONX* module). They have nothing to do with the DMS operands and should not be mistaken for them.
DMS passwords are not assigned.

The password assigned under *WRP* protects the file against updating and erasing in the LEASY-CATALOG utility routine and must be specified in the *PAS* operand for the appropriate commands (**FIL* and **ERA*).

PAS[SWORD] — If a write password has been specified for the file, it must be entered in this operand each time the file attributes are modified with the **FIL* statement.

MOD[E]=N[EW] — The file must be newly created and stored in the LEASY catalog. Also see section “Creating a new file” on page 51f.

=U[PDATE] — The attributes of a file or files are to be updated. Also see section “Modifying the attributes of existing files” on page 56.

=C[ATALOG] — An existing file is to be transferred as a master file into the LEASY file catalog (import function).

=L[INK] — A new file is to be added to the LEASY catalog.
The DMS parameters for this file must be located in a TFT. Also see section “Creating a new file” on page 51f.

LOC[K] — Locks and unlocks a file for subsequent LEASY sessions. This operand has no effect on a main task already loaded.

=N[O]
The file is to be unlocked or not locked.

=Y[ES]
The file is to be locked for subsequent LEASY sessions. Then, when a session is started with the LEASY-MAINTASK utility routine, the file is not transferred to the common memory CMMAIN.

RTF[ ]=(rpos,rlen)
This operand defines a record type field in a record.

rpos
Position of the 1st character of the record type field. The 4-byte record type field must be taken into account when the variable record format is used.

rlen
Length of record type field in bytes; range of values: 1 ≤ *rlen* ≤ 255. The record type field can be part of the primary key, but it does not have to be (ISAM only).

KEY[DEFINITION]
Operand for creating, inserting or deleting secondary key definitions for ISAM, DAM and PAM files.

Each file can have a maximum of 255 secondary keys. See section “Handling of secondary index definitions” on page 57.

keyname
Logical secondary index name (≤ 8 characters, comprising letters, the special characters $, #, @ and digits).

-
The minus character following the SI name *keyname* causes this SI definition to be deleted from the LEASY catalog.

rep
Repetition factor for a multiple secondary key.

Range of values: 1 ≤ *rep* ≤ 32768 (default value: rep=1).

rep specifies the (maximum) number of occurrences of a secondary key which may be composed of key parts.

The number of secondary key occurrences within a data record may be less than the number specified by *rep* when records are of variable length. This is possible where the data record is not long enough to include all occurrences.

isb
Index suppression byte. If all secondary key bytes of a primary data record are identical to this byte, no secondary index pointer is created for this record when the secondary keys are generated, either by the LEASY-LOADSI utility routine or by the LEASY runtime system.

X’h’ *h* = 2 hexadecimal digits (1 byte) or

C’c’ *c* = 1 printable character (1 byte).

Definition of the position and length of a secondary key or secondary key parts:

pos Position of the first character of the secondary key or secondary key part. If the record format of the primary record is variable (*RECORD-FORM=V*), the 4-byte record length field is counted as being part of the record.

len Length of the secondary key or a part thereof.
Range of values: $1 \leq len \leq 253$. Also valid is:

$$\text{keylenprim} + \sum_{i=1}^{k} \text{len i} \leq 254$$

k is the number of key parts. (This also applies to the sum of the key part lengths if the secondary key comprises several parts.)

If the secondary key consists of only one part, the parentheses enclosing *pos* and *len* are omitted. If several key parts are defined, the specification of each part must be placed in parentheses.
A maximum of 253 key parts may be defined per secondary index.

The total number of all key parts of a file must not exceed 512.

dist Distance between the (n)th and (n+1)th occurrence of a key part field belonging to a multiple secondary key.

If *dist* is not specified for a key part, then *dist=len*.

$0 \leq dist < 32768$ is valid as a rule.

The specification *dist=0* for a key part causes the same contents to be taken for each occurrence of the key part field.

dup Value defining whether duplicate secondary keys are permitted:

Y Duplicate key values are permitted.

N Duplicate key values are not permitted for different primary records.

upd This value defines whether the secondary index pointers in the SI file are to be updated by the LEASY runtime system:

Y Changes made in the primary file are also to be effected in the secondary index pointers.

N Secondary index pointers are retained despite changes made to the primary file.

If both are specified, *dup* must precede *upd*.

If the value *dup* is omitted, its comma must still be entered to ensure that *upd* is in the correct position.

See page 57ff for examples of secondary index definitions.

RTP

This operand restricts the validity of the secondary key definition. The secondary key is valid only if the record type field defined with the *RTF* operand has the contents (*record type*) specified here. If the *RTP* operand is omitted, the secondary key definition always applies, irrespective of the contents of any defined record type field.

The *RTP* operand is only permitted if a record type field has already been specified for the file with *RTF* or is specified at the same time.

recordtype

Record type field value or left-justified record type field group value.

3 representations are possible:

C'c...' 1 to *rlen* printable characters.

Apostrophes must be represented as double quotes.

X'nn..' 2 to 2**rlen* hexadecimal digits.

If the length of the record type string is less than the length of the record type field *rlen*, the string is known as a record type field group value.

The secondary key definition applies to all records of which the record type field contains, left-justified, either one of the record type field values or one of the record type field group values specified here.

NONE With records of variable length, it is possible that the record type field defined by the *RTF* operand is situated beyond the record length for some of the records and is, therefore, undefined.
NONE means that a secondary key applies whenever the record type field or part of it is not within the record.

FCBTYPE

Specifies the BS2000 access method for this file. Like all other permissible DMS operands, this operand has the DMS format.

=<u>ISAM</u>

for the indexed-sequential access method.

=SAM

for the sequential access method.

=PAM

for the primary, block-oriented access method, but with *BUFFER-LENGTH* and *RECORD-SIZE* as fixed file attributes.

=DAM

for the direct access method.

BLKCTRL defines the file format for supporting the various disk types.

=PAMKEY K file ormat containing the block-specific management information in the PAM key. With ISAM files, the *PLK* and *SLK* operands are not evaluated.

=DATA File format in which the block-specific management information is available in each data block. On NK4 pubsets this has the same meaning as for *DATA-4K*.

=DATA2K NK-ISAM file on K or NK2 pubsets.

=DATA4K NK-ISAM file on K, NK2 or NK4 pubsets.

=NO File format in which no block-specific management information is stored. With ISAM and SAM files, DMS converts this specification to *BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK*.

The following table shows the default *BLKCTRL* values assigned to the files created by LEASY, as a function of disk type and class2 option *BLKCTRL*.

Disk type →		**K**		**NK2**	**NK4**
CL2 option BLKCTRL →		**PAMKEY**	**NONKEY**	**PK/NK**	**PK/NK**
File type ↓	**FCB type↓**				
Primary file	DAM ISAM PAM SAM	PAMKEY PAMKEY PAMKEY PAMKEY	NO DATA-2K NO DATA	NO DATA-2K NO DATA	NO DATA-4K NO DATA
SI file	ISAM	PAMKEY	DATA-2K	DATA-2K	DATA-4K
Catalog	ISAM	PAMKEY	DATA-2K	DATA-2K	DATA-4K
BIM file	PAM	PAMKEY	NO	NO	NO
AIM file	PAM	PAMKEY	DATA	NO	NO
Status file	ISAM	PAMKEY	DATA-2K	DATA-2K	DATA-4K

PLK Data on the ISAM pool for the LEASY file.

The operand is interpreted only for NK-ISAM primary files.

=<u>*STD</u> The specified LEASY file is to use the global standard ISAM pool *$SYSn* (01 ≤ *n* ≤ 16).

=poollinkname

ISAM pool link name for the LEASY file; up to 8 characters.

The names *$TASKn* (01 ≤ *n* ≤ 16) and *$SYSn* (01 ≤ *n* ≤ 16) must not be used, since they are reserved for the standard pools.

SLK

Data on the ISAM pool for the LEASY-SI file. The operand is only interpreted if the SI file has the NK-ISAM format.

=*POOL

The same ISAM pool as for the primary file is to be used for all SI files.

=*STD

The specified SI file is to use the global standard ISAM pool *$SYSn* (01 ≤ *n* ≤ 16).

=poollinkname

ISAM pool link name for the SI file; up to 8 characters.

The names *$TASKn* (01 ≤ *n* ≤ 16) and *$SYSn* (01 ≤ *n* ≤ 16) must not be used, since they are reserved for the standard ISAM pools.

> ℹ The pool link names of the *PLK/SLK* operands are only processed if *BLKCTRL=DATA* or *BLKCTRL=NO* is set. If *BLKCTRL* retains the default value *PAMKEY*, *PLK/SLK* are checked for syntax errors but are not interpreted, and a warning is issued.

If no appropriate **POO* statement with the pool link name of the *PLK/SLK* operands is specified, LEASY assumes that **user** ISAM pools are employed. In this case, the users are responsible for creating and managing their own ISAM pools (*CREATE-ISAM-POOL* and *ADD-ISAM-POOL-LINK*).

Different ISAM pools can be used for primary files and SI files.

RECSIZE=recsize

Specification of the (maximum) record length. The rules laid down by BS2000 DMS apply to ISAM and SAM files.

recsize ≤ BLKSIZE	for PAM and DAM files
recsize ≤ BLKSIZE - 12	for NK files
recsize = BLKSIZE	default value

If *recsize* exceeds one PAM page (2048 bytes), an appropriate *BLKSIZE* value must be specified.
The primary and secondary assignments (*SPACE* value) must be greater than or equal to the agreed buffer length.

With fixed record length, *RECSIZE* is a file property and causes the logical data blocks of LEASY to always be read and written in the specified length.

With variable record length, *RECSIZE* defines *RECSIMAX* as a LEASY property for the maximum length of the logical data blocks. *RECSIZE* may only be subsequently changed in this case. A change only comes into effect after the LEASY-MAINTASK is restarted.

SHA[RE]
Specifies whether or not the file is shareable.

=N[O]
The file is not shareable. This means that this file can only be accessed from the user ID under which LEASY-CATALOG is running.

=Y[ES]
The file is shareable.

ROM[]
This operand is used to reserve the file for setting READ-ONLY mode using the *ROMS* function of the LEASY-MASTER utility routine. This mode is required to perform an online save.

> This specification overwrites the catalog-wide default setting made using the **CAT* statement.
>
> If the *ROM* operand is not specified, the setting for the existing files remains unchanged and the *ROM* specification of the **CAT* statement applies for new files (*MOD=NEW*).

=Y[ES]
The file is reserved for setting READ-ONLY mode.

The *ROM=YES* specification is permitted only for master files with the ISAM or PAM access method. When other files are involved, this specification is rejected with the message *LEA2166* and *ROM=NO* applies.

=N[O]
The file is not (no longer) reserved for setting READ-ONLY mode.

*INF Information request

This statement causes an information block from the DMS and/or LEASY catalog to be output to SYSOUT.

Operation	Operands
[*]INF[ORMATION]	[{file file.suffix file.}] [,{A[] S[] C[] T[] P[] R[] L[] M[] O[]}]

file
: Logical file name of a master, foreign or temporary file, or of the model in a model file group.

file.suffix
: Logical file name of an instance of a LEASY model file group.

file.
: This partially qualified logical file name selects all the instances of a LEASY model file group.

The specification a file name is ignored if *M* or *O* is specified as the 2nd operand.

If no file name is specified, the information request refers to all files in the LEASY catalog previously selected with **CAT*.

Unless *L*, *M* or *O* is specified, the second operand is mapped to a BS2000 *FSTAT* macro. It controls the amount of information output per file.

In addition to the file name, the following information is output with the respective specification:

A[ ]	all information
S[ ]	standard information
C[ ]	catalog information

In addition, with SI files, specifying the above values means that every key displayed will be followed in parentheses by the internal SI number of the secondar key maintained in the catalog.

T[ ]	file and volume information
P[ ]	password information
R[ ]	number of PAM pages occupied

The attributes stored in the LEASY catalog are also listed in addition to the *FSTAT* macro output. For NK-ISAM files the secondary key attributes are also output in the format of the *SHOW-INDEX-ATTRIBUTES* command.

L[ ] If the *L* operand is specified, only the attributes stored in the LEASY catalog are output.

If *BLKCTRL=DATA/NO* is specified in the **FIL* statement, the LEASY catalog information output also includes the ISAM pools of the primary and SI files unless standard ISAM pools are used.

Any value for the ISAM pool link name of the SI file is merely an image of the LEASY catalog information and does not necessarily mean that an SI file exists.

The values *M* and *O* enable files to be selected for output on the basis of particular criteria. If a file name is specified in conjunction with this specification, it is ignored.

M[ ] Outputs all files for which an inadvertent loss of information in the AIM data set can occur under the following conditions:

- Automatic reconstruction of shadow files is being used, and
- an AIM file generation is released after automatic reconstruction.

All files are output for which *AIM=Y* or *AIM=R* was specified in the **FIL* statement, but only if the catalog contains at least one file with the following attributes:

- LEASY file type is not the same as the model file and
- automatic recording of the shadow file (*AIM=(Y,A)* or *AIM=(R,A)* in the **FIL* statement).

O[ ] Outputs all files which are reserved for setting to READ-ONLY mode. If no files are reserved, the message *LEA5108* is issued.

*PIN Output of information on ISAM pools

The **PIN* statement causes data on the ISAM pools defined in the LEASY catalog to be output to SYSOUT.

Operation	Operands

`[*]PIN[FORMATION]`	`[poolname]`
	`[,A[LL]]`

poolname Name of the ISAM pool.

The pool name is identical with the *poollinkname* specified for the ISAM pool in the **FIL* statement. If the pool name is specified, the associated pool management record is output. If the pool name is missing, the management records of all ISAM pools defined in the LEASY catalog are output.

A[LL] The logical file names of the files connected to the ISAM pool *poolname* are output in addition.
If just *ALL* (without *poolname*) is specified, the management records of all ISAM pools defined in the LEASY catalog plus all logical file names of the connected files are output.

i SI files are output only if they actually exist, i.e. if an SI key was defined with the *KEY* operand in the appropriate **FIL* statement.

*POO Define ISAM pool

The **POO* statement defines the attributes of a LEASY-specific ISAM pool which is used in connection with NK-ISAM (see the partner manual "LEASY (BS2000) - Program Interface and Strategies"). The ISAM pools defined with this statement are created and managed by LEASY.

If either the default value or no pool link name at all is specified in the **FIL* statement for NK-ISAM files, the **POO* statement is not required because the standard ISAM pools *$SYSn* (01 ≤ *n* ≤ 16) are used in this case.

If a pool link name does exist in the **FIL* statement but no **POO* statement is issued, LEASY assumes that **user** ISAM pools are employed. It is then the users' responsibility to create and manage their own ISAM pools (*CREATE-ISAM-POOL* and *ADD-ISAM-POOL-LINK*).

Operation	Operands
[*]POO[L]	poollinkname [,CAT[ID]={*DEF[AULT] catid}] [,SIZ[E]={*STD size}] [,MOD[E]={N[EW] U[PDATE] E[RASE]}] [,PCR[EATION]={R[UNTIME] M[AINTASK]}]

poollinkname ISAM pool link name.

LEASY creates the ISAM pool using the same name (pool name = pool link name). *poollinkname* corresponds to the name specified in the **FIL* statement for the *PLK* and *SLK* operands.

CAT[ID] Defines the pubset on which the ISAM pool is to be created.

=catid The ISAM pool is created on the host computer on which this pubset exists.
catid must be not more than **four** characters **without** enclosing colons.
catid may be regarded as part of the name, i.e. different *catid* values always identify different ISAM pools.

=*DEFAULT
The ISAM pool is created on the default pubset of the user identification (_JOIN_ entry).

SIZ[E]
Defines the size of the ISAM pool to be created by LEASY.

=*STD
The ISAM pool is created with the standard size.

=size
Size of the ISAM pool in PAM pages.

The permissible range is shown in the description of the CREPOOL macro in the "DMS Macros" manual. The calculation of the size of ISAM pools is explained in the "Introductory Guide to DMS" manual.

MOD[E]
Defines how the pool information record is to be handled.

=N[EW]
A new pool information record is generated and entered in the LEASY catalog. If a pool information record already exists in the LEASY catalog, an error message is output and the _*POO_ statement is ignored.

=U[PDATE]
An existing pool information record in the LEASY catalog is altered. Operand values which are not specified remain unaffected. If the pool information record to be updated does not exist in the LEASY catalog, an error message is output and the _*POO_ statement is ignored.

=E[RASE]
The pool information record is removed from the LEASY catalog. If the pool information record to be erased does not exist in the LEASY catalog, an error message is output and the _*POO_ statement is ignored. If operands other than _MOD=E_ are specified, they are checked for syntax errors but ignored during further processing.

PCR[EATION]
Pool creation mode. This defines when and by whom the ISAM pool is to be created.

=R[UNTIME]
LEASY creates the ISAM pool upon the first OPEN issued for the file in the runtime system. Only ISAM pools which are actually needed are created. The ISAM pools are created with _CREATION-MODE=ANY_.

=M[AINTASK]
LEASY creates the pool upon startup of the main task. All ISAM pools for which _PCR=M_ was defined are created. The ISAM pools are created with _CREATION-MODE=NEW_. If the ISAM pool already exists, the main task is terminated and an error message is output.

3.2 Creating a new file

When a file is first created (**FIL* statement, *MOD=N*) its name

```
:catid:$userid.file-catalog.file
```

(or the DMS name explicitly specified) and the other DMS attributes in the **FIL* statement are cataloged. The DMS catalog must not contain any other attributes previously specified for this file.

If a DMS name has been specified in the *NAM* operand of the **FIL* statement, and its user ID is different to that under which LEASY-CATALOG is run, the following BS2000 command must be issued in the user ID specified in the *NAM* operand before the utility routine can be started:

```
/CREATE-FILE FILE-NAME=dmsname,PROT=(USER-ACCESS=*ALL-USERS)
```

If secondary indices are required for the file *dmsname*, the following command must also be issued:

```
/CREATE-FILE FILE-NAME=dmsname-SI,PROT=(USER-ACCESS=*ALL-USERS)
```

The file is opened and closed by the **FIL* statement; the file attributes are thereby entered in the DMS catalog. This does not apply to foreign files. An entry is made for them in the LEASY catalog, but no DMS file is created.
An additional *CAT* macro is issued for specification of the *SHA* operand.

The values of the *LEA, AIM, BIM, RDP, LOC, WRP, FCBTYPE, RECSIZE* and *PAD* operands of the **FIL* statement are stored in one record in the LEASY catalog.

If secondary indices are defined, an SI file bearing the name

```
$userid.file-catalog.file-SI or dmsname-SI
```

together with the appropriate file attributes is created on the same volume on which the primary file index is located, and the secondary index definitions are stored in the LEASY catalog.

3.2.1 Creating new model file groups

When creating model file groups, the model with its logical file name and attributes should first be entered via the **FIL* statement:

```
*FILE file, LEASYTYPE=M, AIM=Y, KEY=(..),...
```

The individual instances can then be entered:

```
*FILE file.suffix1,...
*FILE file.suffix2,...
etc.
```

Further attributes that are specified in these **FIL* statements in addition to the name of the instances are entered. File attributes that are not defined are taken from the model. DMS operands other than the *SHA* operand (corresponds to *USER-ACCESS*) are rejected.

These new instances are generated by a DMS macro on the volume containing the model:

```
COPY-FILE FROM-FILE=file-catalog.file,
           TO-FILE=file-catalog.file.suffix,PROTECTION=*SAME.
```

They then have the same DMS attributes as the model.

If secondary indices are defined, a secondary index file is also created for each instance in the model file group:

```
COPY-FILE file-catalog.file-SI,
          TO-FILE=file-catalog.file.suffix-SI,PROTECTION=*SAME.
```

3.2.2 Specifying the file attributes in DMS

The **FIL* statement (*MOD=N*) of the LEASY-CATALOG utility routine is fundamentally identical with the DMS macro *FILE*, extended by various LEASY-specific operands.

The LEASY-specific operands are evaluated by LEASY-CATALOG itself, while the other operands are forwarded to DMS. To do this, LEASY-CATALOG uses the *FILE* macro of the lowest BS2000/OSD version in which LEASY can run. As a result, operands added in subsequent BS2000/OSD versions can only be specified in corresponding system commands and not in the **FIL* statement. In the main, the operands of the *FILE* macro have been given new names and distributed over the following commands:

/CREATE-FILE

/ADD-FILE-LINK

Distribution of operands for specifying file attributes

If DVS operands which cannot be specified in the **FIL* statement are to be used, the value *L* must be used in the *MOD* operand of the **FIL* statement. The specification of operands then can be distributed over the commands */CREATE-FILE* and */ADD-FILE-LINK* and the **FIL* statement.

Note the following for *MOD=L*:

Before a **FIL* statement can be issued with *MOD=L* in LEASY-CATALOG, the following steps must be executed using DMS commands:

- */CREATE-FILE*: generate a DMS catalog entry.
- */ADD-FILE-LINK*: define a link name and entry in the **t**ask **f**ile **t**able (TFT).

LEASY evaluates the contents of the TFT and transfers the necessary data to the LEASY file catalog.

The */ADD-FILE-LINK* command and the **FIL* statement are linked via the logical file name as a link name. If no TFT entry is found via the logical file name, error message *LEA2158* is output. If the DMS catalog entry is not found, DMS issues error code *0533*.

In the case of foreign files or instances of a model, *MOD=L* is not possible and is rejected with error message *LEA2157*.

If the file attributes of temporary files or models are specified by means of */CREATE-FILE* and */ADD-FILE-LINK*, the file name must comply with the LEASY name structure. In the event of an error, error message *LEA2155* is output and the statement is not accepted.

In the case of model instances, all DMS file attributes are transferred from the model. For foreign files, the file attributes are first determined in the runtime system. Only the FCBTYPE (due to *FCB=DAM*) is entered in the LEASY catalog. This type can be specified in the **FIL* statement.

If *MOD=L* is specified, the **FIL* statement must not contain any DMS-specific operands; otherwise, a warning is issued (*LEA2152*) and the operand is ignored. Exceptions to this rule are the **FIL* operands *NAME* (=FILE-NAME), *FCBTYPE* (=ACCESS-METHOD) and *PLK* (=POOL-LINK), which are described below.

The LEASY-specific operands that can be specified are:
LEA, AIM, BIM, SIO, WRP, RDP, PAS, MOD, LOC, RTF, KEY, SLK.

DMS operands that can be specified in LEASY and/or DMS

- File name (NAM, FILE-NAME):

  A DMS file name that does not comply with the LEASY naming conventions can be specified both in DMS using */ADD-FILE-LINK* (*FILE-NAME* operand), and in LEASY using the *NAM* operand of the **FIL* statement. The following points must be taken into account:

  - If a file name without a *catid/userid* is entered in the *FILE-NAME* operand with */SET-FILE-LINK*, DMS supplements it with the relevant default values. Therefore, a file name without *catid/userid* that does not correspond to the default values can only be specified using the *NAM* operand of the **FIL* statement.
  - A file name that is specified in the NAM operand is transferred to the LEASY catalog without modification (e.g. without *catid, userid*).
  - If a file name is specified by both */ADD-FILE-LINK* and the **FIL* statement, the two names must match. They may differ only in the omission of *catid/userid* in the *NAM* operand. If there are any other differences, the error message *LEA2156* is output and the *FIL statement is not executed.
  - If LEASY is to create a file with the default name *catalogname.linkname*, the *FILE-NAME* operand must not be specified in */ADD-FILE-LINK*. In this case, the catalog entry must be generated by means of */CREATE-FILE*.

- File access method (FCBTYPE, ACCESS-METHOD):

  Since DMS does not recognize DAM files, it is permissible to specify *FCBTYPE=DAM* in the **FIL* statement. In this case, */ADD-FILE-LINK ACCESS-METHOD=*UPAM* must be selected in the associated command or left unspecified. Any other FCBTYPE (ACCESS-METHOD) that is known to DMS must be specified in */ADD-FILE-LINK*. The default value for *ACCESS-METHOD* is *ISAM*.

- ISAM pool link name (PLK, POOL-LINK):

  As before, the pool link name of an NK-ISAM file can be specified in the *PLK* operand of the LEASY **FIL* statement but it can also be specified in the *POOL-LINK* operand of */ADD-FILE-LINK*. If it is defined only in the **FIL* statement, LEASY continues to behave exactly as it did with *MOD=N*. On the other hand, the *POOL-LINK* operand is allowed only with some restrictions.

The following requirements must be met:

- Before calling LEASY-CATALOG, the user must generate the pool and pool link name using */CREATE-ISAM-POOL* and */ADD-ISAM-POOL-LINK*. Otherwise, DMS error 0D82 occurs and LEASY-CATALOG cannot open the file.
- If both operands (*PLK* in **FIL* and *POOL-LINK* in */ADD-FILE-LINK*) are specified, they must match. Otherwise, error message *LEA2154* is output and the **FIL* statement is not accepted.

Example

```
/CREATE-FILE -                                            (1)
/  FILE-NAME=catalog.file, -
/  SUPPORT=*PUB-DISK(SPACE=*RELATIVE(1000,100))

/ADD-FILE-LINK -                                          (2)
/  FILE-NAME=catalog.file, -                              (3)
/  LINK-NAME=file, -                                      (4)
/  ACCESS-METHOD=*UPAM -                                  (5)
/  RECORD-FORMAT=*FIXED(REC-SIZE=80) -                    (6)
/  BUFFER-LENGTH=STD(2)

/START-LEASY-CATALOG
*CAT catalog                                              (7)
*FIL file,MOD=L,FCBTYPE=DAM,AIM=YES                       (8)
*END
```

(1) The file is entered in the DMS catalog.

(2) A TFT entry is generated for the file.

(3) A reference to the catalog entry is created using the file name.
This can be omitted if the LEASY file name structure is used.

(4) The link name serves as a reference to the specifications in the **FIL* statement of LEASY-CATALOG.

(5) The LEASY access method DAM is mapped onto the DMS access method UPAM.

(6) DAM files have a fixed record length.

(7) The catalog to which the file is to be added is specified.

(8) The logical file name must match the link name in the */ADD-FILE-LINK* command.
The LEASY-specific operands are specified, including for the DAM files of the FCBTYPE.

For examples of **FIL* statements, see page 64ff.

3.3 Modifying the attributes of existing files

When modifying file attributes (*MOD=U*) only the following operands can be entered in the **FIL* statement:

AIM, BIM, WRP, RDP, PAS, SHA, PAD, LOC, RTF, KEY, PLK, SLK, RECSIZE, ROM

PAS indicates any password previously specified under *WRP*. *RDP* and *WRP* can be used to define new read and write passwords. The definition of the record type field (*RTF* operand) can only be modified if there are no pointers to secondary keys with a record type restriction for this index for the file in the secondary index file.

RECSIZE may only be changed for files with variable record length (*RECFORM=V*).

A **FIL* statement can modify file attributes as follows:

- for files: **FIL file* or **FIL file.suffix*
- for all the instances of a model file group: **FIL file*

The **FIL* statement cannot be used to modify the DMS attributes of a file (the operands *PAD* and *SHA* are exceptions).

The following should be noted when updating secondary key definitions:

- Only the keys specified are modified; all other keys remain unchanged.
- If the secondary index name has not yet been entered in the LEASY catalog, the new definition is added.
- If the specified secondary index name is already in the LEASY catalog, its definition can only be deleted or modified when the secondary index file contains no pointers for this index. Any pointers must be deleted beforehand with the LEASY-LOADSI utility routine (in the case of a model file group all the instances are checked).
- In the case of model file groups the secondary index information is stored only in the model and can therefore only be modified or deleted there.
- The key length of the secondary key index to be added must not exceed the existing maximum secondary key length for the file specified.
- Modification or deletion of secondary index definitions is only possible when the LEASY catalog concerned is not in use, i.e. no main task has been activated for it.

3.4 Handling of secondary index definitions

Up to 255 secondary keys per file can be defined, modified or deleted by means of the *KEY* operand of the **FIL* statement.

Each secondary key can comprise several key parts which may overlap.

The individual secondary keys may also overlap.

If a secondary key index definition consists of a single key part, the parentheses which enclose (*pos,len*) may be omitted.

The maximum key length of a secondary key can be calculated as follows:

(primary key length + total length of all key parts of the secondary index) ≤ 254

If all the bytes of a secondary key value are identical to the defined index suppression byte (*isb*), no entry is made for this secondary key value in the corresponding secondary index file. The user can thus avoid unnecessarily inflating the index file with blanks. If no index suppression byte is specified, secondary index pointers are created for all records of the primary file.

A minus sign ("-") after *keyname* means that the definition for this secondary index is to be deleted (only if *MOD=U* is specified and if the SI file does not contain any secondary index pointers for this secondary key).
If pointers for the secondary index to be deleted are still present, they must be deleted beforehand using the LEASY-LOADSI utility routine.

The following specifications are possible after the key part specifications *pos* and *len*:

- secondary key duplicates (*dup*):*YES/NO* (default value: YES)
- secondary index update (*upd*):*YES/NO* (default value: YES)

The order is significant. If the secondary index update is defined as *NO*, no pointers are created by the LEASY runtime system for this secondary index in the SI file, which can be useful for time-critical online data entry. The user must then update the secondary index file subsequently by means of the LEASY-LOADSI utility routine.

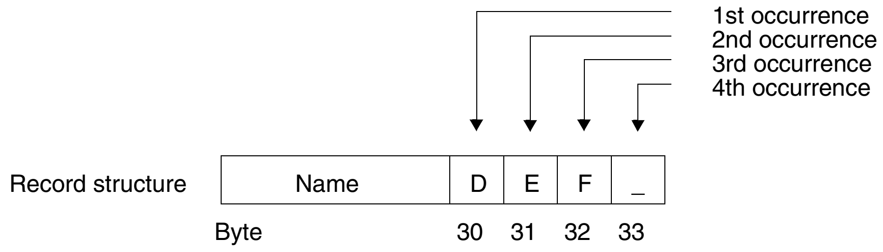
A multiple secondary key can be defined by specifying a repetition factor (*rep*). For each key part field of such a multiple secondary key, the distance between two neighboring occurrences of this key part can be specified.

When storing a file record, secondary indices are then stored in the SI file for a multiple secondary key *rep*. With records of variable record length, the number of occurrences of a secondary key may depend on the current record length if some of the occurrences are situated beyond the current data record length.

A record type field can be defined for each file by specifying position and length (*RTF* operand). The validity of each secondary key definition can be made dependent on the contents of the record type field. The definition is valid only if the contents of the current record type field correspond to the record type field values or record type field group values specified in the *RTP* operand, or, when *NONE* is specified, if the record type field is outside the data record.

Example of a multiple secondary key

An employee file contains a field with 4 columns indicating a person’s knowledge of foreign languages.



Secondary key definition KEY=(4*LANGUAGE,C' _' ,30,1)

The secondary key *LANGUAGE* therefore exists four times for each record:

1st occurrence:	byte 30
2nd occurrence:	byte 31
3rd occurrence:	byte 32
4th occurrence:	byte 33

The secondary key definition causes LEASY to create a secondary key for each key column which is not a blank. In this way it is possible to access a personnel file sorted according to an employee’s knowledge of foreign languages.

Examples of the syntax of secondary index definitions

```
*FIL FILE,KEY=(NAME,C'_',7,20)
```

The secondary key *NAME* at position 7 and with a length of 20 bytes is defined. If the secondary key value of a record consists of blanks only, no pointer is created in the SI file for this record. Duplicates of secondary key values are permitted. Secondary index pointers must be created by the runtime system.

```
*FIL FILE,KEY=(PLZ,30,4,YES,NO)
```

The secondary key *PLZ* is defined at position 30 with a length of 4 bytes. Duplicates of secondary key values are permitted. Secondary index pointers are not to be created.

```
*FIL FILE,KEY=(MULTKEY,(35,2),(14,1),(80,10),,NO)
```

The secondary key *MULTKEY* consists of three parts:

```
part 1 is at position 35, length =  2 bytes
part 2 is at position 14, length =  1 byte
part 3 is at position 80, length = 10 bytes
```

Secondary index pointers are not to be created by the runtime system.

```
*FIL FILE,KEY=(PERSNO,1,8,NO)
```

The secondary key *PERSNO* begins at position 1 and is 8 bytes long. Duplicates of secondary key values are not permitted by the runtime system.

```
*FIL FILE,KEY=(NUMBER,X'00',(50,4))
```

If the secondary key *NUMBER*, at position 50 and 4 bytes long, has the value X'00000000', secondary key pointers are not to be created.

```
*FIL FILE,MOD=U,KEY=(OLDKEY,-)
```

The definition of the secondary key *OLDKEY* is to be deleted.

```
*FIL FILE,KEY=(NAME,7,20)
```

The secondary key *NAME* exists exactly once for each record.

```
*FIL FILE,RECFORM=F,RECSIZE=100,RTF=(1,1),KEY=(5*KEY1,10,3)-
          ,KEY=(KEY2,(30,5),RTP=C'A'),KEYPOS=1,KEYLEN=5
```

The secondary key *KEY1* exists exactly 5 times for each record.

1st occurrence:	byte 10-12
2nd occurrence:	byte 13-15
3rd occurrence:	byte 16-18
4th occurrence:	byte 19-21
5th occurrence:	byte 22-24

The secondary key *KEY2* exists exactly once for all records with record type *A*. The record type field is the first byte of the primary key.

```
*FIL FILE,RECFORM=F-
        ,RECSIZE=100-
        ,KEY=(10*KEY3,X'00',(4,2),(30,2,5),(1,1,0))
```

The secondary key *KEY3* exists 10 times for each record and comprises 3 partial fields.

1st occurrence: bytes 4-5,30-31, 1

2nd occurrence: bytes 6-7,35-36, 1

...

10th occurrence: bytes 22-23,75-76, 1

```
*FIL FILE,KEYPOS=5,KEYLEN=7,RECFORM=V-
        ,RTF=(46,3)-
        ,KEY=(KEY4,(12,4),RTP=(C'AA',X'00'))-
        ,KEY=(KEY5,(20,1),N,RTP=(C'AAA',C'B',NONE))-
        ,KEY=(2000*KEY6,C'_',(49,1),RTP=X'FFFFFF')
```

The record type field is in bytes 46 to 48 of the record.

KEY4 occurs exactly once:

- in each record with record types X’C1C100’ to X’C1C1FF’ (X’C1 corresponds to C’A’).
- in all 47-byte records containing X’C1C1’ in bytes 46 through 47.
- in each record with record types X’000000’ through X’00FFFF’.
- in all 46-byte or 47-byte records containing the character X’00’ in byte 46.

KEY5 occurs:

- in all records < 46 characters.
- in all records ≥ 46 characters which contain a ’B’ in the 46th position.
- in all records ≥ 48 characters which contain the string C’AAA’ in bytes 46 through 48.

KEY6 occurs up to 2000 times in each record with the record type X’FFFFFF’ - depending on the current record length of the variable length record - but not when the key field contains a blank.

3.5 Setting up a LEASY application on private volumes

The LEASY catalog can be set up on a private disk using the *VOL* and *DEV* operands in the **CAT* statement or by means of a *CREATE-FILE* command.

LEASY files (master files, substitutes for temporary files and model files in a model file group, but not foreign files) which have been explicitly created by the user by means of **FIL*, may also be created as desired on private volumes by specifying *VOLUME*, *DEVICE-TYP*, *SPACE* and *PREMOUNT-LIST*.
This means that sequential master files, for example, may be transferred to tape.

Instances of a model file group are stored on the same volume(s) as the model.
If (in the case of an ISAM model file group) the index and data sections are to be on different volumes, these attributes must be defined for the model (see the *CREATE-FILE* command, *DEVICE-TYPE* and *VOLUME* in the "Commands, Volume 1 - 5" manual).

When creating the instance the index section of the ISAM file is written to the volume which has been reserved for the index section of the model file; the data section is written to the volume reserved for the data section of the model file.

The instance is created on the volume(s) of the model in the same way if the ISAM file has no data/index splitting or if it is not an ISAM file.

Only one data volume may be specified in a **FIL* statement. If a file is to extend over several data volumes, the appropriate *MODIFY-FILE-ATTRIBUTES* commands must be issued for this file after executing the LEASY-CATALOG utility routine.

Secondary index files are created on the volume containing the index section of the primary file (in the case of split ISAM files) or on the volume where the primary file begins. They are created on the private disk by means of a *CREATE-FILE* command with the *VOLUME* and *DEVICE-TYPE* operands **prior** to generation by means of LEASY-CATALOG.

```
/CREATE-FILE FILE-NAME=$userid.file-catalog.file-SI
/START-LEASY-CATALOG
.
.
FIL file,...,KEY=...
.
.
```

The instances of temporary files are created only on public volumes, even when their substitutes are on private disks. If LEASY files are created on private volumes, the operator must make these volumes available to the operating system even if the files are generated by LEASY-CATALOG. The files are then opened and closed to save all the attributes in the DMS catalog.

3.6 Transferring existing files to a LEASY catalog

There are several methods of transferring existing files to a LEASY file catalog:

- Import function (master files only)
- Copying the file
- Recataloging the file.

Import function

The import function (**FIL* statement, operand *MOD=C*) for transferring existing files only applies to LEASY master files. A precondition for importing is that all the DMS attributes have already been specified for the file in the DMS catalog.

If the *NAM* operand of the **FIL* statement is specified, the DMS file specified there is used and entered in the LEASY catalog under the same name. This file can also have a different user ID from the LEASY catalog file.

If the *NAM* operand is not specified, it is assumed that a DMS file already exists with a name conforming to the name structure of LEASY master files ($userid.file-catalog.file).

i A DMS file must not be contained or be open for writing in more than one LEASY catalog at a time. This would contradict both the internal lock logic of LEASY and the AIM and BIM data saving methods.

Copying the file

This method of transferring a file to the LEASY file catalog requires the following procedure:

1. The attributes of the file to be transferred must be defined in the LEASY catalog. This is effected with the **FIL* statement using a suitable file name (see page 22ff).
2. The *COPY-FILE* command is used to copy the source file into the file specified in the **FIL* statement.

Recataloging the file

The following steps must be performed:

1. Use the **FIL* statement to define the file attributes
2. Use the *DELETE-FILE* command to erase the file thus created

   ```
   $userid.file-catalog.file
   ```

   from the DMS catalog. The file attributes stored in the LEASY catalog are retained.
3. Use the *MODIFY-FILE-ATTRIBUTES* command to change the name of the old file to:

   ```
   $userid.file-catalog.file
   ```

i The file attributes specified in the **FIL* statement and the DMS attributes of the existing file must be absolutely identical.

3.7 Example

Tracer listing

```
/START-LEASY-CATALOG --------------------------------------------------------------------------------  (1)
 %  BLS0523 ELEMENT 'CATALOG', VERSION '06.2A' FROM LIBRARY
    ’:20SL:$TSOS.SYSPRG.LEASY.062’ IN PROCESS
 %  BLS0524 LLM 'LEASY-CATALOG', VERSION '06.2A' OF '2006-03-08 01:27:31' LOADED
 %  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
 %  LEA0101 LEASY CATALOG PROGRAM VERSION V6.2A STARTED
 CAT TESTCAT,TYP=N,PAS=C'TEST',ROM=N ------------------------------------------------------------  (2)
 FIL FILE1,AIM=R,BIM=N,WRP=C'SUSI',SHA=YES,PAD=50,ROM=Y ----------------------------------------  (3)
 FIL FILE2,LEA=M,AIM=R,WRP=C'OTTO',LOCK=Y,KEY=(KEY1,12,3,,N) -----------------------------------  (4)
 FIL FILE2.Z1 -----------------------------------------------------------------------------------  (5)
 FIL FILE2.Z2 -----------------------------------------------------------------------------------  (6)
 FIL FILE3,LEA=T,KEY=(KEY2,(32,5),(65,12)),KEY=(KEY3,32,10) -------------------------------------  (7)
 FIL FILE4 --------------------------------------------------------------------------------------  (8)
 INF ,A -----------------------------------------------------------------------------------------  (9)
 %   DNAME=FILE1
 %   FNAME=TESTCAT.FILE1
 00000003 :01KC:$LEATEST.TESTCAT.FILE1
   -------------------------------- HISTORY      --------------------------------
   CRE-DATE    = 2006-05-09  ACC-DATE    = 2006-05-09  CHANG-DATE = 2006-05-09
   CRE-TIME    =   10:44:58  ACC-TIME    =   10:44:58  CHANG-TIME =   10:44:58
   ACC-COUNT   = 1           S-ALLO-NUM = 0
   -------------------------------- SECURITY     --------------------------------
   READ-PASS   = NONE        WRITE-PASS = NONE         EXEC-PASS  = NONE
   USER-ACC    = OWNER-ONLY  ACCESS     = WRITE        ACL        = NO
   AUDIT       = NONE        DESTROY    = NO           EXPIR-DATE = 2006-05-09
   SP-REL-LOCK= NO                                     EXPIR-TIME =   00:00:00
   -------------------------------- BACKUP       --------------------------------
   BACK-CLASS = A            SAVED-PAG  = COMPL-FILE   VERSION    = 1
   MIGRATE    = ALLOWED
   -------------------------------- ORGANIZATION --------------------------------
   FILE-STRUC = ISAM         BUF-LEN    = STD(1)       BLK-CONTR  = PAMKEY
   IO(USAGE)  = READ-WRITE   IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
   REC-FORM   = (V,N)        REC-SIZE   = 0
   KEY-LEN    = 8            KEY-POS    = 5
   -------------------------------- ALLOCATION   --------------------------------
   SUPPORT    = PUB          S-ALLOC    = 12           HIGH-US-PA = 2
   EXTENTS     VOLUME      DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
       1       01KC.6      D3480
   NUM-OF-EXT = 1
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
 %   LEASYTYPE=...S.......LOCK=.......NO
 %   FCBTYPE=..ISAM.......PAD=........50
 %   RECSIMAX=02048.......KEYLEN=....008
 %   AIM=YES, SHORT.......BIM=........NO
 %   WRPASS=....YES.......RDPASS=.....NO
 %   ROM=.......YES
 %   DNAME=FILE2
 %   FNAME=TESTCAT.FILE2
 00000003 :01KC:$LEATEST.TESTCAT.FILE2
   ------------------------------- HISTORY      --------------------------------
   CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
   CRE-TIME   =   10:45:06  ACC-TIME   =   10:45:20  CHANG-TIME =   10:45:07
   ACC-COUNT  = 3           S-ALLO-NUM = 0
   ------------------------------- SECURITY     --------------------------------
   READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
   USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
   AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
   SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
   ------------------------------- BACKUP       --------------------------------
   BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
   MIGRATE    = ALLOWED
   ------------------------------- ORGANIZATION --------------------------------
   FILE-STRUC = ISAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
   IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
   REC-FORM   = (V,N)       REC-SIZE   = 0
   KEY-LEN    = 8           KEY-POS    = 5
   ------------------------------- ALLOCATION   --------------------------------
   SUPPORT    = PUB         S-ALLOC    = 12          HIGH-US-PA = 2
   EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
       1       01KC.0       D3480
   NUM-OF-EXT = 1
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
 00000003 :01KC:$LEATEST.TESTCAT.FILE2-SI
   ------------------------------- HISTORY      --------------------------------
   CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
   CRE-TIME   =   10:45:07  ACC-TIME   =   10:45:20  CHANG-TIME =   10:45:07
   ACC-COUNT  = 3           S-ALLO-NUM = 0
   ------------------------------- SECURITY     --------------------------------
   READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
   USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
   AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
   SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
   ------------------------------- BACKUP       --------------------------------
   BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
   MIGRATE    = ALLOWED
   ------------------------------- ORGANIZATION --------------------------------
   FILE-STRUC = ISAM        BUF-LEN    = STD(2)      BLK-CONTR  = PAMKEY
   IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
   REC-FORM   = (V,N)       REC-SIZE   = 0
   KEY-LEN    = 12          KEY-POS    = 5
   ------------------------------- ALLOCATION   --------------------------------
   SUPPORT    = PUB         S-ALLOC    = 2           HIGH-US-PA = 3
   EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
       1       01KC.1       D3480
   NUM-OF-EXT = 1
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
```

```
%   LEASYTYPE=...M.......LOCK=......YES
%   FCBTYPE=..ISAM.......PAD=........15
%   RECSIMAX=02048.......KEYLEN=....008
%   AIM=YES, SHORT.......BIM=.......YES
%   ROM=........NO
%   WRPASS=....YES.......RDPASS=.....NO
%   KEY=.(KEY1    ,      (00012,003),YES,NO)
%   .....................HAS NO POINTERS IN SI FILE ----------------------------  (10)
%   DNAME=FILE2.Z1
%   FNAME=TESTCAT.FILE2.Z1
00000003 :01KC:$LEATEST.TESTCAT.FILE2.Z1
  ------------------------------- HISTORY      --------------------------------
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:14  ACC-TIME   =   10:45:14  CHANG-TIME =   10:45:14
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ------------------------------- SECURITY     --------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ------------------------------- BACKUP       --------------------------------
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION --------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 8           KEY-POS    = 5
  ------------------------------- ALLOCATION   --------------------------------
  SUPPORT    = PUB         S-ALLOC    = 12          HIGH-US-PA = 2
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.3      D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
00000003 :01KC:$LEATEST.TESTCAT.FILE2.Z1-SI
  ------------------------------- HISTORY      --------------------------------
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:14  ACC-TIME   =   10:45:14  CHANG-TIME =   10:45:14
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ------------------------------- SECURITY     --------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ------------------------------- BACKUP       --------------------------------
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION --------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(2)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 12          KEY-POS    = 5
  ─────────────────────────────── ALLOCATION   ───────────────────────────────
  SUPPORT    = PUB         S-ALLOC    = 2          HIGH-US-PA = 3
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.6       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
%   LEASYTYPE=...M.......LOCK=......YES
%   FCBTYPE=..ISAM.......PAD=........15
%   RECSIMAX=02048.......KEYLEN=....008
%   AIM=YES, SHORT.......BIM=.......YES
%   WRPASS=....YES.......RDPASS=.....NO
%   ROM=........NO
%   DNAME=FILE2.Z2
%   FNAME=TESTCAT.FILE2.Z2
00000003 :01KC:$LEATEST.TESTCAT.FILE2.Z2
  ─────────────────────────────── HISTORY      ───────────────────────────────
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:20  ACC-TIME   =   10:45:20  CHANG-TIME =   10:45:20
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ─────────────────────────────── SECURITY     ───────────────────────────────
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ─────────────────────────────── BACKUP       ───────────────────────────────
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ─────────────────────────────── ORGANIZATION ───────────────────────────────
  FILE-STRUC = ISAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 8           KEY-POS    = 5
  ─────────────────────────────── ALLOCATION   ───────────────────────────────
  SUPPORT    = PUB         S-ALLOC    = 12         HIGH-US-PA = 2
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.0       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
00000003 :01KC:$LEATEST.TESTCAT.FILE2.Z2-SI
  ─────────────────────────────── HISTORY      ───────────────────────────────
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:20  ACC-TIME   =   10:45:20  CHANG-TIME =   10:45:20
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ─────────────────────────────── SECURITY     ───────────────────────────────
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ─────────────────────────────── BACKUP       ───────────────────────────────
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION --------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(2)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 12          KEY-POS    = 5
  ------------------------------- ALLOCATION   --------------------------------
  SUPPORT    = PUB         S-ALLOC    = 2           HIGH-US-PA = 3
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.1       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
%   LEASYTYPE=...M.......LOCK=......YES
%   FCBTYPE=..ISAM.......PAD=........15
%   RECSIMAX=02048.......KEYLEN=....008
%   AIM=YES, SHORT.......BIM=.......YES
%   WRPASS=....YES.......RDPASS=.....NO
%   ROM=........NO
%   DNAME=FILE3
%   FNAME=TESTCAT.FILE3
00000003 :01KC:$LEATEST.TESTCAT.FILE3
  ------------------------------- HISTORY      --------------------------------
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:26  ACC-TIME   =   10:45:26  CHANG-TIME =   10:45:26
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ------------------------------- SECURITY     --------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ------------------------------- BACKUP       --------------------------------
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION --------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 8           KEY-POS    = 5
  ------------------------------- ALLOCATION   --------------------------------
  SUPPORT    = PUB         S-ALLOC    = 12          HIGH-US-PA = 2
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.2       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
 00000003 :01KC:$LEATEST.TESTCAT.FILE3-SI
  ------------------------------- HISTORY      --------------------------------
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:26  ACC-TIME   =   10:45:26  CHANG-TIME =   10:45:26
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ------------------------------- SECURITY      -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ------------------------------- BACKUP        -------------------------------
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION  -------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(2)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 26          KEY-POS    = 5
  ------------------------------- ALLOCATION    -------------------------------
  SUPPORT    = PUB         S-ALLOC    = 4           HIGH-US-PA = 3
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.3       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
%   LEASYTYPE=...T.......LOCK=.......NO
%   FCBTYPE=..ISAM.......PAD=........15
%   RECSIMAX=02048.......KEYLEN=....008
%   AIM=........NO.......BIM=.......YES
%   WRPASS=.....NO.......RDPASS=.....NO
%   ROM=........NO
%   KEY=.(KEY2    ,      (00032,005),
%   .....................(00065,012),YES,YES)
%   .....................HAS NO POINTERS IN SI FILE
%   KEY=.(KEY3    ,      (00032,010),YES,YES)
%   .....................HAS NO POINTERS IN SI FILE
%   DNAME=FILE4
%   FNAME=TESTCAT.FILE4
00000003 :01KC:$LEATEST.TESTCAT.FILE4
  ------------------------------- HISTORY       -------------------------------
  CRE-DATE   = 2006-05-09  ACC-DATE   = 2006-05-09  CHANG-DATE = 2006-05-09
  CRE-TIME   =   10:45:31  ACC-TIME   =   10:45:32  CHANG-TIME =   10:45:32
  ACC-COUNT  = 1           S-ALLO-NUM = 0
  ------------------------------- SECURITY      -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = NO          EXPIR-DATE = 2006-05-09
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  ------------------------------- BACKUP        -------------------------------
  BACK-CLASS = A           SAVED-PAG  = COMPL-FILE  VERSION    = 1
  MIGRATE    = ALLOWED
  ------------------------------- ORGANIZATION  -------------------------------
  FILE-STRUC = ISAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
  IO(USAGE)  = READ-WRITE  IO(PERF)   = STD         DISK-WRITE = IMMEDIATE
  REC-FORM   = (V,N)       REC-SIZE   = 0
  KEY-LEN    = 8           KEY-POS    = 5
  ------------------------------- ALLOCATION  -------------------------------
  SUPPORT    = PUB        S-ALLOC    = 12          HIGH-US-PA = 2
  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
      1       01KC.4       D3480
  NUM-OF-EXT = 1
:01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
%   LEASYTYPE=...S.......LOCK=.......NO
%   FCBTYPE=..ISAM.......PAD=........15
%   RECSIMAX=02048.......KEYLEN=....008
%   AIM=........NO.......BIM=.......YES
%   WRPASS=.....NO.......RDPASS=.....NO
%   ROM=........NO
END ---------------------------------------------------------------------------  (11)
%  LEA0110 NORMAL TERMINATION OF LEASY CATALOG PROGRAM
/START-LEASY-CATALOG ----------------------------------------------------------  (12)
%  BLS0523 ELEMENT 'CATALOG', VERSION '06.2A' FROM LIBRARY
   ':2OSL:$TSOS.SYSPRG.LEASY.061' IN PROCESS
%  BLS0524 LLM 'LEASY-CATALOG', VERSION '06.2A' OF '2006-03-08 01:27:31' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0101 LEASY CATALOG PROGRAM VERSION V6.2A STARTED
 CAT TESTCAT,PAS=C'TEST',CPC=COPYCAT,INF=Y  -----------------------------------  (13)
%  LEA5102 ****** NO SESSION CREATED ******
%  LEA5002 SHADOW DIRECTORY NAME: $LEATEST.COPYCAT SHADOW SUFFIX NAME:
%  LEA5105 DMS FILENAMES WITH CATID
%  LEA5108 ROM = NO IS SPECIFIED
 FIL FILE4,MOD=U,KEY=(KEY4,(45,2),(44,10),(67,3)) -----------------------------  (14)
 INF --------------------------------------------------------------------------  (15)
%   DNAME=FILE1
%   FNAME=TESTCAT.FILE1
      3 :01KC:$LEATEST.TESTCAT.FILE1
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
%   DNAME=FILE2
%   FNAME=TESTCAT.FILE2
      3 :01KC:$LEATEST.TESTCAT.FILE2
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
      3 :01KC:$LEATEST.TESTCAT.FILE2-SI
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
%   DNAME=FILE2.Z1
%   FNAME=TESTCAT.FILE2.Z1
      3 :01KC:$LEATEST.TESTCAT.FILE2.Z1
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
      3 :01KC:$LEATEST.TESTCAT.FILE2.Z1-SI
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
%   DNAME=FILE2.Z2
%   FNAME=TESTCAT.FILE2.Z2
      3 :01KC:$LEATEST.TESTCAT.FILE2.Z2
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        1  REL=        0 PAGES
      3 :01KC:$LEATEST.TESTCAT.FILE2.Z2-SI
 :01KC: PUBLIC:      1 FILE  RES=        3  FREE=        0  REL=        0 PAGES
```

```
%   DNAME=FILE3
%   FNAME=TESTCAT.FILE3
       3 :01KC:$LEATEST.TESTCAT.FILE3
 :01KC: PUBLIC:      1 FILE  RES=       3  FREE=       1  REL=       0 PAGES
       3 :01KC:$LEATEST.TESTCAT.FILE3-SI
 :01KC: PUBLIC:      1 FILE  RES=       3  FREE=       0  REL=       0 PAGES
%   DNAME=FILE4
%   FNAME=TESTCAT.FILE4
       3 :01KC:$LEATEST.TESTCAT.FILE4
 :01KC: PUBLIC:      1 FILE  RES=       3  FREE=       1  REL=       0 PAGES
       3 :01KC:$LEATEST.TESTCAT.FILE4-SI
 :01KC: PUBLIC:      1 FILE  RES=       3  FREE=       0  REL=       0 PAGES
 ERA FILE2,PAS=C'OTTO' ——————————————————————————————————————————————————————— (16)
%  LEA2120 ERASE M-FILE MODEL NOT ALLOWED: INSTANCES ARE STILL CATALOGED ————— (17)
 ERA FILE2.Z1,PAS=C'OTTO' ———————————————————————————————————————————————————— (18)
 ERA FILE2.Z2,PAS=C'OTTO'
 ERA FILE2,PAS=C'OTTO' ——————————————————————————————————————————————————————— (19)
 END ————————————————————————————————————————————————————————————————————————— (20)
%  LEA0110 NORMAL TERMINATION OF LEASY CATALOG PROGRAM

/START-LEASY-CATALOG ————————————————————————————————————————————————————————— (21)
%  BLS0523 ELEMENT 'CATALOG', VERSION '06.2A' FROM LIBRARY
   ’:20SL:$TSOS.SYSPRG.LEASY.062’ IN PROCESS
%  BLS0524 LLM 'LEASY-CATALOG', VERSION '06.2A' OF '2006-03-08 01:27:31' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0101 LEASY CATALOG PROGRAM VERSION V6.2A STARTED
 COM DELETE THE FILES AND THE FILE CATALOG
 CAT TESTCAT,TYP=E,PAS=C'TEST' ——————————————————————————————————————————————— (22)
%  LEA2118 LEASY DIRECTORY NOT EMPTY. DO YOU REALLY WANT TO ERASE NON EMPTY LEASY
DIRECTORY TESTCAT? (Y=YES; N=NO)
 N ——————————————————————————————————————————————————————————————————————————— (23)
%  LEA5104 LEASY DIRECTORY NOT ERASED
 ERA FILE1,PAS=C'SUSI' ——————————————————————————————————————————————————————— (24)
 ERA FILE3
 ERA FILE4
 INF
%  LEA2113 NO FILE FOUND
 CAT TESTCAT,TYP=E,PAS=C'TEST' ——————————————————————————————————————————————— (25)
 INF ————————————————————————————————————————————————————————————————————————— (26)
%  LEA6101 UNDEFINED LEASY DIRECTORY
 END ————————————————————————————————————————————————————————————————————————— (27)
%  LEA0110 NORMAL TERMINATION OF LEASY CATALOG PROGRAM
```

Explanation

(1) The LEASY-CATALOG utility routine is called.

(2) The LEASY catalog *TESTCAT* with the password *C'TEST'* is generated. *ROM=NO* is used to specify a default for subsequent *FIL* statements.

(3) The *FILE1* file is generated to be run with AIM saving but without BIM saving. The write password is maintained by the LEASY runtime system but is not evaluated. The file is shareable. When writing the file with the ISAM load mode (USAGE mode EXLD), 50% of the blocks are to be reserved for subsequent extensions. Generation of *FILE1* as the master file is the default value. *FILE1* is reserved for setting to READ-ONLY mode.

(4) The file attributes of the model file group *FILE2* are defined. The instances of this group are to be run with AIM and BIM saving. This model file group with the password *C'OTTO'* is to be locked for subsequent LEASY sessions. Duplicates are permitted for the secondary key *KEY1*, which has position 12 and a length of 3 bytes; no secondary index pointers are to be created by the LEASY runtime system.

(5) (6)
The instance files in model file group *FILE2* are generated.

(7) The temporary file *FILE3* with secondary keys *KEY2* and *KEY3*, whose key areas partially overlap, is created.

(8) The file *FILE4* with the default values is created.

(9) Information is requested on all file attributes of all files in the LEASY catalog *TESTCAT*.

(10) The example of the file attributes of *TESTCAT.FILE2-SI* shows how the message indicating that no pointers have been created for model file groups only appears for the model.
Since the model file does not contain data there cannot be any pointers in the SI file of the model.
The existence of SI pointers in the SI files of instances is only ascertainable from the information statement for the individual instances.

(11) The LEASY-CATALOG utility routine is terminated.

(12) The LEASY-CATALOG utility routine is started.

(13) The existing LEASY catalog *TESTCAT* is assigned with specification of the password. The name structure for shadow files is defined at the same time and information is requested concerning the last LEASY session.

(14) The secondary key *KEY4* is defined for file *FILE4*, which already exists. Specification of the operand *MOD=U* is mandatory.

(15) The files in the LEASY catalog *TESTCAT* are to be listed.

(16) Model file group *FILE2* is deleted; the password must also be specified.

(17) Error message from the utility routine:

Deletion of model file group *FILE2* is not yet possible because instances in this group still exist.

(18) The instances are deleted with specification of the password.

(19) The model file group can now be deleted.

(20) The LEASY-CATALOG utility routine is terminated.

(21) The LEASY-CATALOG utility routine is started.

(22) The LEASY catalog *TESTCAT* is assigned using the password *C'TEST'* and *TYP=E*, i.e. the LEASY catalog is to be deleted.

(23) Utility routine message:

The LEASY catalog still contains files. Should the catalog be erased? Response: *no*.

(24) The files are deleted.

(25) The LEASY catalog may now be deleted.

(26) The information statement indicates that no more LEASY catalogs are assigned.

(27) The LEASY-CATALOG utility routine is terminated.

4 LEASY-CONVERT

The LEASY-CONVERT utility routine is used to convert existing LEASY application files from PAMKEY format to NONKEY format and vice versa. Conversion should be implemented exclusively by means of LEASY-CONVERT, since the use of other commands or programs (COPY, PERCON, ARCHIVE, PAMCONV etc.) could lead to inconsistencies in the LEASY catalog.

Conversion from PAMKEY to NONKEY is necessary for:

- transferring files to NK disk peripherals
- gradually upgrading to the new access methods on PAMKEY peripherals
- using the functional extensions of NK-ISAM (e.g. buffer pools for ISAM files).

Prerequisites for conversion

- The LEASY application must have a consistent status. AIM files must be incorporated into the application files first by means of reconstruction, and BIM files by means of rollbacks.
- LEASY-CONVERT can convert the following LEASY files:
  - LEASY catalogs
  - master files
  - foreign files
  - secondary index files
  - model files and their instances
  - backups of temporary files
- The following LEASY files are not converted:
  - AIM file generations
  - BIM files
  - openUTM status files
  - shadow files
  - instances of temporary files
  - RECONST ENTER files

Shadow files

Since the shadow files have the same status as their original files, users can create them by copying the converted originals.

4.1 LEASY-CONVERT statements

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 3: Overview of statements (LEASY-CONVERT)

The statements are read in via SYSDTA. If column 72 of the statement contains a hyphen ("-"), the next line is interpreted as a continuation line.

The statements *CAT* and *FIL* are at first only checked for syntax and collected. The relevant files are not checked and converted until the *END* statement is issued.

The statements are described below in alphabetical order.

CAT Specify LEASY catalog

The *CAT* statement specifies the LEASY catalog that is to be converted containing the files that are subsequently specified, which are to be converted. The specifications for a LEASY catalog and the files entered in it can be repeated any number of times, i.e. several catalogs can be converted in a single run.

Statement	Operands

`CAT[ALOG]`	`catalog [,C[ATID]= catid]`
	`[,VOL[UME]= vsn]`
	`[,DEV[ICE]= device]`

catalog [:catid:][$userid.][catalogname]

Name of the LEASY source catalog.

catid Global catalog identifier (max. 4 characters).

userid User ID (max. 8 characters).

If the specified user ID deviates from the one under which LEASY-CONVERT was started, the catalog must already have been converted in the direction required.

catalogname LEASY catalog name (max. 8 characters).

CATID=catid Local catalog ID under which the target catalog is to be set up. The local catalog ID has priority over the global one.

The *VOLUME* and *DEVICE* operands must be specified when the target catalog is to be set up on a private disk.

VOLUME=vsn Private disk for the target catalog.

DEVICE=device Device type for the target catalog.

(For the range of possible values see the *DEVICE-TYP* operand in the "Introductory Guide to DMS" manual).

CON Specify conversion direction and attributes

The *CON* statement determines the direction of conversion, the global target catalog ID, the file protection attribute and the record protection. The *CON* statement must be the first statement specified, otherwise a warning is output and the statement is then ignored.

Statement	Operands
CON[VERT]	[TO={ P[AMKEY] 4K-N[ONKEY] N[ONKEY] }][,C[ATID]=catid] [O[VERWRITE]={ Y[ES] N[O] }] [T[RUNCATE]={ Y[ES] N[O] }]

TO= Specifies the direction of conversion.

PAMKEY Conversion to PAMKEY format.

4K-NONKEY Conversion to NK4 format.

NONKEY Conversion to NK2 format.

When converting PAM and DAM files to 4K-NONKEY or NONKEY format, the *BLKCTRL* specification in the *FIL* statement is also evaluated (see page 80f).

If the pubset of the target file has the N4K format, *TO=4K-NONKEY* is automatically generated if *TO=NONKEY* is specified.

CATID=catid Global target catalog (pubset) of the conversion.

If *catid* is specified, all result files are converted to this pubset except for files or LEASY catalogs with a local CATID value. If neither a global nor a local catalog ID is specified, conversion is to the home catalog ID.

OVERWRITE= Write protection for source files if conversion is to the home catalog ID.

YES Target files can overwrite source files.

NO Source files are protected.

TRUNCATE= Protection against record truncation.

YES Records for which the block length of the output file is insufficient are truncated. This only applies to conversion to non-key and maximum block length.

NO Records must not be truncated.

END Terminate statement input

The *END* statement starts the conversion of the files and subsequently terminates LEASY-CONVERT.

Statement	Operands
`END`	

FIL Specify files for conversion

The files to be converted are specified in the *FIL* statement. A *FIL* statement refers to the *CAT* statement that precedes it.

In the case of model files, the model and all its instances are converted. If there are secondary keys for the application file, the relevant SI file is also converted.

<table>
<tr><th>Statement</th><th>Operands</th></tr>
<tr><td>FIL[E]</td><td><pre>
{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 }
</pre></td></tr>
</table>

file Logical file name (≤ 8 characters) of the file to be converted.

*ALL All files of the specified LEASY catalog are converted. The operands described below are not specified in this case.

CATID=catid Local specification for the catalog of the target file.

This has priority over the global catalog ID of the target file.

The *VOLUME* and *DEVICE* operands must be specified when the target file is to be created on private storage media (private disk, tape).

VOLUME=vsn |(vsn1,..)

Private data volumes of the primary target file.

DEVICE=device Device type of the primary target file.
(For the range of possible values see the *DEVICE-TYP* operand in the “Introductory Guide to DMS” manual).

The *SIVOL* and *SIDEV* operands must be specified when the target SI file is to be created on private storage media.

SIVOL=vsn |(vsn1,..) Private data volumes of the SI target file.

SIDEV=device Device type of the SI target file (for range of possible values see the *DEVICE* operand)

BLKSIZE= Block size of the target file.

ADAPT The block size of the target file is adjusted to the required size.

SAME The block size of the source file is adopted in the target file.

(STD,blksize) Standard specification of the block length. 1 ≤ *blksize* ≤ 16.

blksize Non-standard specification of the block length in bytes for tape files. 20 ≤ *blksize* ≤ 32768.

TYPE= Type of file to be converted.

C File entered in a LEASY catalog.

F Foreign file.

A foreign file (i.e. a file that is either not entered in the specified LEASY file or is entered with *LEASYTYPE=F*) must be assigned the appropriate name using a *ADD-FILE-LINK* command before LEASY-CONVERT is called. The logical name specified must be used as the link name.

BLKCTRL= File format, relevant only for DAM and PAM files with the conversion direction *CON TO=NONKEY* or *CON TO=4K-NONKEY* .

DATA The first 16 bytes of the data block contain block-specific administration information, and the remaining 2032 bytes contain user data.

NO The data block contains only user data.

4.2 Conversion types

A file must be converted under the same user ID under which it was set up. If files are distributed over a number of user IDs, the catalog and the files cataloged under its user ID are to be converted first, followed by the other files under their respective user IDs.

In the case of conversion from tape to tape, a different volume must be specified for the target file than for the source file.

The name of the converted target file can only differ from the name of the source file in the *catid* component. The conversion process differentiates between identical and non-identical file names.

File attributes

The following file attributes entered in the DMS catalog are transferred from the source file to the target file:

ACCESS, AUDIT, BACK-CLASS, ACL, COD-CH-SET, DESTROY, IOUSAGE, IOPERF, SAVED-PAG, MIGRATE, EXPIR-DATE, EXPIR-TIME and USER-ACC

If the conversion is performed with the TSOS privilege, the READ-PASS, WRITE-PASS and EXEC-PASS passwords are also transferred.

Converting files to different user IDs

LEASY-CONVERT only converts files within the same user ID, i.e. the user ID of the target file remains identical to the user ID of the source file.

If LEASY-CONVERT is started with the TSOS privilege, files with any user ID can be converted.

From any other user ID, only those files can be converted that are entered under the user ID under which LEASY-CONVERT was started.

Target file size

As primary and secondary size (allocation) of the target file, the relevant values are transferred from the source file.

Conversion at the same location

This is when

- the *catid* components of the source and target files are identical
- the conversion takes place on the same private disk or tape.

LEASY operating procedures during conversion when the source or target file is stored on disk:

1. Convert the source file to an intermediate file called *S.<tsn>.LEA.CONVERT*, where *tsn* is the same as the 4-digit number of the current task.
2. Delete the source file.
3. Rename the intermediate file to the target file (source file name).

If an error occurs, the intermediate file is deleted.

LEASY operating procedures when conversion is from tape to tape:

1. Delete the catalog entry of the source file.
2. Open the source file with the name *S.<tsn>.LEA.CONVERT*. A DMS message is displayed, to which the user or operator should respond with *I* (for “ignore”).
3. Convert the source file to the definitively named target file.

If an error occurs, the catalog entry of the source file is recreated.

Conversion to another file

If an error occurs when the source file is converted to the target file, the target file is deleted.

4.3 Adjusting the block length

LEASY supports the NK format with *BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK* and with *BLOCK-CONTROL-INFO=NO*. In NK format with *BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK*, the addition of the block-specific administration data involved in conversion from KEY to NONKEY can - if the original block length is retained - result in data being lost and, with conversion from NONKEY to KEY, in unnecessary wastage.

The user can therefore either modify the block size of the target file explicitly or have the block length adjusted automatically by LEASY-CONVERT.

The record length of the source file is specified explicitly by the user when the file is set up, or determined implicitly by LEASY on the basis of the block size, the *BLOCK-CONTROL-INFO* value and the FCB type.

For tape files, files with fixed record format, and PAM and DAM files, the *SLS* is assumed to have been specified explicitly.

The following abbreviations are used in this section:

MAX(x,y)	the larger of the two values *x* and *y*
[x]	*x* rounded up to a multiple of 2048
BLS	block size of the source file
BLT	block size of the target file
SLF	length of the largest record in the input file
SLS	(maximum) record length of the source file in the catalog
SLT	(maximum) record length of the target file in the catalog

Conversion to NK format

The block size can only be automatically adjusted for conversion to NK format. It is increased, starting from the block size of the source file, until all the data to be transferred is accommodated.

If even the maximum block size of *STD(SIZE=16)* or 32768 bytes is not enough for a record of the source file, LEASY-CONVERT truncates the record (see the *TRUNCATE* operand of the *CON* statement).

ISAM files

LEASY-CONVERT adjusts the block length automatically.

BLT = BLS

When *SLS > BLS-n*16-16* with *BLS=STD(SIZE=n)*, the user is responsible for ensuring that no overflow blocks are produced when the block length is increased.

The following minimum values apply when *BLT* is specified explicitly:

BLT ≥ SLS + 4	fixed record length
BLT ≥ SLS	variable record format, SLS specified explicitly
BLT ≥ SLF	variable record format, SLS specified implicitly

Record length of the target file:

SLT = SLS	when SLS is specified explicitly
SLT = BLT	otherwise

SAM files

The following value is assumed for *BLT* when the block length is automatically adjusted:

MAX(BLS,[SLS + 16])	files with standard blocks
MAX(BLS,SLS + 16)	tape files with non-standard blocks

When *BLT* is specified explicitly, the following minimum values apply:

BLT ≥ SLS + 16	when SLS is specified explicitly
BLT ≥ SLF + 16	otherwise

Record length of the target file:

SLT = 32752	SLS specified explicitly, SLS > 32752
SLT = SLS	SLS specified explicitly, otherwise
SLT = BLT - 16	SLS determined implicitly

PAM and DAM files

The following value is adopted for *BLT* when the block length is adjusted automatically:

MAX(BLS,[SLT + 12])

When *BLT* is specified explicitly, the following minimum value applies:

BLT ≥ [SLT + 12]

Record length of the target file:

SLT = 32756	if SLQ > 32756
SLT = SLS	otherwise

In the case of PAM files, when the file sizes are transferred a possible change in block size is taken into account as follows:

PST = PSS / BLS * BLT

SST = SSS / BLS * BLT

Abbreviations

PST	primary size of the target file
PSS	primary size of the source file
SST	secondary size of the target file
SSS	secondary size of the source file
BLT	block size of the target file
BLS	block size of the source file

Conversion to PAMKEY format

Unless otherwise specified, the block length is not automatically adjusted:

$\mathrm{BLT} = \mathrm{BLS}$

The minimum values that apply when *BLT* is specified explicitly and the record length of the target file are dependent on the access method, as below:

ISAM files

$\mathrm{BLT} \geq \mathrm{SLS} + 4$	fixed record length
$\mathrm{BLT} \geq \mathrm{SLS}$	variable record format, SLS specified explicitly
$\mathrm{BLT} \geq \mathrm{SLF}$	variable record format, SLS determined implicitly
$\mathrm{SLZ} = \mathrm{SLS}$	when SLS is specified explicitly
$\mathrm{SLZ} = \mathrm{BLT}$	otherwise

SAM files

$\mathrm{BLT} \geq \mathrm{SLS}$	fixed record length
$\mathrm{BLT} \geq \mathrm{SLS} + 4$	variable record format, SLKS specified explicitly
$\mathrm{BLT} \geq \mathrm{SLF} + 4$	variable record format, SLS determined implicitly
$\mathrm{SLZ} = \mathrm{SLS}$	when SLS is specified explicitly
$\mathrm{SLZ} = \mathrm{BLT} - 4$	otherwise

PAM and DAM files

$\mathrm{BLT} \geq \mathrm{SLS}$
$\mathrm{SLT} = \mathrm{SLS}$

Conversion to NK4 format

If the block length is specified explicitly, it must be (STD,2) or a multiple thereof.

The minimum values specified for conversion to non-key format shall apply:

When converting from the PAMKEY format, the block length is determined implicitly according to the formula specified for converting to NK format. When converting from the NK format, the block length is determined from the block length of the source file. The result of the calculation is always rounded up to (STD,2) or a multiple thereof.

Conversion from NK4 format

If the block length is specified explicitly, the minimum values specified for conversion to PAMKEY or NK format shall apply.

The block size of the source file is used implicitly.

4.4 Deleted records in PAM and DAM files

PAM files

In PAMKEY format, a deleted block is identified by 8 X’FF’ in the user part of the PAM key, and in NK format the first byte in the data block is set to X’FF’.

Data blocks containing X’FF’ in the first byte are interpreted as being deleted.

In contrast to the processing of DAM files, conversion from PAMKEY format to NK format is aborted as soon as a data block begins with X’FF’ without an entry (8 X’FF’) in the user part of the PAM key indicating that the block in question has been deleted.

When converting from NK format to PAMKEY format, it is not possible in the case of files created and processed with *BLKCTRL=DATA* and containing X’FF’ in the first data byte to say for certain that the block has been deleted. In this case, analogous to the way DAM files are treated, the block is transferred as deleted.

DAM files

The user part of the PAM key is not used. In both NK and PAMKEY format a deleted record is identified by X’FF’ in the first data byte of the record.

In the course of conversion a deleted record is transferred just like any other record.

4.5 Restrictions affecting conversion

Tape files

The file format *BLOCK-CONTROL-INFO=NO* is not supported for tape files processed with UPAM.

Disk files

Conversion between different NK file formats, e.g. from DATA to NO, or from NO to DATA, is not possible. The only way to do this is to through the intermediate step of conversion to PAMKEY file format.

4.6 LEASY-CONVERT: errors

When errors occur which prevent the complete execution of the desired conversion, the conversion run is terminated abnormally.

The following errors can occur:

Errors in syntax when entering the statement from a file (procedure or ENTER file). The run is terminated after the last statement is read in. When the statements are being entered on a data display terminal, the error can be corrected immediately.

Error during the conversion of a file, such as a DMS error, or the source file was marked as not overwritable during conversion at the same location.

- Error during conversion of a LEASY catalog.

  The files entered in the catalog are not converted either. The conversion run is continued on the next catalog if one is specified, and terminated if not.

- Error during conversion of a primary file.

  In addition to the above error types, there are the following:

  - The block length is too short.
  - *X'FF'* is at the beginning of a PAM file block.

  Associated SI files are not converted either. The conversion run is continued on the next primary file or the next catalog, or it is terminated.

- Error during conversion of a secondary index file.

  The conversion run is continued on the next primary file or the next catalog, or it is terminated.

The block size of the output file is too small to accommodate the converted records.

When the following minor error occurs, a warning message is displayed but the routine is terminated normally:

In the case of conversion from PAMKEY to NK format, even the maximum block size is too small to accommodate the source data. The records are truncated to the maximum possible length.

4.7 Example

In the following example of the LEASY-CONVERT utility routine, a foreign file and two catalog files (including the files they contain) are converted from PMKEY to NK format. The conversion types "conversion at the same location" and "conversion to another file" are used for this.

Files to be converted

```
:20SC:$LEASYTST.FOREIGNFILE
:20SC:$LEASYTST.CAT1.LEASYCAT
:20SC:$LEASYTST.CAT1.ISAM
:20SC:$LEASYTST.CAT1.ISAM-SI
:20SC:$LEASYTST.CAT1.MODEL
:20SC:$LEASYTST.CAT1.MODEL.EXP1
:20SC:$LEASYTST.CAT1.MODEL.EXP2
:20SC:$LEASYTST.CAT1.SAM
:20SC:$LEASYTST.CAT2.LEASYCAT
:20SC:$LEASYTST.CAT2.ISAM1
:20SC:$LEASYTST.CAT2.SAM1
```

Converted files

```
:20SC:$LEASYTST.FOREIGNFILE
:SALI:$LEASYTST.CAT1.LEASYCAT
:20SC:$LEASYTST.CAT2.LEASYCAT
:40SG:$LEASYTST.CAT1.ISAM
:40SG:$LEASYTST.CAT1.ISAM-SI
:40SG:$LEASYTST.CAT1.MODEL
:40SG:$LEASYTST.CAT1.MODEL.EXP1
:40SG:$LEASYTST.CAT1.MODEL.EXP2
:40SG:$LEASYTST.CAT1.SAM
:20SC:$LEASYTST.CAT2.ISAM1
:20SC:$LEASYTST.CAT2.SAM1
```

Description of the conversion run

```
/ADD-FILE-LINK LINK-NAME=FOREIGN,FILENAME=FOREIGNFILE                     (1)
/START-LEASY-CONVERT                                                      (2)
%  BLS0523 ELEMENT 'CONVERT', VERSION '06.2A' FROM LIBRARY
   ’:20SL:$TSOS.SYSPRG.LEASY.062’ IN PROCESS
%  BLS0524 LLM 'LEASY-CONVERT', VERSION '06.1A' OF '2006-04-28 12:05:40’
LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS
RESERVED
%  LEA0801 LEASY CONVERT PROGRAM VERSION V6.2A STARTED
CON OVERWRITE=YES,TO=NONKEY                                               (3)
FILE FOREIGN,TYPE=F                                                       (4)
CAT CAT1,CATID=SALI                                                       (5)
FILE *ALL,CATID=40SG                                                      (6)
CAT CAT2                                                                  (7)
FILE SAM1,BLKSIZE=(STD,4)                                                 (8)
FILE ISAM1                                                                (9)
END                                                                      (10)
%  LEA5801 CONVERSION OF DIRECTORY 'CAT1' INTO NONKEY-FORMAT STARTED
%  LEA5802 CONVERSION OF DIRECTORY 'CAT1' COMPLETED
%  LEA5803 CONVERSION OF FILE 'FOREIGN' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'FOREIGN' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'ISAM' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'ISAM' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'ISAM-SI' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'ISAM-SI' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'MODEL' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'MODEL' WITH 00000000 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'MODEL.EXP1' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'MODEL.EXP1' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'MODEL.EXP2' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'MODEL.EXP2' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'SAM' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'SAM' WITH 00000002 RECORDS COMPLETED
%  LEA5801 CONVERSION OF DIRECTORY 'CAT2' INTO NONKEY-FORMAT STARTED
%  LEA5802 CONVERSION OF DIRECTORY 'CAT2' COMPLETED
%  LEA5803 CONVERSION OF FILE 'SAM1' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'SAM1' WITH 00000002 RECORDS COMPLETED
%  LEA5803 CONVERSION OF FILE 'ISAM1' INTO NONKEY-FORMAT STARTED
%  LEA5804 CONVERSION OF FILE 'ISAM1' WITH 00000002 RECORDS COMPLETED
%  LEA0810 NORMAL TERMINATION OF LEASY CONVERT PROGRAM
```

(1) The link name *FOREIGN* is defined for the reference to the *FOREIGNFILE* file in the FILE statement.

(2) The LEASY-CONVERT utility routine is called.

(3) CON statement: central values are determined for the LEASY-CONVERT run .

OVERWRITE=YES
Target files can overwrite files of the same name.

TO=NONKEY Conversion direction = KEY -> NONKEY
Unless otherwise specified, record truncation is not permitted.

(4) FILE statement: Specifications for selecting a file.

FOREIGN Logical file name (link name of foreign file) of the *FOREIGNFILE* file to be converted.

TYPE=F The file to be converted is a foreign file.

Conversion type: conversion at the same location.

(5) CAT statement: Specification of the catalog to be converted.

CAT1 LEASY catalog name (max. 8 characters).

CATID=SALI The target catalog *CAT1* is set up under the local catalog ID *SALI* .

Conversion type: conversion to another file.

(6) FILE statement: The files in the LEASY catalog *CAT1* are selected.

*ALL All the files in the LEASY catalog *CAT1* are selected.
The following files are converted:

- the master file *CAT1.ISAM* and the associated SI file *CAT1.ISAM-SI*
- the model file *CAT1.MODEL* and its instances *CAT1.MODEL.EXP1* and *CAT1.MODEL.EXP2*
- the master file *CAT1.SAM*

CATID=40SG The target files are generated under catalog ID 40SG.

Conversion type: conversion to another file.

(7) CAT statement: Specification of the catalog to be converted.

CAT2 LEASY catalog name (max. 8 characters).

Conversion type: conversion at the same location.

(8) FILE statement: A file in the LEASY catalog *CAT2* is selected.

SAM1 Logical file name of the file *CAT1.SAM1*.

BLKSIZE=(STD,4) Explicit STD specification of the block size of the target file.

Conversion type: conversion at the same location.

(9) FILE statement: A file in the LEASY catalog *CAT2* is selected.

ISAM1 Logical file name of the file *CAT1.ISAM1*.

Conversion type: conversion at the same location.

(10) END statement: The conversion run is started and the LEASY-CONVERT utility routine is terminated.

A start message is output at the beginning of conversion and an end message at the end for each file selected.

5 LEASY-IOTASK

LEASY is normally generated so that the *LEACONX* module is linked dynamically to each batch/TIAM application program ("linked-in version"). Each of these batch/TIAM application programs opens all the files it requires for processing.

Another type of processing is to generate LEASY as a **separate task**. The *LEACONX* module is no longer in each application program, but (under the name *LEAICNX*) in one or more separate tasks (I/O tasks). LEASY calls of the application program are then no longer passed on by the *LEACON* link module with the aid of subroutine calls (BALR 14,15), but are transferred to *LEAICNX* with the aid of intertask communication, where they are processed and returned to the user. Data transfer between the batch/TIAM application program and the I/O task - and also synchronization - are performed by modules which are stored in the *SYSLNK.LEASY.062.IOH* library. The user interface is retained and is subject to some minor restrictions.

An I/O task is started with the aid of the LEASY-IOTASK utility routine.

The I/O task is a special LEASY application program which accepts jobs and data from batch/TIAM application programs, passes them on to LEASY and sends them back after processing.
The I/O task establishes a connection to the LEASY catalog and opens all the files required for the batch/TIAM application. It waits at an event item for jobs from the application programs. The data (LEASY parameters) is supplied by the batch/TIAM application program in a common memory pool. After a job has been received (LEASY call) the I/O task calls LEASY via the DCAM interface with the transferred parameters. The application program is informed by LEASY via a separate event item after its job has been processed.

5.1 LEASY-IOTASK statements

The statements are described in alphabetical order.

Overview of statements

Statement	Meaning
[*]ARL=arlen	Define the maximum length of the AR area
[*]CAT=[:catid:][$userid.]file-catalog [,suffixname][1]	Access the LEASY catalog
[*]DBL=dblen	Define the length of the DB file allocation
[*]END[2]	Terminate statement input
[*]IOT=maxiotask	Define the maximum number of I/O tasks
[*]KBL=kblen	Define the length of the KB area
[*]KEL=kelen	Define the length of the KE area
[*]OPF=file,mode[3]	Define LEASY files
[*]QUE=$\left\{\begin{array}{l}\text{LIFO}\\\text{FIFO}\end{array}\right\}$	Define queue processing for I/O tasks
[*]USE=maxuser	Define the maximum number of application programs
[*]WAI=time	Define the maximum waiting time for a response from the I/O task

Table 4: 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.

*ARL Define maximum length of AR area

The **ARL* statement specifies the length of the transfer area created in the I/O task for the *AR* area.

Operation	Operands
`[*]ARL[EN]`	`=arlen`

=arlen Specifies the length of the *AR* area in bytes.
12 ≤ *arlen* ≤ 32768
Default value: 2048.

The I/O task creates a transfer buffer with the length *arlen* for the *AR* area. In the case of the *SETL, DLET, LOCK, UNLK, RDIR* and *RHLD* operations, the *AR* area is transferred with the length *arlen* from the application program to the I/O task. In all other cases, the *AR* area is transferred with the length of the current data record between the I/O task and the application program (record length field or fixed-length record). See section “File access via the I/O task”, in the partner manual “LEASY (BS2000) - Program Interface and Strategies”.

The *AR* area must be available with the length specified here in all the programs of the application. All records processed with the aid of the I/O task must not be longer than *arlen*. These two conditions are not checked by the I/O task; they also apply to the **DBL*, **KBL* and **KEL* statements.

The I/O task ascertains the maximum record length of all the files specified in the **OPF* statements. If this maximum record length is greater than *arlen*, a warning is issued specifying the maximum record length, the logical file name of the file in question and also the length (*arlen*) specified by the user. This has no effect on the length of the transferred *AR* area.

*CAT Access LEASY catalog

The **CAT* statement accesses the LEASY catalog for which the I/O task is to be started. This statement is mandatory.

Operation	Operands
`[*]CAT[ALOG]`	`=[:catid:][$userid.]file-catalog[,suffixname]`

catid
Specifies the catalog identifier of the public volume set with the user ID under which the main LEASY task is running. This operand must be specified if the main LEASY task is running under a different ID to I/O task. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-IOTASK was started. The catalog identifier may comprise up to 4 characters, enclosed in colons.

userid
Specifies the user ID under which the LEASY catalog is entered in the public volume set with the catalog identifier *catid*. If *userid* is omitted, LEASY uses the user ID under which LEASY-IOTASK was started.

file-catalog
Specifies the name of the LEASY catalog.

suffixname
Specifies the name of the instances of a model file. This name is valid for all instances of the model file which are accessed. The specified *suffixname* must be identical in all I/O tasks.
If a suffix is specified in the application program for a *CATD* call, it must be the same as the *suffixname* specified in the *CAT* statement.

A *CATD* call is issued internally to LEASY with the catalog name. The user must ensure that the LEASY main task for the specified catalog is already active.

The user must always specify the user ID of the LEASY main task in the *CATD* call for application programs which are running under a different ID from that of the LEASY main task, irrespective of the ID under which the I/O task was started.

*DBL Define length of DB file allocation

The **DBL* statement specifies the length in which the *DB* file allocation (file list) is transferred from the application program to the I/O task.

Operation	Operands
`[*]DBL[EN]`	`=dblen`

=dblen Specifies the length (in bytes) of the *DB* file allocation which is to be transferred between the application program and the I/O task.
12 ≤ *dblen* ≤ 32768.

The length specification is only effective for the *OPFL*, *CLFL*, *OPTR*, *CLTR* and *CINF* operations. The *AR* area is not transferred for these operations. The *DB* file allocation is transferred with a length of 12 bytes for all other operations, provided it was transferred by the user in the LEASY call.

This specification also applies to the transfer of the *CINF* area in a *CINF* operation. The length entry in the *CINF* area is compared with the length specified here. If the area transferred by the user is too large, *IOHLLI10* is supplied by the I/O handler as a return code.

*END Terminate statement input

The **END* statement terminates the input of statements. However, the LEASY-IOTASK utility routine remains loaded.

Operation	Operands
`[*]END[]`	

*IOT Define maximum number of I/O tasks

The **IOT* statement defines the maximum number of I/O tasks of an application which can be active at the same time.

Operation	Operands
`[*]IOT[TASK]`	`=maxiotask`

maxiotask
Specifies the maximum number of I/O tasks which can be active at any one time.
0 < *maxiotask* < 256
Default value: 1.

*KBL Define length of KB area

The **KBL* statement defines the length with which the *KB* area is to be transferred from the application program to the I/O task.

Operation	Operands
`[*]KBL[EN]`	`=kblen`

=kblen
Specifies the length (in bytes) of the KB area which is to be transferred from the application program to the I/O task.
0 < *kblen* < 256
Default value: 255 (maximum for ISAM keys).

*KEL Define length of KE area

The **KEL* statement defines the length with which the *KE* area is to be transferred from the application program to the I/O task.

Operation	Operands
`[*]KEL[EN]`	`=kelen`

=kelen Specifies the length (in bytes) of the *KE* area which is to be transferred from the application program to the I/O task.
0 < *kelen* < 256
Default value: 255 (maximum for ISAM keys).

*OPF Define LEASY files

The **OPF* statement defines the LEASY files (with the appropriate OPEN mode) which are to be processed with the aid of the I/O task. More than one **OPF* statement can be issued.

Operation	Operands
`[*]OPF[L]`	`=file,mode`

file Specifies the logical file name (max. 8 characters).

mode LEASY OPEN mode (1 character).

The following LEASY OPEN modes are permissible:

1	INPUT
2	INPUT,SHARUPD
3	INOUT
4	INOUT,SHARUPD
5	REVERSE
A	OUTIN
B	OUTIN,SHARUPD.

SAM files and foreign files can only be opened for reading. Temporary files are not permitted.

A file list is compiled internally in DB4 format from the individual **OPF* statements. The order of the **OPF* statements in different I/O tasks of the same application must be retained.

*QUE Define queue processing for I/O tasks

The **QUE* statement defines the order in which waiting I/O tasks are to process the LEASY calls of the application programs.

Operation	Operands
[*]QUE[UING]	= { <u>LIFO</u> FIFO }

<u>LIFO</u> If several I/O tasks are ready, the last task to have been entered in the queue receives the next job.

FIFO If several I/O tasks are ready, the task which has been waiting the longest receives the next job.

*USE Define maximum number of application programs

The **USE* statement defines the maximum number of application programs which can work simultaneously with the I/O task.

Operation	Operands
[*]USE[R]	=maxuser

=maxuser Specifies the maximum number of application programs which can work simultaneously with the I/O task.
$0 < maxuser \leq 1800$
Default value: 4.

*WAI Define maximum waiting time for response from I/O task

The **WAI* statement defines the maximum waiting time (in seconds) for a response from the I/O task.

Operation	Operands
`[*]WAI[TING]`	`=time`

=time Specifies the waiting time (in seconds) for a response from the I/O task. This waiting time is accurate to within 60 seconds.
0 < *time* ≤ 43200
Default value: 600.

If a LEASY call is not received from an I/O task within the specified time period, the user receives return code `IOHLLI04`. The LEASY call was not processed because no I/O task was ready to execute the job within the specified time period. The transaction remains open and can be continued when an I/O task is ready again. This can be done by starting an additional I/O task.

5.2 Starting and terminating an I/O task

Starting an I/O task

An I/O task is started by means of the LEASY-IOTASK utility routine. The I/O task creates the environment which the application programs require in order to access files in the LEASY catalog with the aid of the I/O handler.
An I/O task is started in the same way as a main task. This can be carried out from a terminal or by means of an ENTER procedure. If the I/O task is started from a terminal, the terminal is blocked. It is therefore preferable to start the I/O task as a batch task by means of an ENTER procedure.
An I/O task can also be started if application programs have already been started.

Example of an ENTER procedure for I/O tasks

```
/SET-LOGON-PAR JOB-NAME=IOTASK
/START-LEASY-IOTASK
CAT=LCAT
OPF=LOHN1,4
.
.
.
OPF=LOHN2,2
USE=30
ARL=1000
IOT=3
END
/EXIT-JOB
```

Exactly one I/O task is activated with each *ENTER-JOB* command which is used to start this procedure. The number of I/O tasks specified in the *IOT* statement is the maximum number which can be started. The user must ensure that the statements are identical in all the I/O tasks. It is particularly important that the order of *OPF* statements is identical.

Terminating an I/O task

An I/O task is terminated by means of the *IOTE* statement of the LEASY-MASTER utility routine. An I/O task can also be terminated even if application programs are still running, provided sufficient I/O tasks to process the jobs are still active. If an I/O task is terminated with errors during the processing of a job, the transaction which has just been processed is reset. The user receives return code *IOHLLI05*.

5.3 Job variable specifying the number of active I/O tasks

The introduction of a job variable informing the user of the number of active I/O tasks enables programs and procedures to be controlled in accordance with the number of active I/O tasks. This also permits several I/O tasks to be activated under user control.

User action

The user must take the following measures (see the "Job Variables" manual):

1. Catalog a job variable using the *CREATE-JV* command.

   ```
   /CREATE-JV JV=jvname
   ```

   The job variable name *jvname* is freely selectable.

2. Assign the link name *LEAIOST* to the job variable using the *SET-JV-LINK* command.

   ```
   /SET-JV-LINK LINK=LEAIOST,JV=jvname
   ```

   Following these measures, the user can retrieve the information passed to the job variable by LEASY via the *SHOW-JV* command or the *GETJV* macro.

   The user must ensure that the same job variable is assigned/queried in all I/O tasks. LEASY does not check whether a job variable has been assigned.

Return information from LEASY

LEASY passes the following 10-byte information in the job variable:

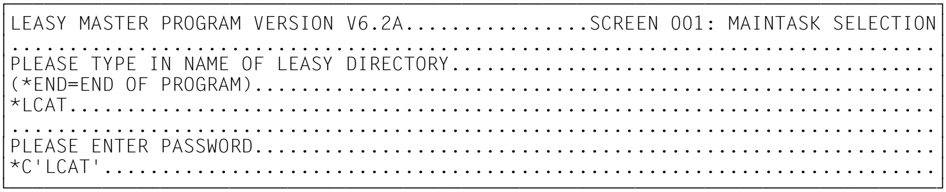
Bytes	1-10
Content	nnn-ACTIVE where *nnn* is the number of active I/O tasks

5.4 Example

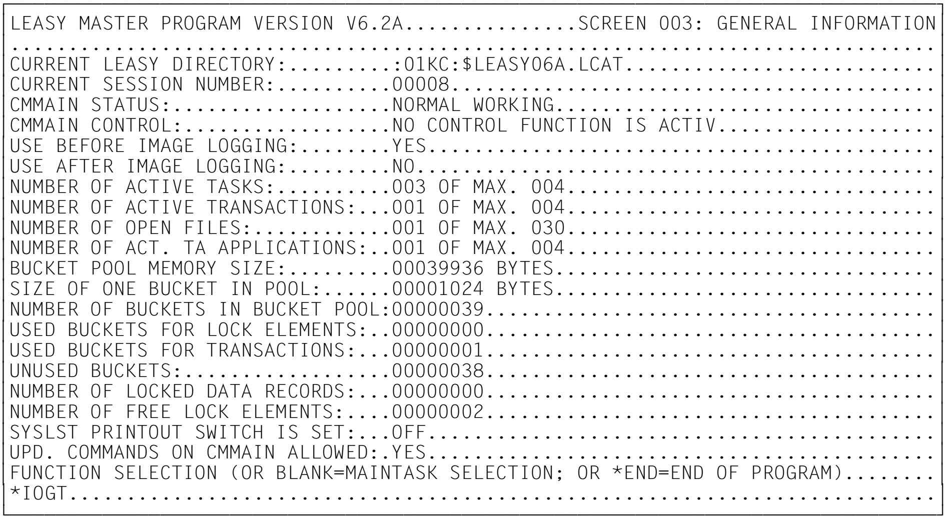
An example of operation with the I/O handler is provided below.

```
/CREATE-JV WATCHIO ---------------------------------------------------------------- (1)
/ENTER-JOB E.MTSK,CPU-LIMIT=50 ---------------------------------------------------- (2)
%  JMS0066 JOB 'MAINTASK' ACCEPTED ON '06-05-10' AT '11:04', TSN = 3432
/SHOW-USER-STATUS
NAME      TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
SCHNEIDR 3187 3 DIALOG1  0 210      0.9060    9000 M4212
MAINTASK 3432 2 BATCH    9 230      0.0410      50 M4212 -------------------------- (3)
%  SPS0171 NO SPOOLOUT JOB PRESENT
/SHOW-FILE E.IOTASK
                                        |
/SET-LOGON-PAR JOB-NAME=IOTASK          |
/SET-JV-LINK LINK=LEAIOST,JV=WATCHIO    |
/START-LEASY-IOTASK                     |
CAT=LCAT                                | ------------------------------------------ (4)
DBL=40                                  }
OPF=MITABDAT,4                          |
IOT=3                                   |
END                                     |
/EXIT-JOB                               |
*END
%  SHO0500 ':01KC:$LEASY06A.E.IOTASK' CLOSED
/ENTER-JOB E.IOTASK,CPU-LIMIT=50--------------------------------------------------- (5)
%  JMS0066 JOB 'IOTASK' ACCEPTED ON '06-05-10' AT '11:05', TSN = 3439
/SHOW-USER-STATUS
NAME      TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
SCHNEIDR 3187 3 DIALOG1  0 210      0.9589    9000 M4212
IOTASK   3439 2 BATCH    9 230      0.0395      50 M4212 -------------------------- (6)
MAINTASK 3432 2 BATCH    9 230      0.0981      50 M4212
%  SPS0171 NO SPOOLOUT JOB PRESENT
/SHOW-JV JV=WATCHIO --------------------------------------------------------------- (7)
%001-ACTIVE ----------------------------------------------------------------------- (8)
/ENTER-JOB E.IOTASK,CPU-LIMIT=50--------------------------------------------------- (9)
%  JMS0066 JOB 'IOTASK' ACCEPTED ON '06-05-10' AT '11:06', TSN = 3449
/SHOW-USER-STATUS
NAME      TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
SCHNEIDR 3187 3 DIALOG1  0 210      1.2201    9000 M4212
IOTASK   3439 2 BATCH    9 230      0.1305      50 M4212
IOTASK   3449 2 BATCH    9 230      0.0835      50 M4212 ------------------------- (10)
MAINTASK 3432 2 BATCH    9 230      0.0981      50 M4212
%  SPS0171 NO SPOOLOUT JOB PRESENT
/SHOW-JV JV=WATCHIO--------------------------------------------------------------- (11)
%002-ACTIVE ---------------------------------------------------------------------- (12)
/ENTER-JOB E.IOTASK,CPU-LIMIT=50-------------------------------------------------- (13)
%  JMS0066 JOB 'IOTASK' ACCEPTED ON '06-05-10' AT '11:08', TSN = 3579
/SHOW-USER-STATUS
NAME      TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
SCHNEIDR 3187 3 DIALOG1  0 210      2.0005    9000 M4212
IOTASK   3439 2 BATCH    9 230      0.1305      50 M4212
IOTASK   3449 2 BATCH    9 230      0.1270      50 M4212
MAINTASK 3432 2 BATCH    9 230      0.0981      50 M4212
IOTASK   3579 2 BATCH    9 230      0.1276      50 M4212 ------------------------- (14)
%  SPS0171 NO SPOOLOUT JOB PRESENT
/SHOW-JV JV=WATCHIO -------------------------------------------------------------- (15)
%003-ACTIVE ---------------------------------------------------------------------- (16)
/SHOW-USER-STATUS
NAME      TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
SCHNEIDR 3187 3 DIALOG1  0 210      2.1602    9000 M4212
IOTASK   3439 2 BATCH    9 230      0.1305      50 M4212
IOTASK   3449 2 BATCH    9 230      0.1270      50 M4212
MAINTASK 3432 2 BATCH    9 230      0.0981      50 M4212
IOTASK   3579 2 BATCH    9 230      0.1276      50 M4212
TASK1    3582 3 DIALOG1  0 210      1.6925    9000 M4212 ------------------------- (17)
```

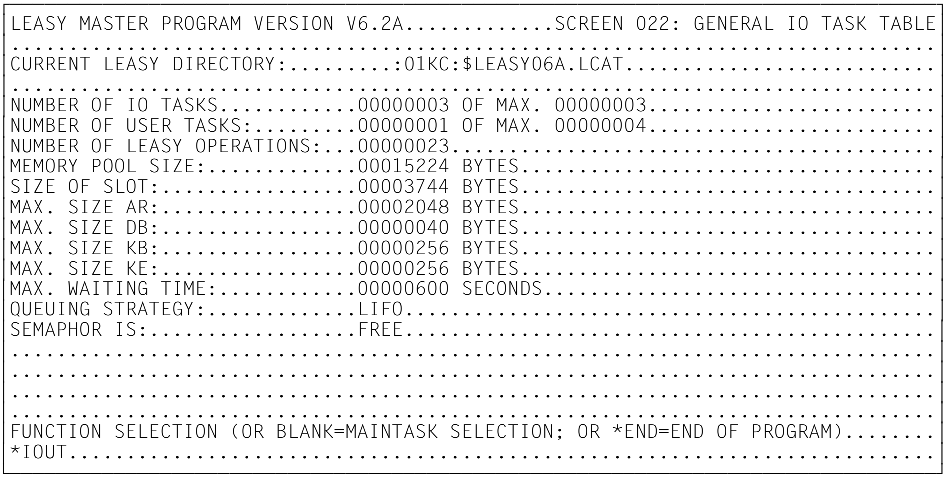
```
/START-LEASY-MASTER ─────────────────────────────────────────────────────────── (18)
%  BLS0523 ELEMENT 'MASTER', VERSION '06.2A' FROM LIBRARY
   ':20SL:$TSOS.SYSPRG.LEASY.062' IN PROCESS
%  BLS0524 LLM 'LEASY-MASTER', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0501 LEASY MASTER PROGRAM VERSION V6.2A STARTED
```



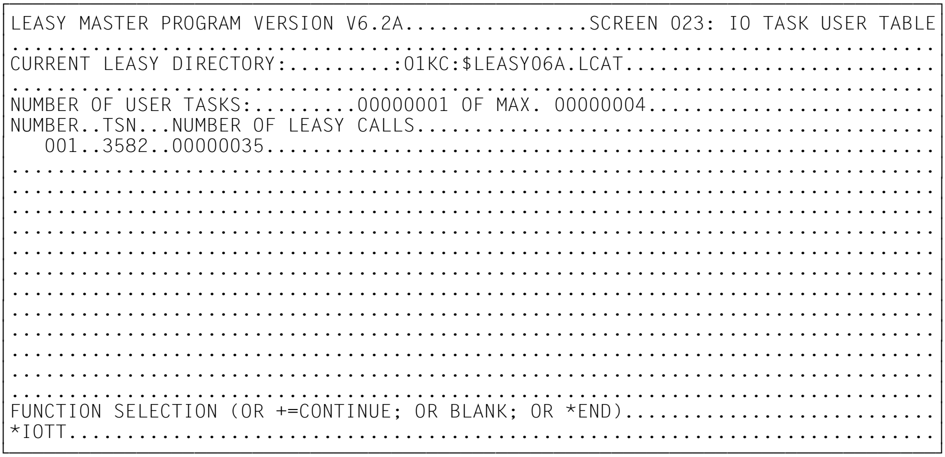
```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 001: MAINTASK SELECTION
................................................................................
PLEASE TYPE IN NAME OF LEASY DIRECTORY..........................................
(*END=END OF PROGRAM)...........................................................
*LCAT...........................................................................   - (19)
................................................................................
PLEASE ENTER PASSWORD...........................................................
*C'LCAT'........................................................................   - (20)
```



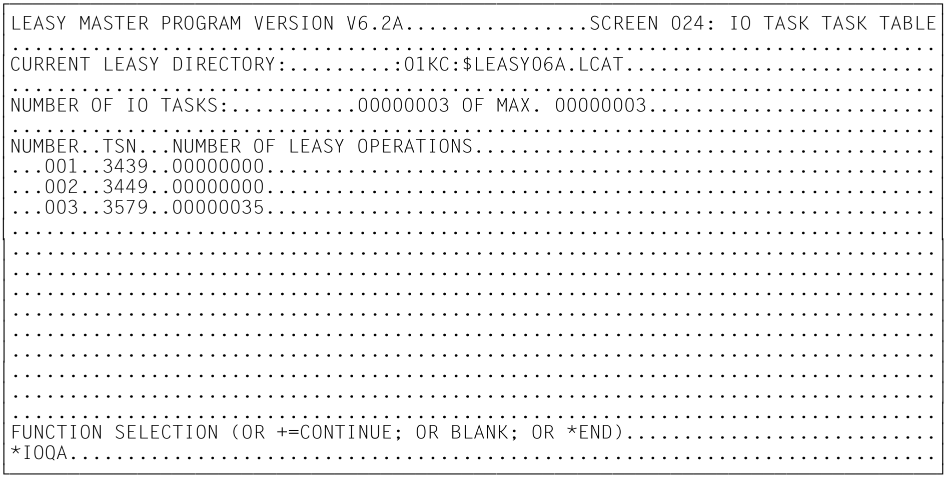
```
LEASY MASTER PROGRAM VERSION V6.2A...............SCREEN 003: GENERAL INFORMATION   - (21)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
CURRENT SESSION NUMBER:..........00008..........................................
CMMAIN STATUS:...................NORMAL WORKING.................................
CMMAIN CONTROL:..................NO CONTROL FUNCTION IS ACTIV...................
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........NO.............................................
NUMBER OF ACTIVE TASKS:..........003 OF MAX. 004................................
NUMBER OF ACTIVE TRANSACTIONS:...001 OF MAX. 004................................
NUMBER OF OPEN FILES:............001 OF MAX. 030................................
NUMBER OF ACT. TA APPLICATIONS:..001 OF MAX. 004................................
BUCKET POOL MEMORY SIZE:.........00039936 BYTES.................................
SIZE OF ONE BUCKET IN POOL:......00001024 BYTES.................................
NUMBER OF BUCKETS IN BUCKET POOL:00000039.......................................
USED BUCKETS FOR LOCK ELEMENTS:..00000000.......................................
USED BUCKETS FOR TRANSACTIONS:...00000001.......................................
UNUSED BUCKETS:..................00000038.......................................
NUMBER OF LOCKED DATA RECORDS:...00000000.......................................
NUMBER OF FREE LOCK ELEMENTS:....00000002.......................................
SYSLST PRINTOUT SWITCH IS SET:...OFF............................................
UPD. COMMANDS ON CMMAIN ALLOWED:.YES............................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOGT...........................................................................   - (22)
```



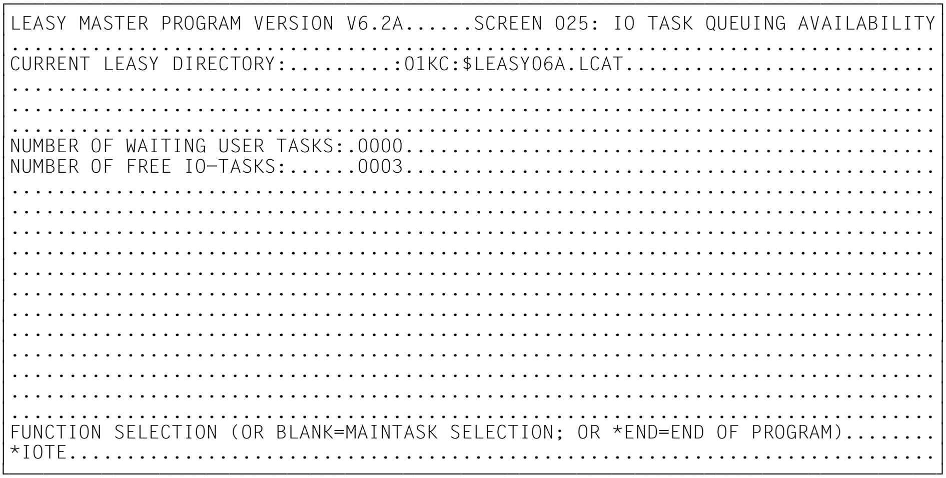
```
LEASY MASTER PROGRAM VERSION V6.2A.............SCREEN 022: GENERAL IO TASK TABLE   - (23)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
NUMBER OF IO TASKS............00000003 OF MAX. 00000003.........................
NUMBER OF USER TASKS:.........00000001 OF MAX. 00000004.........................
NUMBER OF LEASY OPERATIONS:...00000023..........................................
MEMORY POOL SIZE:.............00015224 BYTES....................................
SIZE OF SLOT:.................00003744 BYTES....................................
MAX. SIZE AR:.................00002048 BYTES....................................
MAX. SIZE DB:.................00000040 BYTES....................................
MAX. SIZE KB:.................00000256 BYTES....................................
MAX. SIZE KE:.................00000256 BYTES....................................
MAX. WAITING TIME:............00000600 SECONDS..................................
QUEUING STRATEGY:.............LIFO..............................................
SEMAPHOR IS:..................FREE..............................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOUT...........................................................................   - (24)
```



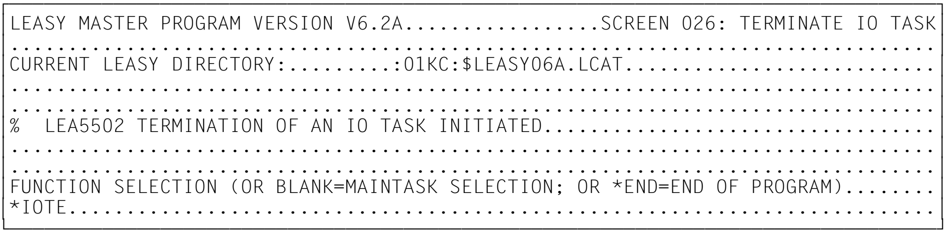
```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 023: IO TASK USER TABLE  - (25)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
NUMBER OF USER TASKS:.........00000001 OF MAX. 00000004.........................
NUMBER..TSN...NUMBER OF LEASY CALLS.............................................
   001..3582..00000035..........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*IOTT...........................................................................  - (26)
```



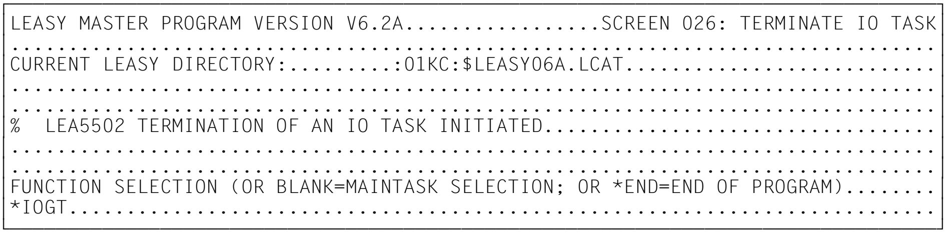
```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 024: IO TASK TASK TABLE  - (27)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
NUMBER OF IO TASKS:...........00000003 OF MAX. 00000003.........................
................................................................................
NUMBER..TSN...NUMBER OF LEASY OPERATIONS........................................
...001..3439..00000000..........................................................
...002..3449..00000000..........................................................
...003..3579..00000035..........................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*IOQA...........................................................................  - (28)
```



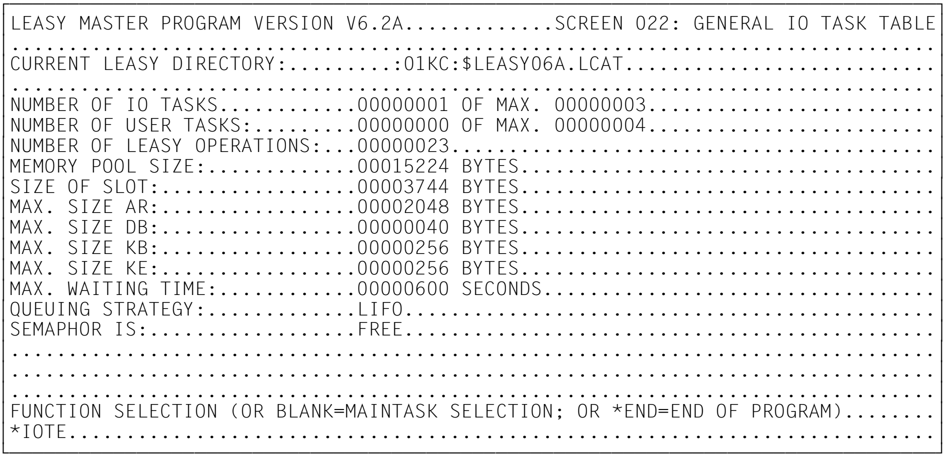
```
LEASY MASTER PROGRAM VERSION V6.2A......SCREEN 025: IO TASK QUEUING AVAILABILITY  - (29)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
................................................................................
................................................................................
NUMBER OF WAITING USER TASKS:.0000..............................................
NUMBER OF FREE IO-TASKS:......0003..............................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOTE...........................................................................  - (30)
```



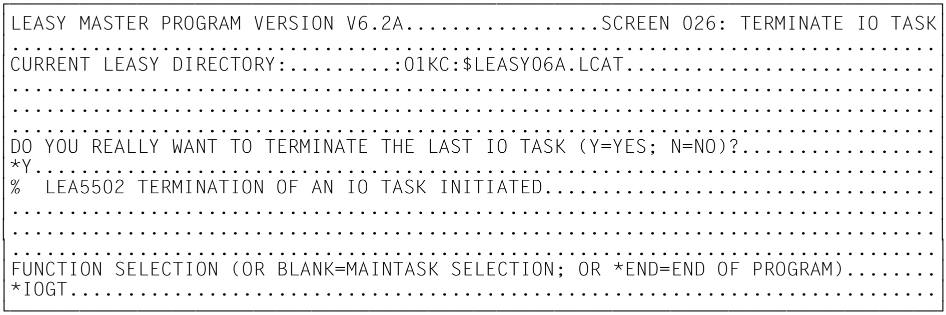
```
LEASY MASTER PROGRAM VERSION V6.2A.................SCREEN 026: TERMINATE IO TASK
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
................................................................................
%  LEA5502 TERMINATION OF AN IO TASK INITIATED..................................  - (31)
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOTE...........................................................................  - (32)
```



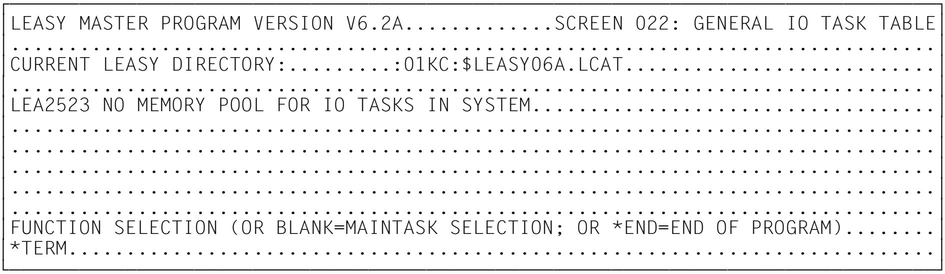
```
LEASY MASTER PROGRAM VERSION V6.2A.................SCREEN 026: TERMINATE IO TASK
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
................................................................................
%  LEA5502 TERMINATION OF AN IO TASK INITIATED..................................  - (33)
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOGT...........................................................................  - (34)
```



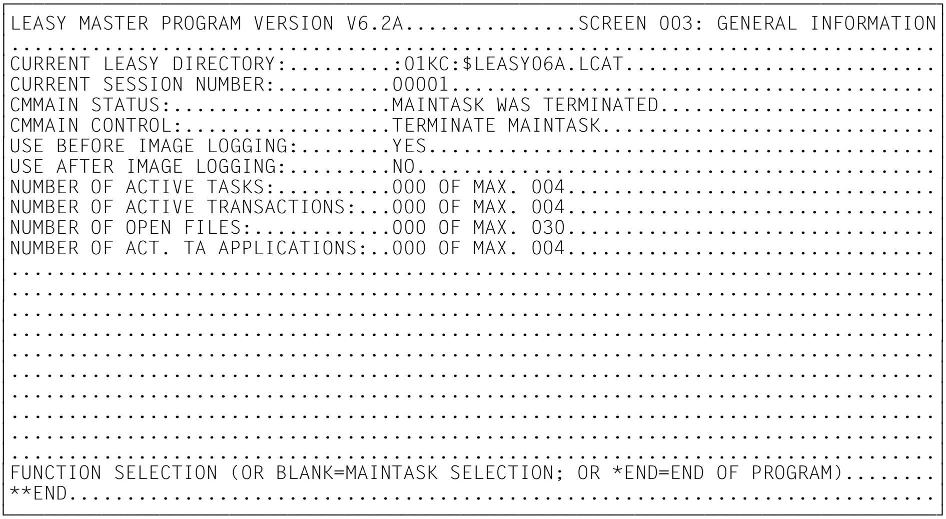
```
LEASY MASTER PROGRAM VERSION V6.2A.............SCREEN 022: GENERAL IO TASK TABLE  - (35)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
NUMBER OF IO TASKS............00000001 OF MAX. 00000003.........................
NUMBER OF USER TASKS:.........00000000 OF MAX. 00000004.........................
NUMBER OF LEASY OPERATIONS:...00000023..........................................
MEMORY POOL SIZE:.............00015224 BYTES....................................
SIZE OF SLOT:.................00003744 BYTES....................................
MAX. SIZE AR:.................00002048 BYTES....................................
MAX. SIZE DB:.................00000040 BYTES....................................
MAX. SIZE KB:.................00000256 BYTES....................................
MAX. SIZE KE:.................00000256 BYTES....................................
MAX. WAITING TIME:............00000600 SECONDS..................................
QUEUING STRATEGY:.............LIFO..............................................
SEMAPHOR IS:..................FREE..............................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOTE...........................................................................  - (36)
```



```
LEASY MASTER PROGRAM VERSION V6.2A.................SCREEN 026: TERMINATE IO TASK
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
................................................................................
................................................................................
DO YOU REALLY WANT TO TERMINATE THE LAST IO TASK (Y=YES; N=NO)?.................
*Y..............................................................................  - (37)
%  LEA5502 TERMINATION OF AN IO TASK INITIATED..................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*IOGT...........................................................................  - (38)
```



```
LEASY MASTER PROGRAM VERSION V6.2A.............SCREEN 022: GENERAL IO TASK TABLE  - (39)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
................................................................................
LEA2523 NO MEMORY POOL FOR IO TASKS IN SYSTEM...................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*TERM...........................................................................  - (40)
```



```
LEASY MASTER PROGRAM VERSION V6.2A...............SCREEN 003: GENERAL INFORMATION  - (41)
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$LEASY06A.LCAT...........................
CURRENT SESSION NUMBER:..........00001..........................................
CMMAIN STATUS:...................MAINTASK WAS TERMINATED........................
CMMAIN CONTROL:..................TERMINATE MAINTASK.............................
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........NO.............................................
NUMBER OF ACTIVE TASKS:..........000 OF MAX. 004................................
NUMBER OF ACTIVE TRANSACTIONS:...000 OF MAX. 004................................
NUMBER OF OPEN FILES:............000 OF MAX. 030................................
NUMBER OF ACT. TA APPLICATIONS:..000 OF MAX. 004................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
**END...........................................................................  - (42)
```

```
%  LEA0510 NORMAL TERMINATION OF LEASY MASTER PROGRAM
```

Explanation

(1) Catalog a job variable for the number of active I/O tasks.

(2) Start the main task with the ENTER procedure *E.MTSK*.

(3) The main task has been started.

(4) Log of the procedure file *E.IOTASK*, which starts an I/O task.

- Job variable *WATCHIO* is assigned to the I/O task via the link name **LEAIOST*.
- LEASY-IOTASK is started in the procedure file.
- The I/O task is to be started for the LEASY catalog *LCAT*.
- The *DB* file allocation is to be transferred with a length of 40 bytes.
- The LEASY file *MITABDAT* is to be processed in the LEASY OPEN mode *INOUT,SHARUPD*.
- No more than three I/O tasks may be active.
- Four user programs can work with I/O tasks concurrently (default).

(5) Start the first I/O task with the ENTER procedure *E.IOTASK*.

(6) The first I/O task has been started.

(7) Job variable *WATCHIO* is queried.

(8) As soon as the first I/O task has been initialized, one active I/O task is indicated in the job variable.

(9) Start the second I/O task with the ENTER procedure *E.IOTASK*.

(10) The second I/O task has been started.

(11) Job variable *WATCHIO* is queried.

(12) As soon as the second I/O task has been initialized, two active I/O tasks are indicated in the job variable.

(13) Start the third I/O task with the ENTER procedure *E.IOTASK*.

(14) The third I/O task has been started.

(15) Job variable *WATCHIO* is queried.

(16) As soon as the third I/O task has been initialized, three active I/O tasks are indicated in the job variable.

(17) A user program has been started.

(18) Start the program LEASY-MASTER.

(19) Assign the LEASY catalog *LCAT*.

(20) The password *C'LCAT'* is requested.

(21) The general information of the common memory CMMAIN is displayed. Three tasks and one transaction are active.

(22) Request the general I/O task table.

(23) The general I/O task table is displayed. Three I/O tasks and one user task are active.

(24) The user desires further information on the user task.

(25) The I/O task user table is displayed.

(26) The user requests further information on the I/O tasks.

(27) The I/O task table is displayed.

(28) The user wants to see the I/O task queue assignment.

(29) The I/O task queuing availability is displayed.

(30) The user terminates an I/O task.

(31) Acknowledgment of initiation of I/O task termination.

(32) Another I/O task is to be terminated.

(33) Acknowledgment of initiation of I/O task termination.

(34) The user again wishes to see the general I/O task table.

(35) The general I/O task table is displayed. No user task is active.

(36) The last I/O task is to be terminated.

(37) For safety's sake LEASY-MASTER does not initiate termination until the user confirms by entering *Y*.

(38) Once again, the general I/O task table is requested by the user.

(39) The general I/O task table is displayed. No I/O task is active.

(40) The user terminates the LEASY-MAINTASK utility routine.

(41) The general information of the common memory CMMAIN is displayed.

(42) The user terminates the LEASY-MASTER utility routine.

6 LEASY-LOADSI

The LEASY-LOADSI utility routine is used for the creation, insertion and deletion of SI pointers in secondary index files (SI files).

SI files are created using the specifications of the secondary indices (SI name, key position, key length, distances, record type interdependencies, multiplication factor and index suppression byte) from the LEASY catalog. Therefore only those secondary indices whose attributes have already been defined with LEASY-CATALOG may be used.

Inputs to LEASY-LOADSI: special features

The LEASY-LOADSI utility routine supports prompted inputting (guided dialog), but can also be implemented in batch mode.

After LEASY-LOADSI has been loaded the following message is displayed:

```
LEA0201  LEASY PROGRAM FOR SECONDARY INDICES LOADSI VERSION V6.2A STARTED
```

Prompting then begins. Each input is requested by means of a LEASY message. The maximum length of an input line (incl. all characters and spaces etc.) is:

Catalog specifications	255 characters
Primary file specifications	255 characters
Specifications for secondary keys	3000 characters

Record chaining is used when entering specifications for secondary keys.

A comment statement can be entered at any point within the program (except within the input record string):

```
*COM[any text]
```

6.1 Assigning the LEASY catalog

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

Prompt

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

Possible responses

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

*END
End of the LEASY-LOADSI run.

*HALT
End of the LEASY-LOADSI run.

where

catid
Specifies the catalog identifier of the public volume set which contains the LEASY catalog. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-LOADSI was started. The catalog identifier may comprise up to 4 characters, enclosed in colons.

userid
Specifies the user ID under which the LEASY catalog is entered in the public volume set with the catalog identifier *catid*. If *userid* is omitted, LEASY uses the user ID under which LEASY-LOADSI was started.

file-catalog
Specifies the name of the LEASY catalog.

After processing of a LEASY catalog is completed (**END*), the prompt is repeated, thus allowing any number of LEASY catalogs to be processed in one run.

i The **END* and **HALT* statements differ as follows:

**END* always leads to the superordinate level and, at the highest level, to program termination. **HALT* leads to immediately program termination, regardless of level.

6.2 Assigning the primary file

Once a LEASY catalog has been assigned as requested by the prompt described in the previous section, a primary file from this catalog must be specified:

Prompt

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

Possible responses

file[.suffix] [PAD=pad][,SIZE=size]
Specifies the primary file.

*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.

where

file Logical file name of a master file or temporary file.

file.suffix Logical file name of an instance of a model file group.

PAD=pad Percentage of blocks to be reserved for future extensions on generation of an SI file.
$0 \le pad \le 99$.
Default value: PAD=15.

SIZE=size Estimated number of records in the primary file. This value serves as an auxiliary operand for the sort run executed within the program and should only be specified if a reasonably accurate estimate is possible.
$1 \le size \le 2147483639$.

The request to specify a file is repeated whenever processing of a primary file has been completed. This means that any number of primary files may be processed for each LEASY catalog.

As temporary files are not generated for the particular task until the time of execution (when the name structure for temporary files comes into play), it is practical to use LEASY-LOADSI on temporary files in the user task at the time of execution only.

6.3 Specifications for secondary index management

If, in reply to the prompt described in the previous section, the name of a primary file was entered, specifications for secondary index management are now required.

Prompt

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

Possible responses

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.

where

keyname Name of the secondary key to be processed, which can be up to 8 characters long.

Any number of secondary keys may be specified but they must have been already defined for the assigned file by means of the LEASY-CATALOG utility routine.

It should be noted that no SI name is required for *NEW*, but that one is required for *ADD* and *DEL*. If SI names are specified, only these names are processed.

NEW[ ] Completely new secondary index pointers are created in the SI file.
Any existing entries are first deleted.
If no secondary key name has been specified, pointers are created for all defined secondary keys.
Application examples:

– The SI file of a primary file, which is new to the catalog, is created for the first time.

– The secondary key pointers of an existing SI file are renewed because the automatic updating of the SI file was switched off (see section “Handling of secondary index definitions” on page 57ff).

If secondary key names are specified, pointers are created for these keys only.

ADD[ ] Creates additional secondary index pointers for the keys specified in *keyname*.
There must not already be any existing pointers for *keyname*.

DEL[ ] Deletes existing secondary index pointers for *keyname*.

The following specifications are equivalent:

```
NEW A, B, C, D     or       NEW A           or       NEW A
                            ADD B, C, D              ADD B
                                                     ADD C
                                                     ADD D
```

The only difference is that specifications in one line are processed together (e.g. in an internal *SORT*). Therefore, in the case of primary files containing a large number of records, the third format is preferable, since *SORT* requires less main memory space. The first format is more suitable for short primary files as it provides faster processing.

The internal sort run can be speeded up if *SORTWK* areas are specified. In the case of large files, the sort run is only possible if tape devices are made available. BS2000 SORT from SORTLIB is used as the sort program (see the "SORT (BS2000/OSD)" manual).

6.4 Last-page pointer as end-of-file criterion

In PAM and DAM files, the LEASY-LOADSI utility routine reads all pages with a valid CFID (Coded File ID) up to the last half page (LHP, see the "Introductory Guide to DMS" manual) and up to the file size specified in *filesize*.

However, the DMS macro *PAM SETLPP* (see the "DMS Macros" manual) can cause the last-page pointer to move towards the start of the file, e.g. in the course of reorganizations.

LEASY-LOADSI uses the LPP (last-page pointer) as the criterion for the end of the file. Normally the LPP points to the last PAM page of the file to which data was written, i.e. to the last page with the CFID belonging to this file.

In the case of files with *BLKCTRL=DATA* the DMS macro *PAM SETLPP* can only be used to position the LPP to the end of the block, but with files with *BLKCTRL=PAMKEY* it is also possible to position to PAM pages within the blocks.

6.5 Error handling

Violations of syntax or semantics in the input data lead to error messages. If serious errors make it seem pointless to continue with processing, current processing is interrupted and the corresponding specifications skipped. If, for example, a DMS error occurs during reading of a primary file, all secondary specifications for this file are skipped up to *END*. If *END* is omitted, the results become unpredictable.

Secondary index specifications which do not begin with *NEW, ADD, DEL* or *END* are rejected.

Some errors lead to program abortion. This may mean that the SI file is only partially created and that the primary file, LEASY catalog and SI file are inconsistent. In such cases a check should be made to see whether deletion of the last secondary index processed is sufficient, or whether the whole SI file must be recreated.

The LEASY-LOADSI utility routine is not aborted when an unauthorized pointer duplicate is detected; it merely informs the user of the detected duplicate.

6.6 Example

```
/SET-TASKLIB LIB=$TSOS.SORTLIB
/START-LEASY-LOADSI─────────────────────────────────────────────────────────────── (1)
% BLS0523 ELEMENT 'LOADSI', VERSION '06.2A' FROM LIBRARY
  ’:20SL:$TSOS.SYSPRG.LEASY.062’ IN PROCESS
%  BLS0524 LLM 'LEASY-LOADSI', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0201 LEASY PROGRAM FOR SECONDARY INDICES LOADSI VERSION V6.2A STARTED
%  LEA0001 PLEASE TYPE IN NAME OF LEASY DIRECTORY
*LCAT ───────────────────────────────────────────────────────────────────────────── (2)
%  LEA0203 PLEASE TYPE IN FILE SPECIFICATION
*MITABDAT PAD=15 ────────────────────────────────────────────────────────────────── (3)
%  LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
*NEW ────────────────────────────────────────────────────────────────────────────── (4)
%  SRT1001 12:31:48/000000.00 SORT/MERGE STARTED, VERSION 7.8A00/BS2000V14.0
%  SRT1018 INSERTED INPUT RECORDS:.............................18 ┐
%  SRT1017 RECORDS TO BE SORTED/MERGED:........................18 ├──────────  (5)
%  SRT1021 SORTED/MERGED RECORDS:..............................18 │
%  SRT1023 DELETED OUTPUT RECORDS:.............................18 ┘
%  SRT1002 10:54:56/000000.00 SORT/MERGE COMPLETED
%  LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
*END ────────────────────────────────────────────────────────────────────────────── (6)
%  LEA0203 PLEASE TYPE IN FILE SPECIFICATION
*MITDAT1 ────────────────────────────────────────────────────────────────────────── (7)
%  LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
*NEW ABT ────────────────────────────────────────────────────────────────────────── (8)
%  SRT1001 12:31:48/000000.00 SORT/MERGE STARTED, VERSION 7.8A00/BS2000V14.0
%  SRT1018 INSERTED INPUT RECORDS:..............................9
%  SRT1017 RECORDS TO BE SORTED/MERGED:.........................9
%  SRT1021 SORTED/MERGED RECORDS:...............................9
%  SRT1023 DELETED OUTPUT RECORDS:..............................9
%  SRT1002 10:56:42/000000.00 SORT/MERGE COMPLETED
%  LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
*ADD NAME ───────────────────────────────────────────────────────────────────────── (9)
%  SRT1001 12:31:48/000000.00 SORT/MERGE STARTED, VERSION 7.8A00/BS2000V14.0
%  SRT1018 INSERTED INPUT-RECORDS:..............................9
%  SRT1017 RECORDS TO SORT/MERGE:...............................9
%  SRT1021 SORTED/MERGED RECORDS:...............................9
%  SRT1023 DELETED OUTPUT RECORDS:..............................9
%  SRT1002 10:57:33/000000.00 SORT/MERGE COMPLETED
%  LEA0204 PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX
*END ────────────────────────────────────────────────────────────────────────────── (10)
%  LEA0203 PLEASE TYPE IN FILE SPECIFICATION
**END ───────────────────────────────────────────────────────────────────────────── (11)
%  LEA0001 PLEASE TYPE IN NAME OF LEASY DIRECTORY
*TESTCAT ────────────────────────────────────────────────────────────────────────── (12)
%  LEA0203 PLEASE TYPE IN FILE SPECIFICATION
**END ───────────────────────────────────────────────────────────────────────────── (13)
%  LEA0001 PLEASE TYPE IN NAME OF LEASY DIRECTORY
**END ───────────────────────────────────────────────────────────────────────────── (14)
%  LEA0210 NORMAL TERMINATION OF LOADSI
/
```

Explanation

(1) The LEASY-LOADSI utility routine is called.

(2) The LEASY catalog *LCAT* is assigned.

(3) The primary file *MITABDAT* is assigned. When creating the corresponding SI file, 15% of the occupied blocks are to be reserved for future extensions.

(4) New secondary index pointers are created in the SI file; existing entries are first deleted.

(5) 18 records are generated when the SI pointers are created.

(6) End of specifications for the assigned file *MITABDAT*.

(7) The SI pointers for the primary file *MITDAT1* are now to be processed.

(8) The SI pointers for the secondary key *ABT* are recreated; the existing pointers are first deleted.

(9) The SI pointers for the secondary key *NAME* are added to the SI file.

(10) End of specifications for the assigned file *MITDAT1*.

(11) End of specifications for the assigned LEASY catalog *LCAT*.

(12) LEASY catalog *TESTCAT* is assigned.

(13) End of specifications for the LEASY catalog *TESTCAT*.

(14) LEASY-LOADSI is terminated.

7 LEASY-MAINTASK

The main LEASY task must have been started and a LEASY session initialized before user programs can use LEASY to access files in a LEASY file catalog. The main LEASY task, which has its own TSN, sets up and maintains the environment required by the LEASY system. The LEASY-MAINTASK utility routine is used to start the main LEASY task.

Leasy-Session

Each LEASY session is given a session number, which is incremented by 1 each time the main task is loaded successfully.

Within each session the individual transactions are numbered consecutively, beginning at 1. Thus each transaction is uniquely identified by

- the session number and
- the transaction number

When the main LEASY task terminates, the LEASY session is also terminated. It should normally be closed using the *TERM*, *CLOS* or *SHUT* statements of the LEASY-MASTER utility routine.

If a warm start has to be performed when loading a new session - resetting unfinished transactions following a system crash - file updates during the rollback are assigned to the old session. The new session begins only after error-free execution of the warm/cold start.

Bucket management

A minimum number of memory units (buckets) is protected against lock element list requests from both BATCH/TIAM tasks and openUTM tasks. The object of this measure is to keep a sufficient area available for transaction elements, even during periods of extensive lock element list requests.

The minimum number of buckets to be held free against lock element list requests from openUTM tasks varies according to the maximum number of transactions specified in the **TRA* statement, see page 148.

The multiplication factor for defining the maximum number of buckets to be held free against lock element list requests from BATCH/TIAM tasks can be set via the **MFB* statement, see page 143.

A batch/TIAM task is terminated with a reset of the last open transaction when it receives the second LS12 return code (overflow of the transaction element area or the lock element list area while trying to create a new lock element). With interactive processes, the return code is passed to the application.

7.1 Inputs to LEASY-MAINTASK: special features

The LEASY-MAINTASK utility routine can be executed in interactive or batch mode. Except where otherwise specified, the following notes apply to both modes.

- Statements may be input in any order, but the input must be terminated with **END*.
- The **CAT* and **END* statements are mandatory; all other statements have default values.
- Chaining of input records is permitted for all statements. The maximum input record length is 1024 bytes.
- Incorrectly input statements can be corrected in interactive mode, but this is not possible in batch mode.
- If incorrect inputs are not corrected in interactive mode, or if incorrect statements are entered in batch mode, further inputs are checked but when **END* is input the initialization run is identified as being errored and is terminated. The statements are not executed.
- The LEASY-MAINTASK utility routine is terminated if an error occurs while the session is being started.
- An equals sign (=) must be inserted between the operation and the operand (e.g. **CAT=TESTCAT*).

7.2 LEASY-MAINTASK statements

System initialization is influenced by the statements of the LEASY-MAINTASK utility routine.

The statements are listed below in alphabetical order.

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=` pamblock-number (pamblock-number,increment)	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

Table 5: Overview of statements (LEASY-MAINTASK)

Statement	Meaning
[*]CAT=file-catalog	Access LEASY catalog
[*]COM [text]	Insert comment
[*]DES= {Y N}	Specify processing of memory space on deletion
[*]END	Terminate statement input
[*]FAA[]= {Y[] N[]}	Always release AIM file generations
[*]FIL=files	Specify number of files
[*]KEY=keylen	Specify key length
[*]LOG= {B[] A[][,{M[] R[]}][,K[]] [,P[]] Y[][,{M[] R[]}] [,P[]] N[][,K[]]}	Determine save method
[*]MEM=mem	Define size of common memory
[*]MFB=mfact	Specify the multiplication factor for defining the maximum number of buckets to be held free against lock element list requests from batch or TIAM tasks
[*]MTT=wtime	Define wait time for 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
[*]STA= {C 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
[*]USE= {N C R}	Define state of common memory

Table 5: Overview of statements (LEASY-MAINTASK)

*ACA Specify public volume set for AIM file

The **ACA* statement specifies the public volume set on which the AIM file generation group is to be generated.

Operation	Operands
`[*]ACA[TID]`	`=acatid`

=acatid Catalog identifier of the public volume set. *acatid* may comprise up to 4 characters.

If the **ACA* statement is missing, the AIM file generation group is placed on the same public volume set as the LEASY catalog.

If a LEASY catalog which has already been generated is converted to MPVS, and if the AIM file generation group is placed on a different public volume set to the LEASY catalog, the **ACA* statement is mandatory when LEASY-MAINTASK is started for the first time.

The **ACA* statement is not executed if *CID=N* has been specified in the *CAT* statement of the LEASY-CATALOG utility routine.

Note the following points when changing from one pubset to another:

- The AIM files must be in a consistent state.
- The AIM files must be deleted from the old pubset.
- The main task is started with the new catalog identifier in the **ACA* statement. The AIM files are created by the main task on the new pubset.

*ADE Specify device type for AIM file

The **ADE* statement defines the device type on which an AIM file generation group is to be created. The device type can be the disk storage type of a private disk, or the tape device type. This statement is only permitted in conjunction with an **AVO* statement.

Operation	Operands
`[*]ADE[VICE]`	`=device`

=device Specifies a disk storage type or a tape device type.

All device types known to BS2000 are permitted.

The AIM file generation group is created by LEASY-MAINTASK using *OVERFLOW-OPTION=REUSE-VOLUME* (see the *CREATE-FILE-GROUP* command in the "Commands, Volume 1 - 5" manuals).

The **ADE* statement is therefore only required until the number of generations specified in the **AGE* statement is reached.

*AGE Specify number of generations of AIM files

The **AGE* statement creates an AIM file generation group or modifies the number of file generations of a cataloged AIM file generation group.

Operation	Operands
`[*]AGE[N]`	`=gen`

=gen Maximum number of simultaneously cataloged AIM file generations. An AIM file generation group is created if no such group has previously been cataloged. Otherwise a new maximum number is defined. Range of values: $1 \leq gen \leq 255$
Default value: 3

*AGF Specify number of AIM file generations which are to be released

The **AGF* statement defines the number of AIM file generations, starting with the oldest AIM file generation, which are to be released for deletion when LEASY-MAINTASK is started.

The **AGF* statement is only permitted in conjunction with the specification **AGE=255*. Otherwise it is ignored (message *LEA5318*) and no AIM file generations are released.

A maximum of 254 AIM file generations can be released. The AIM file generation which is currently in use is never released.

Operation	Operands
`[*]AGF[REE]`	`=gen`

=gen

Number of AIM file generations, starting with the oldest AIM file generation, which are to be released for deletion when LEASY-MAINTASK is started.

Range of values: 0≤ *gen* ≤ 254
Default value: 0

*AIB Define AIM buffer

The **AIB* statement defines the size of the AIM buffer in the common memory CMMAIN.

Operation	Operands
`[*]AIB[UFFER]`	`=page`

=page

Defines the size of the global AIM buffer in the common memory CMMAIN in units of 4 KB.

where:	1 ≤ *page* ≤ 999.
Default value:	2 for **LOG = A/Y*

(with **LOG=B/N* no AIM buffer is created).

i The AIM buffer must be at least twice the size of the largest element entry in the session.

The element lengths are:

for ISAM	max. record length + 24 bytes
for SAM and PAM	max. record length + 28 bytes
for DAM	max. data block length + 28 bytes

If many files are open concurrently (see the **FIL* statement), the OPEN entry in the AIM file may become very large. Element length for the AIM OPEN entry (in bytes):

length = max. no. of files * 45 bytes + 15 bytes

If an attempt is made in a user program to open an AIM-relevant file in write mode, and the maximum record length of this file exceeds that allowed for the allocated AIM buffer, the file is rejected with LEASY error code `L123` (*RC-LC* field of the *RE* area).
The best results are obtained if the buffer is twice as large as all the update entries of average transactions.

*AIO Specify performance attributes

The **AIO* statement specifies the performance attribute of the AIM file for I/O processing. This performance attribute applies to read and write operations and is assumed for the AIM generations described in the session.

Operation	Operands
[*]AIO[PERF]	={ VERY-HIGH HIGH STD USER-MAX }

=VERY-HIGH Highest performance priority. If possible, the entire AIM file is held permanently in a cache.

=HIGH If possible, the AIM file is processed via a cache.

=STD The AIM file is not processed via a cache.

New AIM files are assigned this value by default.

=USER-MAX The AIM file is assigned the highest performance attribute permitted for the user ID.

*AIS Define AIM file size

The **AIS* statement defines the number of the PAM block triggering automatic switchover to the next AIM file generation. This statement also defines the amount by which the number of the PAM block is to be increased if the switchover to the next AIM file generation cannot be executed when the specified PAM block number is reached because the oldest AIM file generation has not been released.

Operation	Operands
[*]AIS[WITCH]=	={ pamblock-number (pamblock-nummer,increment) }

=pamblock-number Number of the PAM block at which switching to the next AIM file generation is effected automatically.
4 < *pam-block-number* ≤ 16 775 000.

Default value: 524288 (this is equivalent to a file size of 1 Gbyte)

=increment

This operand controls behavior when the PAM block specified with *pamblock-number* is reached but it is not possible to switch to the next AIM file generation because the oldest AIM file generation has not been released. If this case occurs, a new value for *pamblock-number* is calculated internally:

pamblock-number(new) = pamblock-number(old) + increment

The current AIM file generation can then be used until this new value is also reached. If the oldest AIM file generation has then still not been released, a new value for *pamblock-number* is again calculated in the same way. If required, this procedure is repeated until *pamblock-number* has reached the maximum value of 16 775 000. All LEASY statements are then rejected until it is possible to switch the AIM file generation successfully.

If *increment* has the value 0, the value of *pamblock-number* is not recalculated. All LEASY statements are rejected up to the next successful switchover.

After the AIM file generation has been switched successfully, *pamblock-number* is reset again to the value specified in the **AIS* statement.
$0 \leq increment \leq 524288$

If *increment* is not specified, operations continue using the current file generation up to the maximum file size of 16775000 PAM blocks.

*APP Specify size of inquiry-and-transaction mode application table

The **APP* statement specifies the number of inquiry-and-transaction mode applications (openUTM and DCAM applications). It influences the size of the inquiry-and-transaction mode application table contained in the common memory CMMAIN.

Operation	Operands
[*]APP[]	=noita

=noita

Maximum number of inquiry-and-transaction mode applications running concurrently.

1 ≤ *noita* ≤ 255

Default value: number of tasks permitted simultaneously.

*ASP Specify memory space of new AIM file generation

The **ASP* statement specifies the primary and secondary assignments of memory space for a new AIM file generation.

Operation	Operands
[*]ASP[ACE]	={ primary (primary,secondary) TAPE[] }

=primary

A new file generation is created. The generation is assigned new *primary* PAM blocks (only for AIM files on disk).

=(primary,secondary)

The secondary assignment *secondary* specifies the number of PAM blocks which are to be reserved for the file generation as soon as more memory space is required (only for AIM files on disk).
Default value: (1920,192)

=TAPE[ ]

A new AIM file generation group is created on the next tape.

*AUT Specify automatic reconstruction

The *AUT statement defines whether AIM file generations are to be automatically reconstructed in shadow files.

Operation	Operands
[*]AUT[OTMATIC-RECONSTRUCTION]	$=\left\{\begin{array}{l}\texttt{Y[]}\\ \\ \underline{\texttt{N}}\texttt{[]}\end{array}\right\}$

=Y[ES] The AIM file generations are to be automatically reconstructed in shadow files.

=N[O] The AIM file generations are not to be automatically reconstructed in shadow files.

When **AUT=Y* is specified, AIM file generations are automatically reconstructed in only those shadow files for which automatic keeping in the AIM file was specified using the LEASY-CATALOG utility routine (*AIM=(Y,A)* or *AIM=(R,A)* entry in the **FIL* statement).

The AIM records must be written by the main task. For this reason, *LOG=(A,M)* or *LOG=(Y,M)* must be specified.

A *REN* statement must be specified for the RECONST task.

*AVO Specify VSN of volume containing AIM file

The **AVO* statement specifies the VSN of the volume on which an AIM file generation group is to be created.
The volume can be a private disk or a tape.
This statement is only permitted in conjunction with an *ADE statement.

Operation	Operands
[*]AVO[LUME]	=vsn

=vsn Specifies the volume serial number of the desired tape or private disk.

The AIM file generation group is created by LEASY-MAINTASK using *OVERFLOW-OPTION=REUSE-VOLUME* (see the *CREATE-FILE-GROUP* command in the "Commands, Volume 1 - 5" manual).

The **AVO* statement is therefore only required until the number of generations specified in the **AGE* statement is reached.

*BCA Specify public volume set for BIM file

The **BCA* statement specifies the public volume set on which a BIM file is to be generated.

Operation	Operands
[*]BCA[TID]	= bcatid

=bcatid Specifies the catalog identifier of the public volume set. *bcatid* may comprise up to 4 characters.

If the **BCA* statement is omitted, the BIM file is placed on the same public volume set as the LEASY catalog.

If a LEASY catalog which has already been generated is converted to MPVS, and if the BIM file is placed on a different public volume set to the LEASY catalog, the **BCA* statement is mandatory when LEASY-MAINTASK is started for the first time.

The **BCA* statement is not executed if *CID=N* has been specified in the *CAT* statement of the LEASY-CATALOG utility routine.

*BDE Specify disk storage type for BIM file

The **BDE* statement specifies the disk storage type of a private disk on which a BIM file is to be created. This statement is only permitted in conjunction with a **BVO* statement. It is only valid for BIM files created in addition to the BIM files already existing for the new LEASY session.

Operation	Operands
`[*]BDE[VICE]`	`=device`

=device

Specifies a disk storage unit.

All disk storage types known to BS2000/OSD are permitted.

*BIO Specify performance attributes

The **BIO* statement specifies the performance attribute of the BIM file for I/O processing. The specified performance attribute applies to read and write operations and is assumed for new and existing BIM files.

Operation	Operands
`[*]BIO[PERF]`	= VERY-HIGH HIGH STD USER-MAX

=VERY-HIGH

Highest performance priority. If possible, the entire BIM file is held permanently in a cache.

=HIGH

If possible, the BIM file is processed via a cache.

=STD

The BIM file is not processed via a cache.

=USER-MAX

The BIM file is assigned the highest performance attribute permitted for the user ID.

*BVO Specify VSN of disk containing BIM file

The **BVO* statement specifies the VSN of a private disk on which a BIM file is to be created. This statement is only permitted in conjunction with a **BDE* statement. It is only valid for BIM files created in addition to the BIM files already existing for the new LEASY session (see page 151ff).

Operation	Operands
`[*]BVO[LUME]`	`=vsn`

=vsn	Specifies the volume serial number of the desired private disk.

*CAT Access LEASY catalog

The **CAT* statement accesses the LEASY catalog for which the main task is to be started. It must always be specified.

Operation	Operands
`[*]CAT[ALOG]`	`=[:catid:]file-catalog`

catid — Catalog identifier of the public volume set on which the LEASY catalog is generated.
If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-MAINTASK was started.
catid may comprise up to 4 characters, enclosed in colons.

=file-catalog — Logical name of the LEASY catalog (≤ 8 characters) for which the main task is to be started.

*COM Insert comment

The **COM* statement is used to insert comment lines at any point within the statement sequence. The text does not influence actual execution of the utility routine.

Operation	Operands
`[*]COM[MENT]`	`[text]`

text	Any text.

*DES Specify processing of memory space on deletion

The *DES* statement specifies whether the memory space is to be overwritten when BIM files or an AIM file generation group are deleted, or whether this space is to be returned unchanged to the system.
This statement applies to a new AIM file generation group and to BIM files created in addition to the BIM files already existing for the new LEASY session.

Operation	Operands
[*]DES[TROY]	= { Y[ES] N[O] }

=Y[ES] Each time new BIM files or a new AIM file generation group are deleted the memory space is overwritten with binary zeros.

=N[O] The memory space of the files is returned unchanged to the system following deletion, unless destruction of the data has been expressly requested (see the LEASY-MASTER utility routine, *AIME* function on page 169f).

*END Terminate statement input

The *END* statement terminates the statement input. The program remains loaded, however (see page 154ff).

Operation	Operands
[*]END[]	

*FAA Always release AIM file generations

The **FAA* statement specifies whether or not the AIM file generation currently being processed should immediately be released for deletion after the next switchover to a new AIM file generation. This enables the LEASY administrator to activate the behavior of LEASY versions ≤ V6.1 for releasing AIM file generations. In this case the oldest AIM file generation is lost without any message being issued when switchover to a new AIM file generation takes place.

Operation	Operands
[*]FAA[IM-GENERATION]	={Y[] N[]}

=Y[ES] The behavior of LEASY versions ≤ V6.1 is activated, which means:

The AIM file generation currently being processed is released immediately after a successful switchover to the next AIM file generation.

In the case of automatic reconstruction of shadow files (*AUT=Y*), files for which *AIM=Y* or *AIM=R* is defined in the **FIL* statement of the LEASY-CATALOG utility routine are also permitted. In this case message *LEA2322* is issued to warn of a potential loss of information in the AIM data set.

CAUTION!

When *FAA=Y* is specified, measures must be taken to ensure that the AIM generations are saved promptly. Otherwise there is a danger that information in the AIM data set will be lost, and this makes reconstruction impossible.

=N[O] The AIM file generation currently being processed is protected against being deleted following successful switchover to the next AIM file generation and can be released only using the *AIMA* function of the LEASY-MASTER utility routine (default value).

In the case of automatic reconstruction of shadow files (*AUT=Y*), all files must be specified in the **FIL* statement of the LEASY-CATALOG utility routine using *AIM=(Y,A)*, *AIM=(R,A)* or *AIM=N*. Otherwise the LEASY maintask will be aborted and message *LEA2321* issued because there is a danger that information in the AIM data set will be lost.

*FIL Specify number of files

The **FIL* statement is used to specify the maximum number of files that can be open at any one time.

Operation	Operands
`[*]FIL[ES]`	`=files`

=files

Maximum number of files that can be open at any one time.

1 ≤ *files* ≤ 999

Default value: 30.

*KEY Specify key length

The **KEY* statement specifies the maximum key length of all ISAM, DAM and PAM files in the LEASY catalog for which lock elements are to be enforced during the current session.

Operation	Operands
`[*]KEY[LEN]`	`=keylen`

=keylen

Maximum key length of all ISAM, DAM and PAM files in the catalog for which lock log elements are to be enforced during the current session.

where: 1 ≤ *keylen* ≤ 255

Default value: maximum key length of all ISAM, DAM and PAM files in the LEASY catalog.

If **KEY* is used to specify a value greater than the default value calculated by the LEASY-MAINTASK utility routine, then the default value used for calculating the length of a lock log element is used.

If **KEY* is used to specify a value less than the calculated default value, the lock log elements are created with the length specified. In other words, the lower value is adopted in each case. This function may be used, for example, when the LEASY catalog contains a file which is not accessed in the subsequent LEASY session and which has an excessive key length. A low value prevents this file from inflating the lock log elements.

*LOG Determine save method

The **LOG* statement determines the save method to be used in this LEASY session.

Operation	Operands
[*]LOG[]	B[] M[] A[][,][,K[]] [,P[]] R[] = M[] Y[][,] [,P[]] R[] N[][,K[]]

=B[IM] In this session only BIM files are to be used.

=A[IM] In this session only AIM files are to be used.

=A,M[AINTASK] The AIM buffer is written by means of the main LEASY task.

=A,R[UNTIME] The AIM buffer is written by means of the LEASY runtime system.

=K[EEP-BIM-FILES] BIM files are retained. If this operand is not specified, existing BIM files are deleted.

=Y[ES] In this session both BIM and AIM files are to be used.

=Y,M[AINTASK] The AIM buffer is written by means of the main LEASY task.

=Y,R[UNTIME] The AIM buffer is written by means of the LEASY runtime system.

=N[O] No save files are to be used in this session.

=P[ROTOCOL] The read operations, the *LOCK* and *UNLK* operations and the *CINF* information function are also logged in the AIM file.

If **LOG=B* or **LOG=Y* is specified, only those files are stored in before-image save files for which this has been specified in the LEASY-CATALOG utility routine by means of a **FIL* statement and the *BIM=Y* operand.

If **LOG=A* or **LOG=Y* is specified, these specifications likewise only apply to those files for which after-image saving has been specified in the LEASY-CATALOG utility routine by means of *AIM=YES*.

If the AIM file is on tape, the AIM buffer is automatically written with the aid of the main LEASY task even if the *M* operand has not been specified.

If **LOG=A,M* or **LOG=Y,M* is specified, the AIM file is not opened with *SHARED-UPDATE=YES*. Concurrent processing of the AIM file with LEASY-RECONST (**MOD ,UPD=N*) is thus not possible.

If *P[ROTOCOL]* is specified, the read operations, the *LOCK* and *UNLK* operations and the *CINF* information function can also be logged in the AIM file. This provides a clearer overview of all the actions of a LEASY application, which can serve as a basis for optimization.

*MEM Define size of common memory

The **MEM* statement defines the total size of the common memory CMMAIN in segments (units of 64 Kb). This specification also influences the maximum possible number of lock log elements.

Operation	Operands
`[*]MEM[ORY]`	`=mem`

=mem
The total size of the common memory should also take into account the fact that this memory contains the AIM buffer (**AIB* statement).

where: 1 ≤ *mem* ≤ 999

Default value: 1.

*MFB Specify the multiplication factor for defining the maximum number of buckets to be held free

The **MFB* statement specifies the multiplication factor to be used to define the number of buckets that are held free against lock element list requests from batch or TIAM tasks (see also section “Bucket management” on page 123).

Operation	Operands
`[*]MFB[]`	`=mfact`

=mfact

Multiplication factor that is multiplied with the *maxtrans* value of the **TRA* statement to define the maximum number of buckets that are to be held free against lock element list requests from batch/TIAM tasks. These buckets then remain reserved for transaction elements.

where: 1 <= *mfact* <= 8

Default value:1

*MTT Define wait time for main task termination

The **MTT* statement defines the number of minutes waited for transactions that have not been completed with a main task AIM write when the task has been terminated via the *SHUT* or *CLOS* function of the LEASY-MASTER utility routine.

Operation	Operands
`[*]MTT[IME]`	`=wtime`

=wtime maximum wait time in minutes for transactions that have not been completed with a main task AIM write when the task has been terminated via the *SHUT* or *CLOS* function of the LEASY-MASTER utility routine.

where: 5 <= *wtime* <= 120

Default value:10

If the wait time is set too low, the main task terminates abnormally, i.e. transactions remain open. However, these are rolled back via the STXIT routine of LEASY. Due to a CLTR element to be set in the AIM file that must be written by the no longer existent main task, an LS65 return code is output (Maintask ended with errors) with DUMP. The lower limit of the adjustable wait time is set to 5 minutes to avoid this condition as far as possible.

Any further requests to the main task functions after it has been terminated are also rejected with the LS65 return code, without any wait times being generated.

It is advisable to terminate the main task with the *CLOS* function of LEASY-MASTER. The LEASY runtime system then does not take on any new transactions and terminates the LEASY session after setting a consistency point.

*MUS Define size of memory units (buckets)

The **MUS* statement defines the size of the memory units (buckets) for the unit memory in the common memory.

Operation	Operands
`[*]MUS[]`	`=unitsize`

=unitsize

The size in bytes of a memory unit (bucket) for the unit memory (bucket pool) of the CMMAIN.

where: 256 ≤ *unitsize* ≤ 32 768

Default value: 1024.

The buckets are used primarily to store information concerning the file identifiers of the current transaction. They are also used to store lock elements if the contiguous lock element memory provided for this purpose overflows.

The size of the buckets specified should be such that, if possible, the data of all file identifiers for each transaction has room in one bucket. The data belonging to the file identifiers is divided into file-specific elements and sequence-identifier-specific elements.

The file-specific elements are 44 bytes long.

The sequence-identifier-specific elements are of variable length:

- for SAM files: 24 bytes
- for ISAM, DAM and PAM files: 12 + (3*length of primary key)
- for files with secondary keys: 12 + (3*key length of SI file).

A sequence-identifier-specific element is created for each file identifier opened in the transaction.
The current sequence-identifier-specific elements are duplicated for the LEASY operation *CLTR* with *OPE2=T* (twice as much space required).

i The assignment of buckets to lock elements or transaction information is retained throughout a given LEASY session. The total number of buckets required is thus the sum of

- the maximum number of buckets needed for lock elements plus
- the maximum number of buckets needed for transaction information.

Under "unfavorable" conditions (e.g. when numerous records are locked in a transaction on session startup and then a large number of transactions locking few records are opened) the CMMAIN common memory size must therefore be relatively large. In certain cases, it may also be necessary to use the **MFB* statement to reserve a sufficient area for transaction information.

*PAS Pass passwords for RECONST task

The **PAS* statement supplies the RECONST task with passwords for files protected by means of a DMS password (e.g. shadow files). More than one **PAS* statement can be entered. **PAS* statements may only be specified after the associated **REN* statement.

Operation	Operands
`[*]PAS[WORD]`	`=password`

=password

Defines a password up to 4 bytes in length, which can be represented as follows:

C’c..c’ — *c* stands for 1 to 4 alphanumeric characters and special characters.

X’h..h’ — *h* represents 1 to 8 hexadecimal numbers.

d — *d* is a decimal number of up to 8 digits, the value of which is converted to binary form.

The password with the value *X’00000000’* is ignored.

*REN Define ENTER-JOB command for RECONST task

The **REN* statement defines an *ENTER-JOB* command for starting the RECONST task with automatic reconstruction of shadow files.

Operation	Operands
`[*]REN[TER]`	`=enter-command`

=enter-command *ENTER-JOB* command with operands. See the *ENTER-JOB* command in the "Commands, Volume 1 - 5" manual for permissible specifications.

The file specified under *enter-command* need not be entered in the DMS catalog, since it is created by LEASY-MAINTASK each time a LEASY session is started and it does not already exist.

The operands in *enter-command* are not checked by LEASY-MAINTASK, rather they are passed on unchanged with the *CMD* macro.

The name of the ENTER file must be specified as a positional operand, i.e. in position 1 and without *FROM-FILE*. Library elements must not be specified as ENTER files.

ENTER procedures are not permitted.

*STA Determine cold/warm start

The **STA* statement determines whether the session can only be cold-started or whether a warm start is also possible (see page 151ff).

Operation	Operands

`[*]STA[RT]`	`={C`
	`W[,R]}`

=C[OLD] Only a cold start is possible.

=W[ARM] A warm start is also possible.

If *STA=W* is specified, a cold start is also performed if the previous session was correctly terminated.

=W,R[ESET] In the case of a warm start suspended transactions are reset independently of openUTM.

*TIM Define maximum waiting time for canceling lock

The **TIM* statement defines the maximum waiting time in seconds for canceling a lock.

Operation	Operands
`[*]TIM[E]`	`=time`

=time

where: 0 ≤ *time* ≤ 999

Default value: 10.

*TRA Define maximum number of transactions

The **TRA* statement defines the maximum number of transactions which can be run concurrently.

Operation	Operands
`[*]TRA[NS]`	`=maxtrans`

=maxtrans

Maximum number of transactions which can be run concurrently.

where: 1 ≤ *maxtrans* ≤ 255

(default value=4)

i No lock element lists are kept if **TRA=1* is specified. This can lead to considerable improvements in performance.

*TSK Specify size of task table

The **TSK* statement influences the size of the task table contained in the common memory CMMAIN.

Operation	Operands
`[*]TSK[]`	`=anotask`

=notask

Maximum number of parallel tasks accessing this CMMAIN (timesharing, batch, inquiry and transaction processing tasks).

where: 1 ≤ *notask* ≤ 255

Default value: Number of permitted parallel transactions

*USE Define state of common memory

The **USE* statement determines the state of the common memory CMMAIN following initialization.

Operation	Operands
`[*]USE[]`	`=` N C R

=N[ORMAL] The common memory CMMAIN is released for the LEASY runtime system.

=C[ONSISTENCE] Following initialization, the common memory CMMAIN is released for the purpose of ensuring consistency. This closes any transactions that are still open.
No new AIM generation or LEASY session is initiated; instead, the main task is terminated after it has rolled back all transactions that were not closed.

=R[ECONSTRUCTION]
The common memory CMMAIN is made available for an after-image reconstruction run. In this state, access attempts by the runtime system are rejected. Only after a successful reconstruction run does LEASY-RECONST release the common memory for the LEASY runtime system.

Reconstruction of shadow files with LEASY-RECONST is possible not only if CMMAIN is available for an after-image reconstruction run (**USE=R*), but also if the CMMAIN is released for the LEASY runtime system (**USE=N*).

7.3 Main task activities

The main task performs the following tasks from the time it is started up until it terminates:

- Reads and validates the input.
- Requests common memory.
- Opens the LEASY catalog in INOUT mode and stores its records in the common memory CMMAIN.
- Initializes tables in the common memory CMMAIN.
- Creates an AIM file generation group if the appropriate statements are entered.
- Opens an AIM file and create an AIM buffer in CMMAIN.
- Writes an MTSK element to the AIM file.
- Executes a cold or warm start (**STA*) as required; issues *VERIFY* macros for all files opened during the warm start and for which the *EXLST-OPENC* exit is activated. The macros are issued with *REPAIR=NO* for ISAM files, and with *REPAIR=YES* for all other files.
- Increments the session number by 1 and writes the session element to the AIM file.
- Closes all files and reopens the LEASY catalog with *INPUT/SHARED-UPDATE*.
- If no error has occurred, the common memory CMMAIN is released, either for linking the LEASY application programs or for after-image reconstruction, i.e. initialization is terminated and the main task enters a wait loop.
- If an error has occurred, the LEASY-MAINTASK utility routine terminates with *TERM MODE=ABNORMAL,UNIT=STEP* in order to be ready to respond in procedures if the warm or cold start is unsuccessful.
- If **USE=C*, the main task is executed for the purpose of restoring consistency by closing transactions that are still open after errors have occurred and is then terminated.

7.4 Cold start, warm start and ensuring consistency

Cold start

A cold start is only possible if no open transactions are encountered, i.e if all BIM files have been defined as "empty".

If the statement **LOG=B* or **LOG=Y* has been specified in the LEASY-MAINTASK utility routine, a check is made to see if there are more or fewer BIM files than the number of parallel transactions permitted. If there are more than necessary, the surplus is deleted in descending order starting with the highest number. If there are not enough BIM files, those additionally required are created (**BVO* and **BDE* are, if specified, effective for these only).

They are given the name

```
:catid:$userid.file-catalog.BIM#.nnn
```

where *nnn* is a 3-digit decimal number assigned in ascending order starting with the first number so far not available.

If **LOG=N* or **LOG=A* has been specified, no new BIM files are initialized; the session is run without the rollback facility.

Warm start

In the case of a warm start all related BIM files from the last session are analyzed, i.e. all files bearing the name

```
:catid:$userid.file-catalog.BIM#.nnn
```

(where *nnn* = 001,002,003... up to the first gap).

If a BIM file is not logically empty, a check is performed to determine whether the transaction is open or whether it is suspended. Open transactions are rolled back and the file declared empty. The entry in the DMS catalog is preserved. Suspended transactions are entered in a common memory table. They are not terminated or reset until the relevant openUTM applications have been warm-started and linked to the common memory of openUTM.

The main task enters a wait loop, where it remains until all suspended transactions have been completed. During this time only those openUTM applications for which there are suspended transactions can be linked to the common memory, and only internal calls can be used to process them. A waiting time of up to 5 minutes applies to new transactions (*INIT*), as with the *CATD* operation (accessing the LEASY catalog); during this waiting time a check is performed once every second to determine whether or not the initialization phase has already been terminated.

If *STA=W,R* is set, suspended transactions are processed in the same manner as open transactions, i.e. they are rolled back by main task.

If all the old BIM files have been processed successfully, the same procedure is followed as for a cold start.

i Users must never delete BIM files themselves, otherwise all higher BIM files will be ignored in a subsequent warm start.

If AIM saving was activated for the last session, the new session must also be executed with AIM saving if a warm start is performed. This is necessary because entries are again written into the AIM file as a result of rolling back open transactions.

The user can set up BIM files before the LEASY-MAINTASK utility routine is called. This enables the BIM files to be distributed among several disks and on cylinders where accessing is easy. This is useful for time-critical applications.

The names of these BIM files must be formed in accordance with the name structure used by LEASY for generating BIM file names.

Ensuring consistency

When it terminates a session, LEASY normally leaves the applications in a consistent state. All open transactions are terminated or rolled back. Even when the program is terminated abnormally, LEASY has STXIT routines that serve to reset any open transactions.

In the following exceptional cases, however, a LEASY session may terminate without leaving the associated files in a consistent state:

- Files could not be rolled back (e.g. access was not possible).
- When the program was terminated abnormally, no STXIT routine was addressed (e.g. with *FORCE-JOB-CANCEL*).
- A LEASY call could no longer be issued due to loops.

In earlier LEASY versions, a consistent data set could be generated in such cases only by performing a warm start to initiate a new LEASY session.

The *function *USE=C,* which resets open transactions, is provided in order to simplify operation for the user.

A main task started with **USE=C* differs from a warm start (*USE=N, STA=W)* in the following respects:

– AIM writing continues in any case in the current AIM generation; a new generation is not initiated.

– No LEASY session is initiated in which applications can run.

– The main task is terminated after it has rolled back all open transactions.
If complete data consistency could not be achieved, the main task is terminated with *TERM MODE=ABNORMAL, UNIT=STEP.*

If there are any suspended openUTM transactions, the main task switches to a wait loop until all openUTM applications have terminated or rolled back their transactions; only then is the main task terminated. New transactions are not permitted in this state.

If the consistency of only the LEASY catalog is to be ensured without having to start up openUTM applications, *STA=W,R* must be specified in addition to the *operand *USE=C*. Suspended transactions are then rolled back by the main task and a openUTM status file is created by openUTM.

7.5 Opening and terminating a session

Opening a session

The main task initializes the common memory CMMAIN, which it maintains for the entire session. If the main task is started from a terminal, the terminal (or line) is thus blocked. It is therefore advisable to run the main task as a batch task and to start it with an *ENTER* procedure.

Example

```
/SET-LOGON-PAR JOB-NAME=SE$LOHN
/START-LEASY-MAINTASK
*CAT=SE$LOHN
*FILES=3
*KEYLEN=20
*TIME=20
*TRANS=4
*BVOLUME=PRIVAT
*BDEVICE=D5882
*END
/EXIT-JOB
```

Terminating a session

A session is terminated by closing the main task with the aid of the LEASY-MASTER operation *TERM*, *CLOS* or *SHUT*. The common memory CMMAIN is then only retained for as long as user programs are still linked to it. Only when all programs belonging to this user group are terminated can a new system initialization for the same user group be effected by means of a new main task.

7.6 Example

Tracer listing 1

Existing common memory CMMAIN

```
/START-LEASY-MAINTASK ------------------------------------------------------------ (1)
%  BLS0523 ELEMENT 'MAINTASK', VERSION '06.2A' FROM LIBRARY
   ':20SL:$TSOS.SYSPRG.LEASY.062' IN PROCESS
%  BLS0524 LLM 'LEASY-MAINTASK', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0301 LEASY MAINTASK VERSION V6.2A STARTED
*CAT=LCAT ------------------------------------------------------------------------ (2)
*END ----------------------------------------------------------------------------- (3)
%  LEA2301 COMMON MEMORY ALREADY EXISTS  ------------------------------------------ (4)
%  LEA5305 *LEASY MAINTASK   INITIALIZATION FINISHED WITH ERROR*
```

Explanation

(1) The LEASY-MAINTASK utility routine is called.

(2) The main task is to be initialized for the LEASY catalog *LCAT*:

(3) Statement input is terminated and the main task is started (all other statements are given default values).

(4) Messages from the LEASY-MAINTASK utility routine: The common memory CMMAIN already exists; the program is aborted.

Tracer listing 2

AIM reconstruction run

```
/START-LEASY-MAINTASK────────────────────────────────────────────────────────────────────────  (1)
%  BLS0523 ELEMENT 'MAINTASK', VERSION '06.2A' FROM LIBRARY
   ':2OSL:$TSOS.SYSPRG.LEASY.062' IN PROCESS
%  BLS0524 LLM 'LEASY-MAINTASK', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0301 LEASY MAINTASK VERSION V6.2A STARTED
*CAT=LCAT ───────────────────────────────────────────────────────────────────────────────────  (2)
*USE=R ──────────────────────────────────────────────────────────────────────────────────────  (3)
*LOG=Y ──────────────────────────────────────────────────────────────────────────────────────  (4)
*END ────────────────────────────────────────────────────────────────────────────────────────  (5)
%  LEA5306 *LEASY MAINTASK :01KC:$LEASY06A.LCAT READY FOR RECONSTRUCTION* ───────────────────  (6)
%  LEA5303 WARM/COLD START SUCCESSFUL ───────────────────────────────────────────────────────  (7)
%  LEA5307 NEW LEASY SESSION CREATED: SESSION NUMBER = 00008,       } ────────────────────────  (8)
%          DATE = 2006-05-20, TIME = 10:27:49-S.                    }
%  LEA5304 *LEASY MAINTASK :01KC:$LEASY06A.LCAT INITIALIZATION COMPLETED ────────────────────  (9)
%  LEA0310 NORMAL TERMINATION OF LEASY MAINTASK BECAUSE OF CLOSE DOWN FUNCTION ──────────── (10)
```

Explanation

(1) The LEASY-MAINTASK utility routine is called.

(2) The main task is to be started for the LEASY catalog *LCAT*.

(3) The common memory CMMAIN is made available for a reconstruction run.

(4) The session is to be run with both AIM and BIM saving.

(5) Statement input is terminated and the main task is started.

(6) The common memory CMMAIN is ready for reconstruction.

(7) The cold start (default value) was completed without error after reconstruction.

(8) A new LEASY session bearing the session number 00008 has been created.

(9) The main task is ready for the LEASY runtime system.

(10) The LEASY session is terminated by the LEASY-MASTER utility routine.

Tracer listing 3

Opening and terminating a session

```
/START-LEASY-MAINTASK------------------------------------------------------------ (1)
%  BLS0523 ELEMENT 'MAINTASK', VERSION '06.2A' FROM LIBRARY
   ':2OSL:$TSOS.SYSPRG.LEASY.062' IN PROCESS
%  BLS0524 LLM 'LEASY-MAINTASK', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0301 LEASY MAINTASK VERSION V6.2A STARTED
*CAT=LCAT ----------------------------------------------------------------------- (2)
*FILES=2 ------------------------------------------------------------------------ (3)
*LOG=Y -------------------------------------------------------------------------- (4)
*TRANS=3 ------------------------------------------------------------------------ (5)
*TIM=40 ------------------------------------------------------------------------- (6)
*END ---------------------------------------------------------------------------- (7)
%  LEA5303 WARM/COLD START SUCCESSFUL --------------------------------------------- (8)
%  LEA5307 NEW LEASY SESSION CREATED: SESSION NUMBER = 00010,      } --------------- (9)
           DATE = 2006-05-20, TIME = 11:12:09-S.                   }
%  LEA5304 *LEASY MAINTASK :01KC:$LEASY06A.LCAT INITIALIZATION COMPLETED --------- (10)
%  LEA5003 START OF AIM FILE GENERATION SWITCHING ON 2006-05-20 AT 11:47:38-S ---- (11)
PASSWORD SSSSSSSSSSS
PASSWORD SSSSSSSSSSS
PASSWORD SSSSSSSSSSS
PASSWORD SSSSSSSSSSS
%  LEA5004 AIM FILE GENERATION SWITCHING SUCCESSFUL ------------------------------ (12)
%  LEA0310 NORMAL TERMINATION OF LEASY MAINTASK BECAUSE OF CLOSE DOWN FUNCTION --- (13)
```

Explanation

(1) The LEASY-MAINTASK utility routine is called.

(2) The main task is to be started for the LEASY catalog *LCAT*.

(3) Two files may be opened simultaneously.

(4) The session is to be run with both AIM and BIM saving.

(5) Three transactions may be executed concurrently.

(6) Maximum waiting time of 40 seconds for the removal of a lock.

(7) Statement input is terminated and the main task is started.

(8) The cold start was completed without errors.

(9) A new LEASY session bearing the session number 00010 has been created.

(10) The main task is ready for the LEASY runtime system.

(11) The LEASY-MASTER utility routine is used to switch over to the next AIM file generation.

(12) The switchover procedure to the next AIM file generation has been terminated correctly.

(13) The LEASY session is terminated by the LEASY-MASTER utility routine.

8 LEASY-MASTER

The LEASY-MASTER utility routine controls and monitors the common memory CMMAIN.

LEASY-MASTER enables the user to

- terminate or suspend the main task, the LEASY session or the I/O task
- lock and release transactions or files
- roll back transactions
- switch to the next AIM file generation
- erase AIM file generation groups or AIM file generations
- release AIM file generations
- replace original files by shadow files
- set or reset READ-ONLY mode
- output information concerning the CMMAIN, the LEASY session, tasks and transactions.

The utility routine provides prompting. It can however also be used in batch mode. Input/output takes place in interactive mode via SYSDTA/SYSOUT in the form of screen masks. In batch mode, input is via SYSDTA and output via SYSLST. The SYSLST output can be activated additionally for interactive mode. The first line of each mask contains the mask number together with the function of the mask.

The utility routine responds with:

```
LEA0501 LEASY MASTER PROGRAM VERSION V6.2A STARTED
```

The dialog with the user is then started.

ℹ Input record chaining is not permitted.

8.1 Selecting a common memory CMMAIN

The mask appears with:

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 001: MAINTASK SELECTION
```

Prompt

```
PLEASE ENTER NAME OF LEASY DIRECTORY
(*END = END OF PROGRAM)
```

Possible responses

`[:catid:][$userid.]file-catalog`

Logical name of the LEASY catalog

`*END` Termination of the program run.

where:

catid Specifies the catalog identifier of the public volume set which contains the LEASY catalog. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set which contains the user ID under which LEASY-MASTER was started.
catid may comprise up to 4 characters, enclosed in colons.

userid Specifies the user ID under which the LEASY catalog is entered in the public volume set with the catalog identifier *catid*. If *userid* is omitted, LEASY uses the user ID under which LEASY-MASTER was started.

file-catalog Specifies the name of the LEASY catalog.

Any number of common memories CMMAIN can be processed successively in one run by specifying different catalog names one after the other.

The relevant CMMAIN is selected by an ENAMP macro immediately after a catalog name is entered. If the user terminates the dialog with a particular CMMAIN and returns to the main task menu, the connection to the CMMAIN is broken by a *DISMP* macro. If the main task was terminated in the preceding dialog, however, the same common memory can only be reselected for as long as the common memory exists in the system.

If the LEASY catalog possesses a LEASY password, this password is requested for execution of the LEASY-MASTER utility routine:

```
PLEASE ENTER PASSWORD
```

The input

C’c...c’ where _c_ represents 1 to 4 printable characters

X’h...h’ where _h_ represents 1 to 8 hexadecimal characters

is the same as for the _*CAT_ statement in the LEASY-CATALOG utility routine.

The LEASY-MASTER utility routine can be employed simultaneously by several users for the same common memory. However, this means that the first user is privileged: only he/she may issue functions that trigger responses in the main LEASY task or in the LEASY runtime system. Other users may use only display functions.

The utility routine then responds with the general information mask.

8.2 Functions of the LEASY-MASTER utility routine

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.

The functions are described below in alphabetical order.

AIMA Administer AIM files

The mask appears with the message:

```
LEASY MASTER PROGRAM VERSION V6.2A      SCREEN 036: AIM-FILE-ADMINISTRATION
```

and displays the name of the selected common memory CMMAIN.

A display then indicates whether automatic reconstruction of shadow files is being used.

> **i** When automatic reconstruction of shadow files is **not** being used, the mask contains no further information and a new function can be switched to immediately only by specifying a function code.

When automatic reconstruction of shadow files **is** being used, the following information lines also appear:

- Number of AIM file generations
- AIM file generations released for deletion
- AIM file generations protected against deletion
- AIM file generations currently used
- Information on the number of the PAM page at which the AIM file generation is to be switched over:
  - Current value
  - Normal value (from the MAINTASK parameter *AIS*)
  - Increment by which this value will be increased if the switchover cannot be performed because no AIM file generation is free
- Number of switchovers which were unsuccessful since the last successful switchover and
  reason for the last unsuccessful switchover (*AIMC*, *AIMI*, *AIMW* or *IMPL* = implicit switchover because the file size of the AIM file generation specified in the MAINTASK parameter AIS has been reached).

A new function code for switching to a new function can then be entered or one of the following subfunctions can be called:

F	Release AIM file generations (FREE)
P	Protect AIM file generations (PROTECT)

Subfunction F (release AIM file generations for deletion)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A      SCREEN 037: UNPROTECT AIM-FILES
```

Prompt

```
PLEASE  ENTER  THE  QUANTITY  OF  AIM-FILE-GEN.  TO  FREE  (BLANK=NO  AIM-FILE  TO
FREE):
```

Possible response

Number of consecutive AIM file generations which are to be released (starting with the oldest AID file generation which is protected against deletion).

CAUTION!

The LEASY administrator is responsible for ensuring that the AIM file generations which they have released are no longer required. They should either be saved or reconstructed.

The next function required can then be switched to by specifying a function code.

Subfunction P (protect AIM file generations against deletion)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A      SCREEN 038: PROTECT AIM-FILES
```

Prompt

```
PLEASE ENTER THE QUANTITY OF AIM-FILE-GEN. TO PROTECT (BLANK=NO PROTECTION):
```

Possible response

Number of consecutive AIM file generations which are to be protected against deletion (starting with the file generation released most recently).

The next function required can then be switched to by specifying a function code.

AIMC Switch AIM file generation

The *AIMC* function switches the AIM file generations after all the current transactions have been terminated. This must have been induced by a *QUIE* function. The function is not accepted until all the transactions have been terminated.

The general information mask then appears.

The following is necessary for switching:

- the main LEASY task must still be active
- the session generated must operate with AIM saving
- no switching must currently be in progress
- At least one free AIM file generation still exists, i.e. the number of generations specified in the **AGE* statement of the LEASY-MAINTASK utility routine has not yet been reached or (at least) one AIM file generation has been released for deletion.

The actual switching process (writing a key record to the old AIM file; preparing a new AIM file) is undertaken by the main task during an interruption of its wait state.

LEASY-MASTER waits for the switching routine to be completed and reports either the successful execution or the errors that occurred in main task during switching.

LEASY provides automatic switching to the next AIM file generation. This switching process is initiated by the file size of the AIM generation and is carried out independently by the main task. It produces the same effect as the LEASY-MASTER function *AIMI* and requires no program run of the LEASY-MASTER utility routine.

AIME Erase AIM files

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 021: ERASE AIM FILE
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER AIM FILE GENERATION NUMBER YOU WANT TO ERASE
(BLANK = NO ERASE)
```

Possible responses

```
⎧W[HOLE]              ⎫   ⎧DESTROY⎫
⎪                     ⎪   ⎪DATA   ⎪
⎨⎧*⎫      ⎧A[FTER]  ⎫ ⎬[,⎨       ⎬][,VOLUME=vsn]
⎪⎨+⎬gen,⎨          ⎬ ⎪   ⎪SPACE  ⎪
⎩⎩-⎭      ⎩B[EFORE]]⎭ ⎭   ⎩CATALOG⎭
```

where

W[HOLE] The file generation group is erased in its entirety.

Any *AFTER* or *BEFORE* specification has no effect.

* The *gen* specification is absolute.

+ or - The *gen* specification is relative to the base.

gen Specifies a generation number to which the *, +, -, *AFTER* and *BEFORE* specifications refer. The generation number comprises four digits; leading zeros must be included.

A[FTER] Defines the relative or absolute generation number after which files are to be erased. The specified generation must exist; it is not erased.

B[EFORE] Defines the relative or absolute generation number up to which files are to be erased. The specified generation must exist; it is not erased.

DESTROY Specifies that the memory space of the specified generation(s) is to be released and also the data overwritten with binary zeros.

DATA Specifies that the data of the specified generation(s) is to be logically erased. The individual file remains cataloged and the memory space allocated.

SPACE
The memory space allocated for the specified generations(s) is released. The individual file remains cataloged, but is processed as if it were logically erased. This operand is not permitted for private disks.

CATALOG
The catalog entry for the specified generation is erased, whilst the catalog entry for the group is updated.
The *CATALOG* operand is only permitted for file generations on private volumes.

VOLUME=vsn
This operand is only permitted for private disks in conjunction with the *CATALOG* operand.
It erases all catalog entries for file generations occupying memory space on the specified volume. Gaps may result in this file generation group.

If a mistake has been made and the generation is not to be erased, at least one blank should be entered.

The operands described above are supplied to an internally issued *ERASE* macro, but are subjected only to a syntax check by the LEASY-MASTER utility routine. Errored specifications can therefore only be detected by DMS (see the "Introductory Guide to DMS" manual).

Only file generations which have not been opened by LEASY can be erased.

If all generations are to be deleted (response: *WHOLE*), the main task must be started with *LOG=A, R* or *LOG=Y, R*. No function controlling LEASY-MAINTASK may be active (*TERM/CLOS/SHUT*) and no user tasks may be using the AIM file generation.

Once all generations have been deleted, logging for the current session is deactivated.

If erasure is successful, it is then possible to switch immediately to the next function desired by specifying a function code.

AIMI Switch AIM file generation

The *AIMI* function is used for switching the AIM file generations directly. In contrast to the *AIMC* function, switching takes place irrespective of whether or not all the transactions have been terminated.

The general information mask then appears.

The following is necessary for switching:

- The main LEASY task must still be active
- The session generated must operate with AIM saving
- No switching must currently be in progress
- At least one free AIM file generation still exists, i.e. the number of generations specified in the **AGE* statement of the LEASY-MAINTASK utility routine has not yet been reached or (at least) one AIM file generation has been released for deletion.

The actual switching process (writing a key record to the old AIM file; preparing a new AIM file) is undertaken by the main task during an interruption of its wait state.

LEASY-MASTER waits for the switching routine to be completed and reports either the successful execution or the errors that occurred in LEASY-MAINTASK during switching.

LEASY provides automatic switching to the next AIM file generation. This switching process is initiated by the file size of the AIM generation and is carried out independently by the main task. It produces the same effect as the LEASY-MASTER function *AIMI* and requires no program run of the LEASY-MASTER utility routine.

AIMS Show state of the AIM generation

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 012: STATE OF AIM FILES
```

Afterwards the following information lines appear:

- display of the name of the selected CMMAIN
- indication as to whether the AIM file generations are to be automatically reconstructed in shadow files. If so, the following continuation lines appear:
- number of AIM file generations
- free AIM file generations
- AIM file generation to which data is written
- AIM file generation which is ready for reconstruction
- AIM file generation waiting for the end of the switchover procedure
- AIM file generation which is being reconstructed

Afterwards a new function code can be entered.

AIMW Switch AIM file generation

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 032: AIM−SWITCH WITH WAIT
```

The *AIMW* function switches the AIM file generation after it has waited for the end of the transaction. This must have been induced by a *QUIE* function.

The general information mask then appears.

The following is necessary for switching:

- The main LEASY task must still be active
- The session generated must operate with AIM saving
- No switching must currently be in progress
- At least one free AIM file generation still exists, i.e. the number of generations specified in the **AGE* statement of the LEASY-MAINTASK utility routine has not yet been reached or (at least) one AIM file generation has been released for deletion.

The number of checks for the end of the transaction must be entered in the AIMW mask (range of values: 0 to 99). AIMW is terminated if "0" is entered. A "1" has the same effect as *AIMC*. If any other value is entered, the user will be prompted to enter the time interval (in seconds) between the individual queries. The default value is 5 queries at 5-second intervals.

The actual switching process (writing a key record to the old AIM file; preparing a new AIM file) is undertaken by the main task during an interruption of its wait state.

LEASY-MASTER waits for the switching routine to be completed and reports either the successful execution or the errors that occurred in the main task during switching.

LEASY provides automatic switching to the next AIM file generation. This switching process is initiated by the file size of the AIM generation and is carried out independently by the main task. It produces the same effect as the LEASY-MASTER function *AIMI* and requires no program run of the LEASY-MASTER utility routine.

CLOS Terminate LEASY-MAINTASK

The *CLOS* function terminates the main task and thus also the LEASY session. New transactions are no longer permitted; current transactions are not impeded.

Effect

Operations permitted at the program interface:

- *CLTR, CLTR* with *OPE1=R*
- *CLFL, CINF*
- all record operations
- *CATD* for no CMMAIN or for a different one.

Prohibited operations (return code):

- *CATD* for this CMMAIN
- *OPFL, OPTR*.

Operations changed by the program interface (return code):

- *CLTR* with *OPE2=T* becomes *CLTR*
- *CLTR* with *OPE1=R* and *OPE2=T* becomes *CLTR* with *OPE1=R*.

The general information mask then appears.

CONT General cancelation of a transaction lock

A wait state set on transactions by the functions *HOLD* and *QUIE* (general) or *LOCT* (specific) is canceled again.

The general information mask then appears.

CYCI Cyclic display of table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 004: DISPLAY TABLE CYCLIC
```

and the selected common memory CMMAIN is displayed.

The number of display cycles must be entered first.

Prompt

```
PLEASE ENTER NUMBER OF CYCLES (BLANK = >n<; MAX = 9999)
```

Possible responses

n

The desired number of display repetitions must be entered in numeric form.
$0 \leq n \leq 9999$.

No table is displayed if zero is entered, only the general function selection.

_

The last entered number n that appears in the request is always used.
Initial setting: $n=10$.

The display time for each screen page is then requested:

Prompt

```
PLEASE ENTER DISPLAY TIME (BLANK = >t< SECONDS; MIN = 0, MAX = 9999):
```

Possible responses

t

The desired display time for each information mask must be entered in numeric form.
$0 \leq t \leq 9999$.

_

The last entered display time t that appears in the request is always used.
Initial setting: $t=10$ seconds.

The display time corresponds roughly to the interval for retrieving information from the common memory CMMAIN. This makes it possible to trace modifications in the CMMAIN over an extended period of time without the need for external inputs.

The utility routine then requests the function code of the table to be displayed cyclically:

Prompt

```
PLEASE ENTER FUNCTION CODE OF TABLE YOU WANT TO DISPLAY CYCLICALLY
(BLANK = NO CYCLIC DISPLAY)
```

Possible responses

GENT The general information mask is displayed repeatedly.

GENC The general counters of a LEASY session are displayed repeatedly.

TSKT The task table is displayed repeatedly.

TRAT The transaction table is displayed repeatedly.

OPFT The table of the master and model files opened by transactions is displayed repeatedly.

THAT The inquiry-and-transaction mode application table is displayed repeatedly.

ℹ It is possible to specify *UTMA* instead of *THAT* for reasons of compatibility.

IOGT The general I/O task table is displayed repeatedly.

IOQA The I/O task queuing availability is displayed repeatedly.

IOUT The I/O task user table is displayed repeatedly.

IOTT The I/O task task table is displayed repeatedly.

SHLE The table of lock elements is displayed repeatedly.

_ No table is displayed; the function selection mask appears on the screen.

If a table covers several screen pages, all screen pages of the table are displayed. The display time also applies with reference to two screen pages of the same table.

After a correct function selection has been entered, the display cycle is started immediately. As soon as the last display of the last screen page of the table is reached, a new function code may be entered.

DPRC Dump on return code

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 029: DUMP ON RETURNCODE
```

The *DPRC* function is used to create, display and delete return codes which produce a memory dump when they appear. A maximum of 70 return codes can be specified.

When the *DPRC* function is called the return codes already set are displayed in a mask.

Subfunction ADD

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 030: ADD RETURNCODE
```

The *ADD* subfunction (enter: *A*) is used to add further return codes. The *ADRC* mask is displayed for this purpose. Users can continue entering return codes until they enter a blank to return to the *DPRC* mask.

Return codes are entered in the *ADRC* mask in the following format:

```
llll  ... LEASY-specific return code
```

A dump cannot be created for return codes *L000, LI01, LI12, LP10, LP19, LS01, LS30, LS31, LS35, LS36, LS68* and *LU11*.

Return codes which contain a DMS code can only be specified generally for a file type in the following format:

```
   dXXX       d ... A (AIM file)
                    B (BIM file)
                    C (catalog file)
                    D (primary file)
                    J (job variables)
                    S (secondary index file)
                    T (status file)
```

LS17, for example, is converted to *JXXX*. A memory dump is output if a DMS error is reported for the specified file type.

i Use of the *ADD* subfunction is restricted to the main master.

Subfunction DEL

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 031: DELETE RETURNCODE
```

The *DEL* subfunction (enter: *D*) is used for deleting return codes. The *DLRC* mask is displayed for this purpose.

The return code to be deleted must be specified in the form in which it appears in the *DPRC* mask. If *A* (ALL) is entered, all the return codes entered are deleted.

Users can continue deleting return codes until they enter a blank or *A* (ALL) to return to the *DPRC* mask.

*END Terminate LEASY-MASTER

The **END* function is used to terminate the LEASY-MASTER utility routine immediately without having to exit via the catalog menu.

FILT Show table of all files of common memory CMMAIN

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 009: TOTAL FILE TABLE
```

and then displays the name of the selected common memory CMMAIN.

The following information is output in one line from left to right for each file transferred from the LEASY catalog:

- file number
- logical file name
- FCBTYPE (SAM, ISAM, PAM, DAM)
- LEASY file type (S, F, T, M)
- selected save method for this file (AIM, BIM; A+B stands for AIM and BIM)
- number of secondary keys defined for this file
- type of any file lock enforced by means of the LEASY-MASTER utility routine
- DMS file linked to the logical name (except foreign files) or the instance names for model file groups.

Up to 15 files per mask are displayed. Paging forward is possible by entering "+" instead of a function code. If all the files have already been displayed and "+" is entered again, the first screen page of the table will be displayed again.

Each time a screen page has been edited, a new function code may be entered.

GENC Show general statistics counters

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 015: GENERAL STATISTIC COUNTER
```

The following information lines then appear:

name of the CMMAIN selected

counters from a LEASY session:

- number of LEASY calls
- number of *OPFL* calls
- number of file opening operations
- number of *OPTR* calls
- number of *CLTR* calls
- number of *SETL* calls
- number of read calls
- number of write calls
- number of BIM write calls
- number of AIM write calls
- number of AIM records
- number of calls from the lock protocol
- number of waits from the lock protocol
- number of rejected calls from the lock protocol
- number of transaction rollbacks
- number of deadlock rejections.

A new function code may then be entered.

GENT Show general information

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 003: GENERAL INFORMATION
```

The following information lines then appear:

name of the selected CMMAIN;

current LEASY session number of the CMMAIN;

CMMAIN status; one of the following:

- initialization phase during startup of the main LEASY task
- normal state, in which the CMMAIN is available to the LEASY runtime system
- ready for after-image reconstruction (the state after initiation of the LEASY-MAINTASK utility routine with the operand **USE=R* and preceding or during reconstruction)
- end of AIM reconstruction (the state after termination of the reconstruction program but before release of CMMAIN for user programs)
- the main task is inactive (the state following the LEASY-MASTER functions *TERM*, *CLOS*, *SHUT*)
- switching of the AIM file generations being processed (the state following the LEASY-MASTER functions *AIMI*, *AIMC* and *AIMW*)
- ready for *PETR* processing by openUTM
- *PETR* processing by openUTM active

CMMAIN control; one of the following:

- termination of the main LEASY task (the state following a LEASY-MASTER function *TERM*)
- termination state, in which no more new transactions are accepted (the state following a LEASY-MASTER function *CLOS*)
- suspended state, in which no more new transactions are accepted and active ones are rolled back (the state following a LEASY-MASTER function *SHUT*)
- general suspended state (the state following a LEASY-MASTER function *HOLD*)

Use of before-image saving (LEASY-MAINTASK statement **LOG*).

Use of after-image saving (LEASY-MAINTASK statement **LOG*); when AIM saving is activated, the current number of the AIM file generation also appears.

Number of connected tasks out of the possible maximum (LEASY-MAINTASK statement **TSK*).

Number of currently active parallel transactions out of the possible maximum (LEASY-MAINTASK statement **TRA*).

Number of files opened by all active transactions out of the possible maximum (LEASY-MAINTASK statement **FIL*).

Number of currently active inquiry-and-transaction mode applications out of the possible maximum (LEASY-MAINTASK statement **APP*).

Bucket pool memory size (in bytes) for lock and transaction elements.

Size (in bytes) of one bucket (memory unit) in the lock and transaction bucket pool.

Number of buckets in lock and transaction bucket pool.

Number of buckets requested for lock elements.

Number of buckets currently being used for transaction elements.

Number of buckets in the lock and transaction bucket pool not currently in use.

Number of all enforced lock elements (total number of locked records of all transactions).

Number of released lock elements that are available in the lock element bucket pool for reuse.

Status of SYSLST log switch.

Permission for action calls at CMMAIN (privileged/non-privileged users).

A new function code may then be entered.

HELP Help and function selection

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 002: FUNCTION-SELECTION
```

The current LEASY catalog is also output.

The mask contains all permitted functions in the order specified below and ends with

Prompt

```
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION: OR *END=END OF PROGRAMM)
```

Possible responses

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 task is terminated.
The general information mask then appears.

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.

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

LOCT
A transaction is locked.
The transactions to be locked are 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.

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

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

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

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.

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

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

IOTE Termination of an I/O task.

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

REPO The original files are replaced by shadow files.

ROMS READ-ONLY mode is set.

ROMR READ-ONLY mode is reset.

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.

TRAT The transaction table is displayed in a separate mask.

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

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.

TSKT The task table is displayed in a separate mask.

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

THAT The table of inquiry-and-transaction mode applications is displayed in a separate mask.

ℹ *UTMA* may also be specified instead of *THAT* for reasons of compatibility.

AIMS The status of the AIM file generations is displayed.

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.

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

HELP All possible utility routine functions are displayed in a separate mask.

PINF ISAM pool information is displayed in a separate mask.

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

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 SYSLST logging is deactivated.
The general information mask then appears.

_ 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.

Invalid function codes are acknowledged by displaying the function selection mask. This applies also where the invalid function code was entered in a mask other than the function selection mask.

The functions are displayed on three screen pages.

- The first screen page contains all the control functions.
- The second screen page contains all the display functions.
- The third screen page contains the print functions.
- Enter "+" to scroll to the next page.

Only the second and third screen pages are displayed to nonprivileged users.

HOLD Suspend

The *HOLD* function locks all the active transactions. New transactions are rejected.

Effect

During this wait state, all calls at the program interface are rejected by the LEASY runtime system (return code).

The lock is ineffective against rollback calls by openUTM itself, i.e. against the KDCS operations *RSET*, *PEND ER* and automatic rollback in the event of an error.

The wait state is displayed in the general information mask and can be canceled by the *CONT* function.

The general information mask then appears.

IOGT Show general I/O task table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 022: GENERAL IO TASK TABLE
```

and displays the name of the selected common memory CMMAIN.

Then the following information lines appear:

- number of I/O tasks
- number of connected user programs
- number of jobs carried out by the I/O tasks
- size of the common memory in bytes
- length of the transfer area
- maximum length of the *AR* area in bytes
- maximum length of the *DB* file allocation in bytes
- maximum length of the *KB* area in bytes
- maximum length of the *KE* area in bytes
- maximum waiting time for a response from the I/O task
- queue processing of the I/O tasks
- specification as to whether or not the semaphore (protected variable) of a TSN contains data

A new function code can then be entered.

IOQA Show I/O task queue allocation

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 025: IO TASK QUEUING AVAILABILITY
```

and the name of the selected common memory CMMAIN is displayed.

Then the following information lines appear:

- number of available user tasks
- number of free I/O tasks

IOTE Terminate I/O task

The mask appears with the message

```
LEASY MASTER PROGRAM V6.2A SCREEN 026: TERMINATE IO TASK
```

and the name of the selected common memory CMMAIN is displayed.

A termination request is issued to an I/O task for each *IOTE* function call.

If only one I/O task is (still) active, the user is asked

```
DO YOU REALLY WANT TO TERMINATE THE LAST IO TASK (Y=YES; N=NO)?
```

If the user responds with *Y*, a termination request is issued to the I/O task. LEASY-MASTER then detaches itself from the IOH memory pool.

The following message is used to inform the LEASY administrator that a termination request has been issued:

```
LEA5502 TERMINATION OF AN IO TASK INITIATED
```

The specification of a function code can then be used to switch immediately to the next desired function.

IOTT Show I/O task table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 024: IO TASK TASK TABLE
```

Then the following information lines appear:

- name of the selected common memory CMMAIN
- number of active I/O tasks from the maximum possible number
- the active tasks (numbered consecutively and identified by their TSNs with the number of LEASY calls which have been executed)

The number of output lines per page is 15. If the total number of tasks is greater than 15, only 15 tasks are displayed in any one screen page. Paging forwards is possible by entering "+" instead of a function code. If all the tasks have already been displayed and "+" is entered, the first screen page of the task table will be displayed again. Each time a page has been displayed, a new function code may be entered.

IOUT Show I/O task user table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 023: IO TASK USER TABLE
```

Then the following information lines appear:

- name of the selected common memory CMMAIN
- number of active user tasks from the maximum possible number
- the active tasks (numbered consecutively and identified by their TSNs with the number of LEASY calls which have been issued)

The number of output lines per screen is 15. If the total number of tasks is greater than 15, only 15 tasks are displayed on any one screen page. Paging forwards is possible by entering "+" instead of a function code. If all the tasks have already been displayed and "+" is entered, the first screen page of the user table will be displayed again.
Each time a page has been displayed, a new function code may be entered.

LOCF Lock file

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 013: LOCK FILE
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER LOGICAL NAME OF FILE YOU WANT TO LOCK (BLANK = NO LOCK)
```

Possible responses

file[,U]
Logical file name of the master, foreign or temporary file (≤ 8 characters) that is to be locked.

file.[,U]
Name of a model file group that is to be locked (≤ 9 characters including the period).

file.suffix[,U]
Name of an instance of a model file group that is to be locked (≤ 29 characters).

U
The additional operand enables the file(s) to be locked against opening in write mode only; opening in read mode is then allowed. If the operand is not specified, no files are allowed to be opened.

If a file lock of the other type has already been set, it is replaced by the new specification.

If a file lock has been set, this is shown in the file table in the column *LOCKED FROM MASTER*.

_
If a mistake has been made and a file lock is no longer desired for processing, at least 1 blank should be entered.

Effect

A file lock affects both physical (*OPFL*) and logical (*OPTR*) opening. Depending on the specification of the additional operand *U*, either only openings in read mode are permitted for the file(s) involved or no openings at all.

Subsequent specification of a function code causes an immediate switch to the function required.

The lock must be used cautiously with foreign files because only the logical assignment is locked and not the DMS file as such. This can lead to difficulties where several tasks use several foreign files and have not assigned these in the same manner, or where in different tasks different DMS files are assigned to one logical file name.

LOCT Lock transaction

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 006: LOCK TRANSACTION
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER
OR IDENTIFICATION OF TRANSACTION YOU WANT TO LOCK
(BLANK=NO LOCK)
```

Possible responses

trnum
The numeric transaction number to be locked should be entered here. The transaction number is visible in the transaction table in the column *SESSION-SPEC TRANS-NUMBER*.

trsel
The transaction identification comprises three parts, i.e. the name of processor, the application name and the user name.

Those parts of the transaction identification not used for selection must be marked with an asterisk (*); the individual parts must be specified separated by commas. Superfluous identification parts on the right may be omitted.

Example of selection in timesharing mode:

```
*,$DIALOG
```

Example of selection in inquiry and transaction mode:

```
*,*,ABC
```

A partial qualification can be obtained by using the asterisk (*) appropriately. All transactions belonging to the specifically selected identification group are locked.

Specification of *C* causes all openUTM transactions of processor *C* to be locked. Once a transaction lock is set, it will be shown in the transaction table. If the transactions were previously set to ROLLBACK (*RLBT* functions), no lock is set.

i If a mistake has been made and a transaction lock is no longer desired, at least one blank should be entered instead of *trnum* or *trsel*.

Effect

The following operations are prohibited for the transactions involved (return code):

- all record operations
- all types of *CLTR*.

The lock is ineffective against rollback calls by openUTM itself, i.e. against the KDCS operations *RSET*, *PEND ER* and automatic openUTM rollback in the case of errors.

The lock can be canceled by means of the *CONT* function.

It is then possible to switch over immediately to the next function required by specifying a function code.

OFFP/ONPR

Printout

The following applies to **interactive mode**:

- There is no printout as long as the *ONPR* function code has not been set.
  When the *ONPR* function code is set, the entire dialog for the selected CMMAIN is logged to SYSLST from that moment on.
- When the *OFFP* function code is set, SYSLST logging is suppressed during further processing.
- Activation/deactivation of logging is indicated by a separate print line which includes the date and time.

The following applies to **batch mode**:

- After the load message `LEA0501`, the system file SYSOUT is assigned to the dummy file *DUMMY and SYSLST logging is activated. When the program is terminated, the system file SYSOUT is returned to its primary assignment

  ```
  (/ASSIGN-SYSOUT TO-FILE=*PRIMARY).
  ```
- The output appears in the format displayed on the data terminal, i.e. always two screen pages per print page. Each screen page logged includes the date and time.

OPFT Show table of open files

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 010: OPEN FILE TABLE
```

and displays the name of the selected common memory CMMAIN and the total number of open master and model files. Foreign and temporary files are not contained in the table.

One line is output for each file, containing the following specifications from left to right:

- file number
- logical file name
- FCBTYPE (SAM, ISAM, PAM, DAM)
- LEASY file type (S, M)
- selected save method (AIM, BIM; A+B stands for AIM and BIM)
- number of tasks that have opened the file
- number of transactions that have opened the file for retrieval
- number of transactions that have opened the file other than for retrieval
- specification of the internal number of the transaction locking the file, if such a lock exists
- specification of whether a file lock is currently being enforced by the LEASY-MASTER utility routine (function code *LOCF*)
- specification of whether records in a file are locked

A maximum of 14 files per mask can be displayed. If the number of files opened exceeds this, then 14 are displayed at a time. Paging forward is possible by entering "+" instead of a function code. If all opened files have already been displayed and "+" is entered again, the first screen page of the table will be displayed again.

Each time a screen page has been edited, a new function code may be entered.

PINF Show ISAM pool information

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 027: ISAM POOL TABLE
```

and then displays the name of the selected common memory CMMAIN.

For each ISAM pool transferred from the LEASY catalog, the following information is output in one line from left to right:

- pool name (pool link name)
- pool size (**STD* or size in PAM pages)
- pool creation mode (runtime system or main task)
- pubset on which the ISAM pool was created.

Up to 16 ISAM pool information lines are displayed per mask. If the number of information lines exceeds 16, forward scrolling is possible with the subsequent function code input by entering "+". If all ISAM pool information lines have already been displayed, "+" leads to a renewed display of the first screen.

A new function code may be input every time a screen has been completely filled.

QUIE Suspend

The *QUIE* function impedes the opening of new transactions. No active transactions are impeded.

Effect

The following operations are permitted at the program interface:

- *CATD* in all variations,
- *OPFL, CLFL, CINF, CLTR*
- all record operations

The `OPTR` operation is prohibited (return code: LS06).

In the case of *CLTR* operations with *OPE2=T*, the new separate transaction is locked internally after the operation, thereby obtaining the same effect as with *LOCT*.

The wait state is displayed in the general information mask and can be canceled by the *CONT* function.

The general information mask then appears.

REPO Replace original file by shadow file during ongoing operation

i The *REPO* function is available only if automatic reconstruction of shadow files is being used.

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 039: COPY SHADOWFILE
```

and then displays the name of the selected common memory CMMAIN.

The following information lines then appear:

- Wait time for terminating the transactions
- Response to transactions that have not been terminated

Prompt

```
FUNCTION SELECTION (OR R=REACTION, IN CASE OF OPEN TRANSACTIONS AFTER
WAITING TIME; OR W=ENTER WAITING TIME; OR F=FILE SELECTION;
OR S=START FUNCTION PROCESSING; OR BLANK=MAINTASK SELECTION;
OR *END=END OF PROGRAM)
```

Possible responses

Function code The function selected is executed immediately.

F Selects the files whose associated shadow files are to be brought into a current and consistent state using the automatic RECONST and then copied to the original files (FILE SELECTION)

W Specifies the wait time which applies when waiting for termination of the transactions which are accessing at least one of the selected files (WAITING TIME)

R Further procedure if not all transactions concerned have terminated after the wait time has elapsed (REACTION)

S Starts actual processing of the function (START FUNCTION)

Subfunction F (FILE SELECTION)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 042: FILE SELECTION
```

and then displays the name of the selected common memory CMMAIN and a list of the files that have already been selected. The first time the subfunction is called, no files are selected.

Prompt

```
FILE SELECTION (A=ADD FILENAME; OR R=REMOVE FILENAME;
OR E=END OF FILE SELECTION)
```

Possible responses

A Add file
Specifies the logical file name of a file which is to be added to the list of selected files. The file must be contained in the detailed list of the Common Memory Pool and automatic recording of shadow files must have been defined for it.
The mask for adding files is displayed repeatedly. This enables you to add all the files required one after the other. To display the updated mask of the FILE SELECTION subfunction, enter a blank instead of a file name.

i The selected files and any associated SI files must be opened using SHARED-UPDATE=YES.

They must be master files with the access method ISAM or PAM.

The selected files must be specified using *AIM= (Y,A)* or *AIM=(R,A)* in the **FIL* statement of the LEASY-CATALOG utility routine.

Files which fail to satisfy one or more of these conditions are ignored and a message is issued.

R Remove file
Specifies the logical file name of a file which is to be removed from the list of selected files.
The mask for removing files from the list is displayed repeatedly. This enables you to remove multiple files one after the other. To display the updated mask of the FILE SELECTION subfunction, enter a blank instead of a file name.

E End
Displays the updated mask of the *REPO* function.

Subfunction W (WAITING TIME)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 040: ENTER WAITING TIME
```

Prompt

```
PLEASE ENTER THE TIME TO WAIT FOR THE COMPLETION OF NOT YET CLOSED
TRANSACTIONS. (0<=WAITING TIME<=120; BLANK=5 MINUTES (STANDARD VALUE))
```

Possible responses

Wait time which applies when waiting for termination of transactions which are working with at least one of the files which have been selected or are still to be selected for the *REPO* function. The wait time can be between 0 and 120 minutes. If a blank is entered or the WAITING TIME subfunction is not called, the default value for the wait time is 5 minutes.

The updated mask of the *REPO* function is then displayed.

Subfunction R (REACTION)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2 SCREEN 041: REACTION AT END OF WAITING TIME
```

Prompt

```
PLEASE ENTER REACTION IF THERE ARE UNFINISHED TRANSACTIONS AT THE END OF
WAITING TIME
R=RESET UNFINISHED TRANSACTIONS AND RESUME FUNCTION PROCESSING;
I=PROCESS ONLY THOSE FILES, WHOSE CORRESPONDING TRANSACTIONS ARE FINISHED
  AFTER END OF WAITING TIME;
C=CANCEL FUNCTION, IF THERE ARE OPEN TRANSACTIONS AFTER END OF WAITING TIME;
```

Possible responses

Further procedure if not all transactions concerned have terminated after the wait time has elapsed:

R Reset all open transactions affected by the *REPO* function and continues the *REPO* function.

Note: Transactions which refer to files for which no before-image saving is defined cannot be reset.

I Continue the *REPO* function; only files whose associated transactions have terminated by the end of the wait time are replaced.

C Abort the *REPO* function.

The updated mask of the *REPO* function is then displayed.

Subfunction S (START FUNCTION)

Starts execution of the *REPO* function with the specified settings.

RLBT Roll back transaction

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 011: ROLLBACK TRANSACTION
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER
OR IDENTIFICATION YOU WANT TO ROLLBACK
(BLANK=NO ROLLBACK)
```

Possible responses

trnum

The numeric transaction number of the transaction to be rolled back is entered. The transaction number is shown in the transaction table in the column *SESSION-SPEC TRANS-NUMBER*.

trsel

The transaction identification comprises three parts, i.e. the name of processor, the application name and the user name.

Those parts of the transaction identification not used for selection must be marked with an * (asterisk); the individual parts must be specified separated by commas. Superfluous identification parts on the right may be omitted.

A partial qualification can be obtained by using the asterisk (*) appropriately. All transactions belonging to the specifically selected identification group are then rolled back.

A rollback function set for a transaction is shown in the transaction table; a previously set lock on the transaction (*HOLD* or *LOCT* function) is canceled.

_

If a mistake has been made and a transaction rollback is no longer desired, at least one blank should be entered instead of *trnum* or *trsel*.

Procedure for active transactions

The assigned task is active and can process further LEASY operations.

The following operations are permitted at the program interface:

- `CLTR` and
- `CLTR` with `OPE1=R`

The following transaction operations are modified as appropriate (return code):

- `CLTR` with `OPE1=' '` or `OPE1='R'` is retained
- `OPE2=T` becomes `OPE2=' '`
- all other operations become `CLTR` with `OPE1='R'`

It is then possible to switch immediately to the next function desired by specifying a function code.

Procedure for aborted transactions

The assigned task is no longer available in the system. The LEASY runtime system cannot process any further operations.

If the LEASY STXIT routine was activated when the task was aborted, then the rollback operation is initiated by this routine. If the STXIT routine is not activated the transaction remains open and hence the application remains in an inconsistent state.

In TIAM mode the rollback operation can be performed under the same USERID or under a privileged USERID.

If these requirements are not fulfilled, LEASY-MASTER cannot perform a rollback.

Procedure for selection of active and aborted transactions

If transactions are selected via *trsel*, the partial qualification option can be used to select active and aborted transactions.

In this case, the messages of the aborted transactions are output first. These consist of a start message, possibly an error message, and an end message.

The following message is then displayed for the active transactions:

```
LEA9573    ROLLBACK FOR OTHER TRANSACTION(S) WILL BE DISPLAYED
           IN THE TRANSACTION TABLE
```

A function code can then be specified to switch immediately to another function.

ROMR Reset READ-ONLYmode

The *ROMR* function cancels the write lock which was set using the *ROMS* function. This enables write operations to be performed again on files which were selected using the *ROMS* function.

⚠ **CAUTION!**

The *ROMR* function may be executed only if the online save operation for which the write lock was used has been completed, otherwise inconsistent save files can be created.

The *ROMR* function is rejected during a save operation using LEASY-SAVE.

ROMS Set READ-ONLY mode

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 045: SET READONLY-MODE
```

and then displays the name of the selected common memory Common Memory CMMAIN.

The following information lines then appear:

- Wait time for terminating the transactions
- Response to transactions that have not been terminated

Prompt

```
FUNCTION SELECTION (OR R=REACTION, IN CASE OF OPEN TRANSACTIONS AFTER
WAITING TIME; OR W=ENTER WAITING TIME; OR F=FILE SELECTION;
OR S=START FUNCTION PROCESSING; OR BLANK=MAINTASK SELECTION;
OR *END=END OF PROGRAM)
```

Possible responses

Function code	The function selected is executed immediately.
F	Selects the files which are to be brought into a consistent state and assigned a write lock for the later online save (FILE SELECTION)
W	Specifies the wait time which applies when waiting for termination of the transactions which are accessing at least one of the selected files (WAITING TIME)
R	Further procedure if not all transactions concerned have terminated after the wait time has elapsed (REACTION)
S	Starts actual processing of the function (START FUNCTION)

Subfunction F (FILE SELECTION)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 042: FILE SELECTION
```

and then displays the name of the selected common memory CMMAIN and a list of the files that have already been selected. The first time the subfunction is called, no files are selected.

Prompt

```
FILE SELECTION (A=ADD FILENAME; OR R=REMOVE FILENAME;
OR E=END OF FILE SELECTION)
```

Possible responses

A Add file
Specifies the logical file name of a file which is to be added to the list of selected files. The file must be contained in the detailed list of the Common Memory Pool and it must be reserved using the LEASY-CATALOG utility routine (ROM=Y operand of the **CAT* or **FIL* statement).

The '*ALL' specification in place of a file name causes all the files reserved with the LEASY-CATALOG utility routine to be added.

The mask for adding files is displayed repeatedly. This enables you to add all the files required one after the other. To display the updated mask of the FILE SELECTION subfunction, enter a blank instead of a file name.

> The selected files and any associated SI files must be opened using SHARED-UPDATE=YES.
>
> They must be master files with the access method ISAM or PAM.
>
> If the subsequent online save is to take place using LEASY-SAVE, '*ALL' must be specified.
>
> The selected files must be reserved for setting READ-ONLY mode by specifying *ROM= YES* in the **CAT* or **FIL* statement of the LEASY-CATALOG utility routine.
>
> Files which fail to satisfy one or more of these conditions are ignored and a message is issued.

R Remove file
Specifies the logical file name of a file which is to be removed from the list of selected files.

The ’*ALL’ specification in place of a file name causes all the files reserved with the LEASY-CATALOG utility routine to be removed from the list.

The mask for removing files from the list is displayed repeatedly. This enables you to remove multiple files one after the other. To display the updated mask of the FILE SELECTION subfunction, enter a blank instead of a file name.

E End
Displays the updated mask of the *ROMS* function.

Subfunction W (WAITING TIME)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A    SCREEN 040: ENTER WAITING TIME
```

Prompt

```
PLEASE ENTER THE TIME TO WAIT FOR THE COMPLETION OF NOT YET CLOSED
TRANSACTIONS. (0<=WAITING TIME<=120; BLANK=5 MINUTES (STANDARD VALUE))
```

Possible responses

Wait time which applies when waiting for termination of the transactions which are working with at least one of the files which have been selected or are still to be selected for the *ROMS* function. The wait time can be between 0 and 120 minutes. If a blank is entered or the WAITING TIME subfunction is not called, the default value for the wait time is 5 minutes.

The updated mask of the *ROMS* function is then displayed.

Subfunction R (REACTION)

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2 SCREEN 041: REACTION AT END OF WAITING TIME
```

Prompt

```
PLEASE ENTER REACTION IF THERE ARE UNFINISHED TRANSACTIONS AT THE END OF
WAITING TIME
R=RESET UNFINISHED TRANSACTIONS AND RESUME FUNCTION PROCESSING;
I=PROCESS ONLY THOSE FILES, WHOSE CORRESPONDING TRANSACTIONS ARE FINISHED
  AFTER END OF WAITING TIME;
C=CANCEL FUNCTION, IF THERE ARE OPEN TRANSACTIONS AFTER END OF WAITING TIME;
```

Possible responses

Further procedure if not all transactions concerned have terminated after the wait time has elapsed:

R Reset all open transactions affected by the *ROMS* function and continue the *ROMS* function.

Note: Transactions which refer to files for which no before-image saving is defined cannot be reset.

I Continue the *ROMS* function; only files whose associated transactions have terminated by the end of the wait time are processed.

C Abort the *ROMS* function.

The updated mask of the *ROMS* function is then displayed.

Subfunction S (START FUNCTION)

Starts execution of the *ROMS* function with the specified settings.

It is then possible to switch immediately to the next function desired by specifying a function code.

SHLE Show lock elements

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 028: SHOW LOCK ELEMENTS
```

The *SHLE* (SHow Lock Elements) function displays the LEASY lock elements.

When the *SHLE* function is selected the following information is displayed in the *SHLE* mask for each lock element:

- internal number of the locking transaction
- logical name of the file to which the lock relates
- type of lock (READ=read lock, WRIT=write lock, UPDT=update lock)
- interval indicator
- key of the locked record

The way in which the key is shown depends on the FCBTYPE of the file.

ISAM ISAM key (blanks and non-printable characters are output as ’?’)

PAM hexadecimal block number (X’hhhhhh’)

DAM decimal record number

For locked intervals the interval indicator ’---’ precedes the second key.

i *SHLE* can also be called cyclically via the *CYCI* function. The number of remaining cycles is displayed at the bottom of the screen.

SHUT Terminate LEASY session immediately (emergency stop)

The *SHUT* function terminates LEASY-MAINTASK. After this, no more new transactions are permitted. Current transactions may be terminated (*CLTR*); any other call causes the transaction to be rolled back.

Effect

Operations permitted at the program interface:

- *CLTR, CLTR* with *OPE1=R*
- *CLFL*
- *CATD* for no CMMAIN or for a different one.

Prohibited operations (return code):

- *CATD* for this CMMAIN
- *OPFL, OPTR*.

Operations changed by the program interface (return code):

- *CLTR* `with` *OPE2=T* `becomes` *CLTR*
- *CLTR* `with` *OPE1=R* `and` *OPE2=T* `becomes` *CLTR* `with` *OPE1=R*
- `all record operations become` *CLTR* `with` *OPE1=R*

The general information mask then appears.

TERM Terminate the main task

The *TERM* function terminates the program run of the LEASY-MAINTASK utility routine. This function does not affect current transactions. New transactions are accepted as long as the common memory CMMAIN exists in the system.

THAT Display inquiry-and-transaction mode application table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 008: TH APPLICATION TABLE
```

and displays the name of the selected common memory CMMAIN and the number of active inquiry-and-transaction mode (TH) applications.

One line with the following information appears for each inquiry-and-transaction mode application connected to the CMMAIN:

- number of the table entry
- name of the inquiry-and-transaction mode application
- number of tasks in the inquiry-and-transaction mode task pool of the application

A maximum of 15 applications per screen page are displayed. Should the total number of applications exceed 15, paging forwards to the next screen page is possible by entering "+" instead of a function code. If the entire table has already been displayed and "+" is entered, the first screen page of the table will be displayed again.

Each time a screen page has been edited, a new function code may be entered.

openUTM test applications do not appear in the application table.

It is possible to specify *UTMA* instead of *THAT* for reasons of compatibility.

TRAC Display transaction-specific counters

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 019: SELECT TRANS STATISTIC
```

The name of the selected CMMAIN is then displayed together with a request for a transaction specification.

Prompt

```
PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER
OR IDENTIFICATION YOU WANT TO DISPLAY STATISTICALLY
```

Possible responses

trnum

The number of the transaction must be entered. The transaction number is shown in the transaction table in the column *SESSION-SPEC TRANS-NUMBER*.

trid

The transaction identification comprises three parts, i.e. the processor name, the application name and the user name.
The parts of the transaction identifier not used for selection must be marked with an asterisk (*); the individual parts must be separated by commas.
Superfluous identification parts on the right may be omitted.

_

If a mistake has been made and no transaction-specific counters require to be displayed, at least 1 blank should be entered instead of *trnum* or *trid*.

The display time for each screen page is then requested:

Prompt

```
PLEASE ENTER DISPLAY TIME (BLANK = >t< SECONDS; MIN = 0, MAX = 99)
```

Possible responses

t

The desired display time must be entered in numeric form for each information mask.
0 ≤ t ≤ 99.

_

The last entered display time t appearing in the prompt is used.
Initial setting: $t=16$ seconds.

The display time also roughly corresponds to the interval for retrieving information from the common memory CMMAIN. This makes it possible to trace modifications in the CMMAIN over an extended period of time without the need for external inputs.

Effect

The state of the selected transaction is displayed as in the transaction table, together with the following:

- number of all LEASY calls
- number of OPTR calls
- number of SETL calls
- number of read calls
- number of write calls
- number of BIM write calls
- number of AIM write calls
- number of AIM records
- number of calls from the lock log
- number of waits from the lock log
- number of rejected calls from the lock log
- number of locked records
- number of deadlock rejections
- A new function code may then be entered.

TRAT Display transaction table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 005: TRANSACTION TABLE
```

The name of the selected common memory CMMAIN and the number of active transactions are then displayed.

The following information is displayed in one line for each transaction:

- *name of the processor* for openUTM transactions in productive mode, *UTM + tsn* for openUTM transactions in test mode, *DCA + tsn* for DCAM applications, *TSN + tsn* for transactions in timesharing mode (*tsn* = task sequence number)
- identification name ($DIALOG in timesharing mode, application name in inquiry and transaction mode)
- user name for openUTM transactions, user ID for transactions in timesharing mode
- number of open file identifiers in the transaction
- internal transaction number (ITN)
- current transaction number of the session
- transaction state (active, locked or to be rolled back)

Only the ITN is displayed for inactive transactions; all other fields are deleted.

14 lines are displayed per screen page. If the total number of transactions exceeds this, then 14 transactions are displayed at a time on one screen page. Paging forwards is possible by entering "+" instead of a function code. If all transactions have already been displayed and "+" is entered again, the first screen page of the transaction table will be displayed again.

Each time a screen page has been edited, a new function code may be entered.

TSKC Display counters of a particular task

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 020: SELECT TASK FOR STATISTIC
```

The following information line then appears:

The name of the selected CMMAIN is displayed.

Prompt

```
PLEASE ENTER TASK SEQUENCE NUMBER FOR TASK
YOU WANT TO DISPLAY STATISTICALLY
```

The same counters now appear for the TSN entered as with the *GENC* function for all TSNs; the counters relate only to the selected task.

A new function code may then be entered.

TSKT Display task table

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 018: TASK TABLE
```

The following information lines then appear:

- the name of the selected CMMAIN is displayed
- number of active tasks (out of the possible maximum)
- the active tasks are numbered consecutively and identified by the TSN and task type.

Only the task number is displayed for inactive tasks; all other fields are deleted.

There are 15 output lines per screen page. If the total number of tasks exceeds this, then 15 tasks at a time are displayed on a screen page. Paging forward is possible by entering "+" instead of a function code. If all tasks have already been displayed and "+" is entered again, the first screen page of the task table will be displayed again.

Each time a screen page has been edited, a new function code may be entered.

UNLF Unlock file

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 014: UNLOCK FILE
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER LOGICAL NAME OF FILE YOU WANT TO UNLOCK
(BLANK = NO UNLOCK)
```

Possible responses

file Logical name of a master, foreign or temporary file (≤ 8 characters) to be unlocked.

file. The name of a model file group to be unlocked (≤ 9 characters including the period).

file.suffix The name of an instance of a model file group to be unlocked (≤ 29 characters).

_ If a mistake has been made and the file is no longer required to be unlocked, at least 1 blank should be entered.

It is then possible to switch to the next function desired by entering a function code.

UNLT Unlock transaction

The mask appears with the message

```
LEASY MASTER PROGRAM VERSION V6.2A SCREEN 007: UNLOCK TRANSACTION
```

and displays the name of the selected common memory CMMAIN.

Prompt

```
PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER
OR IDENTIFICATION YOU WANT TO UNLOCK
(BLANK=NO UNLOCK)
```

Possible responses

trnum — The numeric transaction number to be unlocked should be entered here.

trsel — The transaction identification comprises three parts, i.e. the name of processor, the application name and the user name.

Those parts of the transaction identification not used for selection must be marked with an asterisk (*); the individual parts must be specified separated by commas. Superfluous identification parts on the right may be omitted.

A partial qualification can be obtained by using the asterisk (*) appropriately. All transactions are then released that belong to the specifically selected group.

An existing transaction lock is shown in the transaction table.

_ — If the transaction is no longer required to be unlocked, at least one blank should be entered instead of *trnum* or *trsel*.

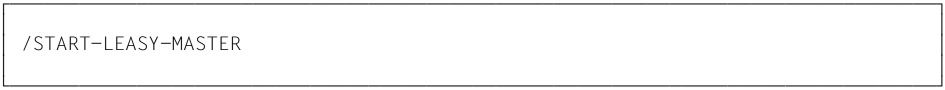
Any desired function code may then be entered.

8.3 Example

This section contains a description of some of the functions of the LEASY-MASTER utility routine in the form of an example.

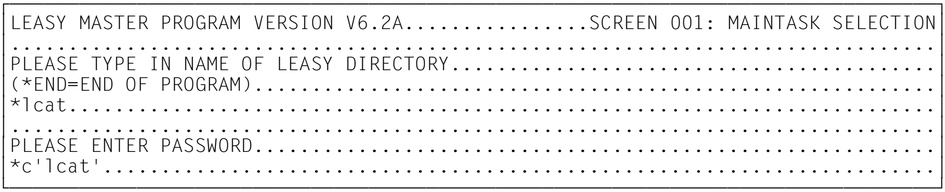
The messages and responses displayed within a frame are output to one screen page. The statement to be executed is entered directly after the asterisk in the last line of the screen.

Calling the LEASY-MASTER utility routine



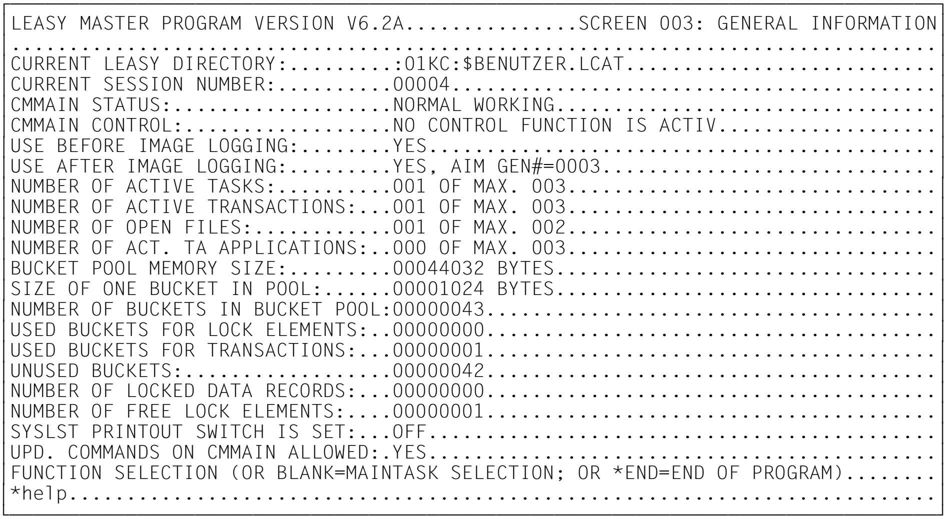
```
/START-LEASY-MASTER
```

(1) Common memory selection



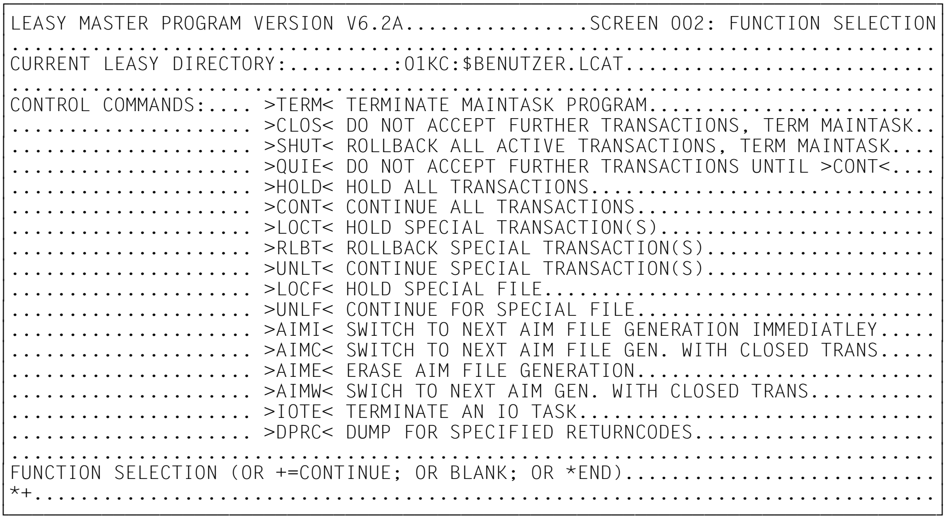
```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 001: MAINTASK SELECTION
................................................................................
PLEASE TYPE IN NAME OF LEASY DIRECTORY..........................................
(*END=END OF PROGRAM)...........................................................
*lcat...........................................................................
................................................................................
PLEASE ENTER PASSWORD...........................................................
*c'lcat'........................................................................
```

(2) General Information

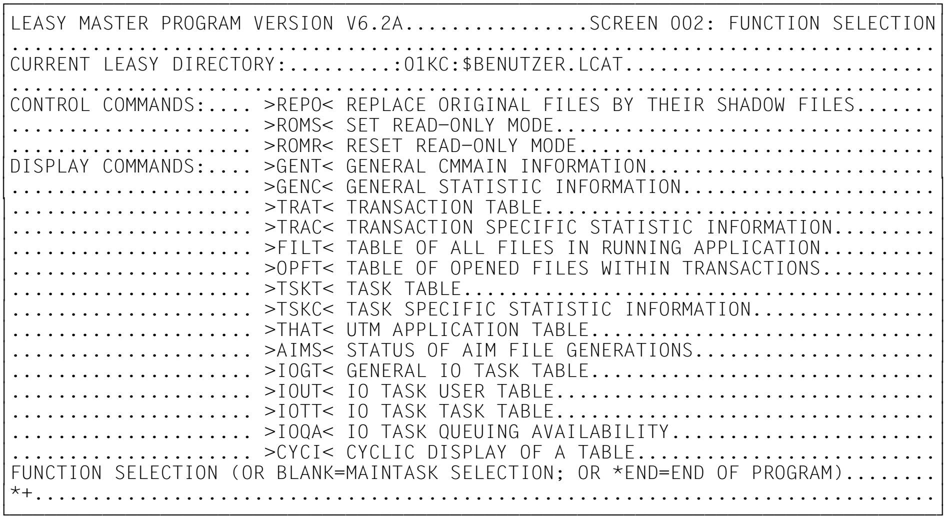


```
LEASY MASTER PROGRAM VERSION V6.2A..............SCREEN 003: GENERAL INFORMATION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
CURRENT SESSION NUMBER:..........00004..........................................
CMMAIN STATUS:...................NORMAL WORKING.................................
CMMAIN CONTROL:..................NO CONTROL FUNCTION IS ACTIV...................
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........YES, AIM GEN#=0003.............................
NUMBER OF ACTIVE TASKS:..........001 OF MAX. 003................................
NUMBER OF ACTIVE TRANSACTIONS:...001 OF MAX. 003................................
NUMBER OF OPEN FILES:............001 OF MAX. 002................................
NUMBER OF ACT. TA APPLICATIONS:..000 OF MAX. 003................................
BUCKET POOL MEMORY SIZE:.........00044032 BYTES.................................
SIZE OF ONE BUCKET IN POOL:......00001024 BYTES.................................
NUMBER OF BUCKETS IN BUCKET POOL:00000043.......................................
USED BUCKETS FOR LOCK ELEMENTS:..00000000.......................................
USED BUCKETS FOR TRANSACTIONS:...00000001.......................................
UNUSED BUCKETS:..................00000042.......................................
NUMBER OF LOCKED DATA RECORDS:...00000000.......................................
NUMBER OF FREE LOCK ELEMENTS:....00000001.......................................
SYSLST PRINTOUT SWITCH IS SET:...OFF............................................
UPD. COMMANDS ON CMMAIN ALLOWED:.YES............................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*help...........................................................................
```

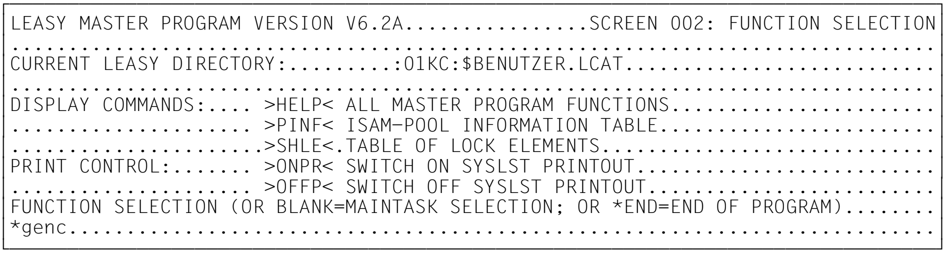
(3) Function selection (3 pages)



```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 002: FUNCTION SELECTION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT..........................
................................................................................
CONTROL COMMANDS:.... >TERM< TERMINATE MAINTASK PROGRAM.........................
..................... >CLOS< DO NOT ACCEPT FURTHER TRANSACTIONS, TERM MAINTASK..
..................... >SHUT< ROLLBACK ALL ACTIVE TRANSACTIONS, TERM MAINTASK....
..................... >QUIE< DO NOT ACCEPT FURTHER TRANSACTIONS UNTIL >CONT<....
..................... >HOLD< HOLD ALL TRANSACTIONS..............................
..................... >CONT< CONTINUE ALL TRANSACTIONS..........................
..................... >LOCT< HOLD SPECIAL TRANSACTION(S)........................
..................... >RLBT< ROLLBACK SPECIAL TRANSACTION(S)....................
..................... >UNLT< CONTINUE SPECIAL TRANSACTION(S)....................
..................... >LOCF< HOLD SPECIAL FILE..................................
..................... >UNLF< CONTINUE FOR SPECIAL FILE..........................
..................... >AIMI< SWITCH TO NEXT AIM FILE GENERATION IMMEDIATLEY.....
..................... >AIMC< SWITCH TO NEXT AIM FILE GEN. WITH CLOSED TRANS.....
..................... >AIME< ERASE AIM FILE GENERATION..........................
..................... >AIMW< SWICH TO NEXT AIM GEN. WITH CLOSED TRANS...........
..................... >IOTE< TERMINATE AN IO TASK...............................
..................... >DPRC< DUMP FOR SPECIFIED RETURNCODES.....................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*+..............................................................................
```

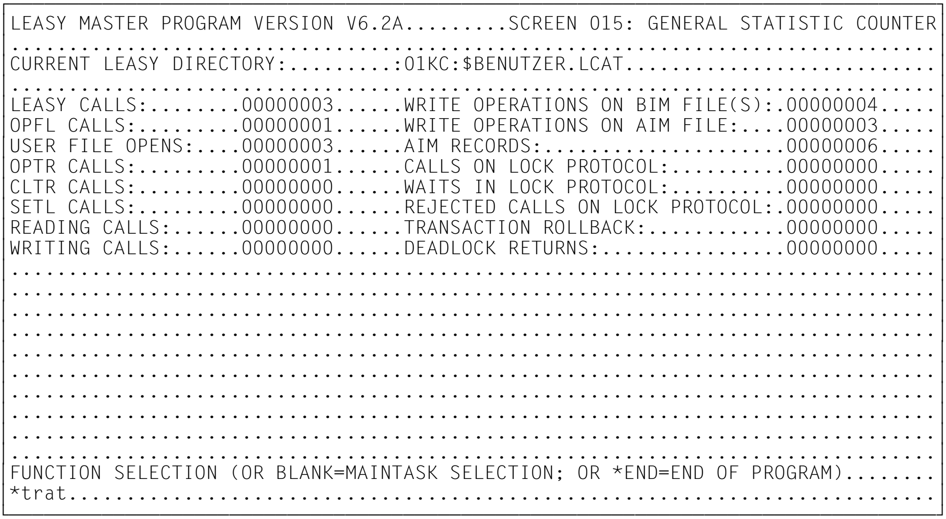


```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 002: FUNCTION SELECTION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT..........................
................................................................................
CONTROL COMMANDS:.... >REPO< REPLACE ORIGINAL FILES BY THEIR SHADOW FILES.......
..................... >ROMS< SET READ-ONLY MODE.................................
..................... >ROMR< RESET READ-ONLY MODE...............................
DISPLAY COMMANDS:.... >GENT< GENERAL CMMAIN INFORMATION.........................
..................... >GENC< GENERAL STATISTIC INFORMATION......................
..................... >TRAT< TRANSACTION TABLE..................................
..................... >TRAC< TRANSACTION SPECIFIC STATISTIC INFORMATION.........
..................... >FILT< TABLE OF ALL FILES IN RUNNING APPLICATION..........
..................... >OPFT< TABLE OF OPENED FILES WITHIN TRANSACTIONS..........
..................... >TSKT< TASK TABLE.........................................
..................... >TSKC< TASK SPECIFIC STATISTIC INFORMATION................
..................... >THAT< UTM APPLICATION TABLE..............................
..................... >AIMS< STATUS OF AIM FILE GENERATIONS.....................
..................... >IOGT< GENERAL IO TASK TABLE..............................
..................... >IOUT< IO TASK USER TABLE.................................
..................... >IOTT< IO TASK TASK TABLE.................................
..................... >IOQA< IO TASK QUEUING AVAILABILITY.......................
..................... >CYCI< CYCLIC DISPLAY OF A TABLE..........................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*+..............................................................................
```



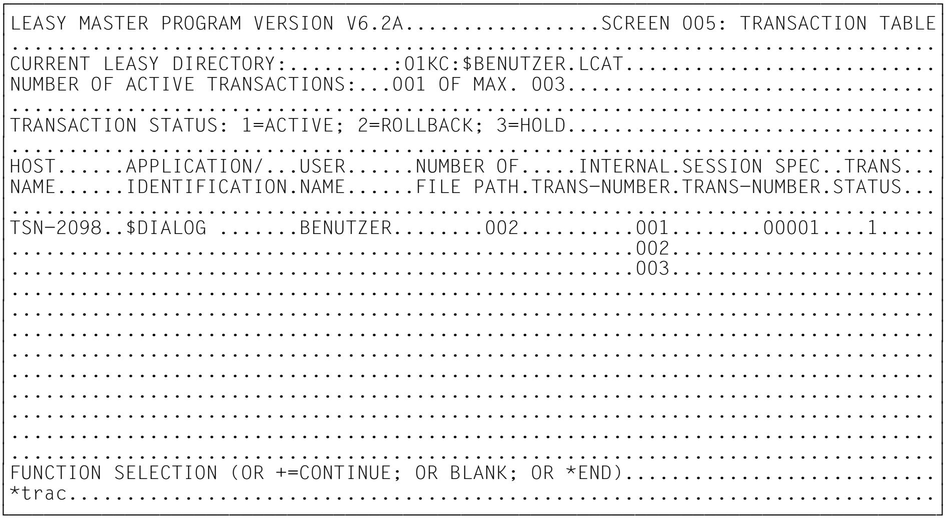
```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 002: FUNCTION SELECTION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
................................................................................
DISPLAY COMMANDS:.... >HELP< ALL MASTER PROGRAM FUNCTIONS.......................
..................... >PINF< ISAM-POOL INFORMATION TABLE........................
......................>SHLE<.TABLE OF LOCK ELEMENTS.............................
PRINT CONTROL:....... >ONPR< SWITCH ON SYSLST PRINTOUT..........................
..................... >OFFP< SWITCH OFF SYSLST PRINTOUT.........................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*genc...........................................................................
```

(4) General counters



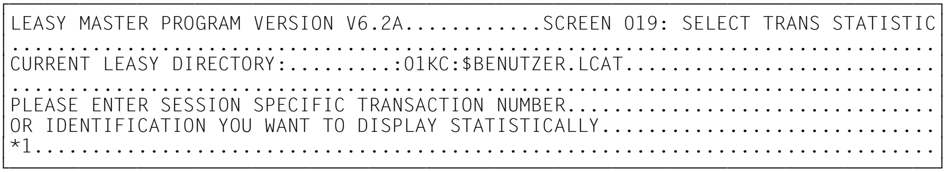
```
LEASY MASTER PROGRAM VERSION V6.2A.........SCREEN 015: GENERAL STATISTIC COUNTER
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
................................................................................
LEASY CALLS:........00000003......WRITE OPERATIONS ON BIM FILE(S):.00000004.....
OPFL CALLS:.........00000001......WRITE OPERATIONS ON AIM FILE:....00000003.....
USER FILE OPENS:....00000003......AIM RECORDS:.....................00000006.....
OPTR CALLS:.........00000001......CALLS ON LOCK PROTOCOL:..........00000000.....
CLTR CALLS:.........00000000......WAITS IN LOCK PROTOCOL:..........00000000.....
SETL CALLS:.........00000000......REJECTED CALLS ON LOCK PROTOCOL:.00000000.....
READING CALLS:......00000000......TRANSACTION ROLLBACK:............00000000.....
WRITING CALLS:......00000000......DEADLOCK RETURNS:................00000000.....
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*trat...........................................................................
```

(5) Transaction table



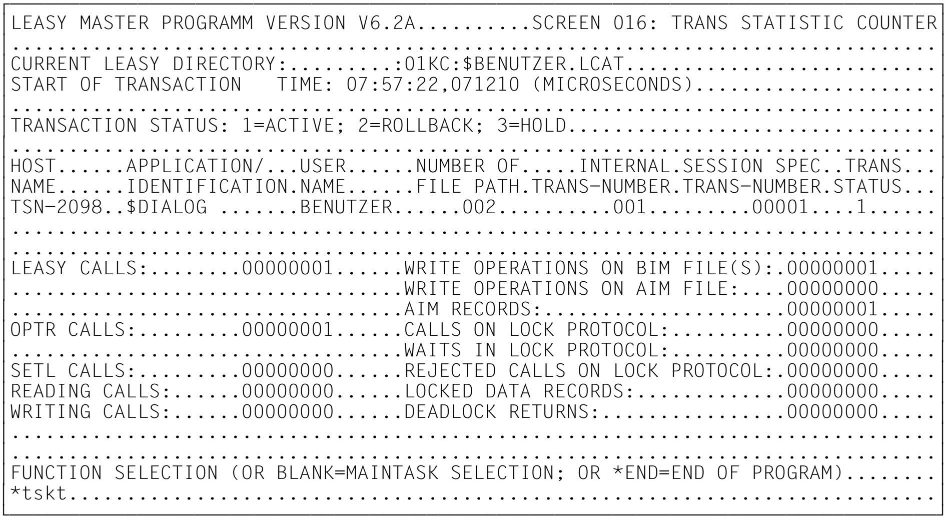
```
LEASY MASTER PROGRAM VERSION V6.2A.................SCREEN 005: TRANSACTION TABLE
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
NUMBER OF ACTIVE TRANSACTIONS:...001 OF MAX. 003................................
................................................................................
TRANSACTION STATUS: 1=ACTIVE; 2=ROLLBACK; 3=HOLD................................
................................................................................
HOST......APPLICATION/...USER......NUMBER OF.....INTERNAL.SESSION SPEC..TRANS...
NAME......IDENTIFICATION.NAME......FILE PATH.TRANS-NUMBER.TRANS-NUMBER.STATUS...
................................................................................
TSN-2098..$DIALOG .......BENUTZER........002..........001........00001....1.....
......................................................002.......................
......................................................003.......................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*trac...........................................................................
```

(6) Transaction-specific counter selection



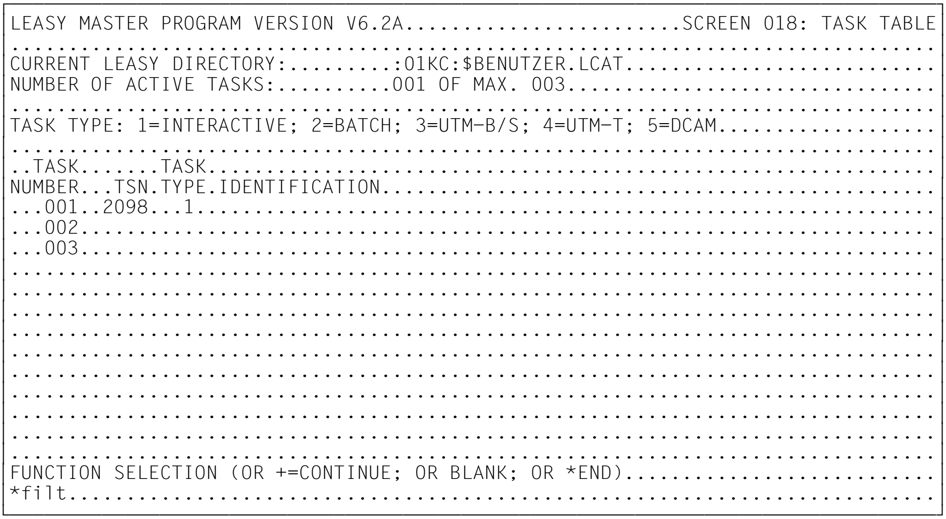
```
LEASY MASTER PROGRAM VERSION V6.2A............SCREEN 019: SELECT TRANS STATISTIC
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
................................................................................
PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER................................
OR IDENTIFICATION YOU WANT TO DISPLAY STATISTICALLY.............................
*1..............................................................................
```

(7) Transaction-specific counters



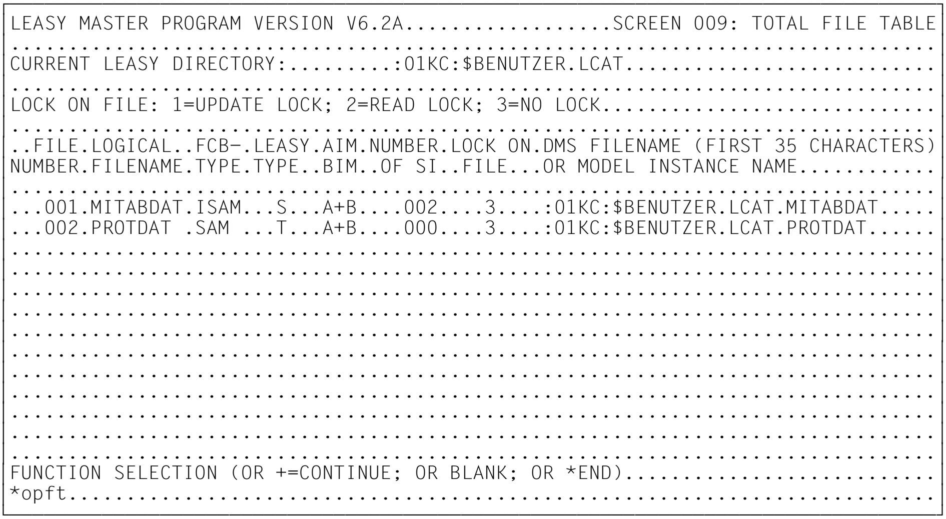
```
LEASY MASTER PROGRAMM VERSION V6.2A..........SCREEN 016: TRANS STATISTIC COUNTER
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
START OF TRANSACTION   TIME: 07:57:22,071210 (MICROSECONDS).....................
................................................................................
TRANSACTION STATUS: 1=ACTIVE; 2=ROLLBACK; 3=HOLD................................
................................................................................
HOST......APPLICATION/...USER......NUMBER OF.....INTERNAL.SESSION SPEC..TRANS...
NAME......IDENTIFICATION.NAME......FILE PATH.TRANS-NUMBER.TRANS-NUMBER.STATUS...
TSN-2098..$DIALOG .......BENUTZER......002..........001.........00001....1......
................................................................................
................................................................................
LEASY CALLS:........00000001......WRITE OPERATIONS ON BIM FILE(S):.00000001.....
..................................WRITE OPERATIONS ON AIM FILE:....00000000.....
..................................AIM RECORDS:.....................00000001.....
OPTR CALLS:.........00000001......CALLS ON LOCK PROTOCOL:..........00000000.....
..................................WAITS IN LOCK PROTOCOL:..........00000000.....
SETL CALLS:.........00000000......REJECTED CALLS ON LOCK PROTOCOL:.00000000.....
READING CALLS:......00000000......LOCKED DATA RECORDS:.............00000000.....
WRITING CALLS:......00000000......DEADLOCK RETURNS:................00000000.....
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*tskt...........................................................................
```

(8) Task table



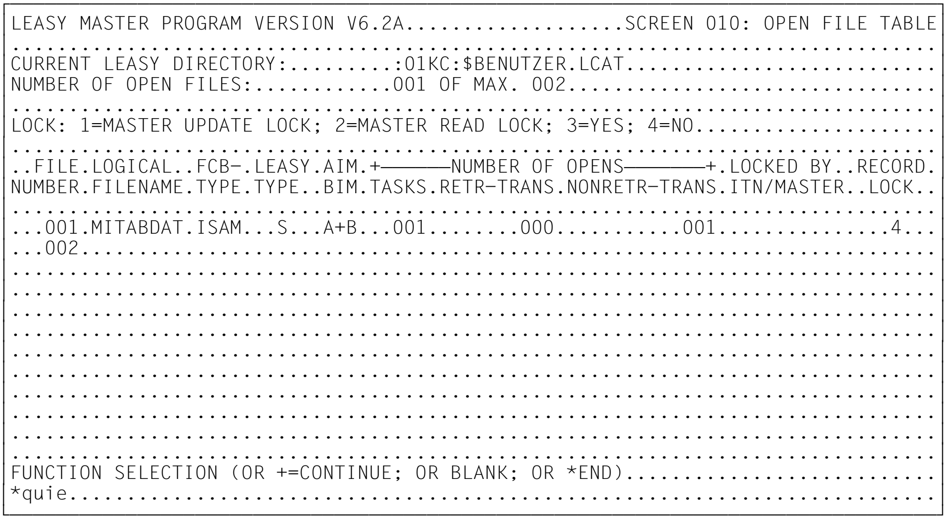
```
LEASY MASTER PROGRAM VERSION V6.2A........................SCREEN 018: TASK TABLE
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
NUMBER OF ACTIVE TASKS:..........001 OF MAX. 003................................
................................................................................
TASK TYPE: 1=INTERACTIVE; 2=BATCH; 3=UTM-B/S; 4=UTM-T; 5=DCAM...................
................................................................................
..TASK.......TASK...............................................................
NUMBER...TSN.TYPE.IDENTIFICATION................................................
...001..2098...1................................................................
...002..........................................................................
...003..........................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*filt...........................................................................
```

(9) Table of files



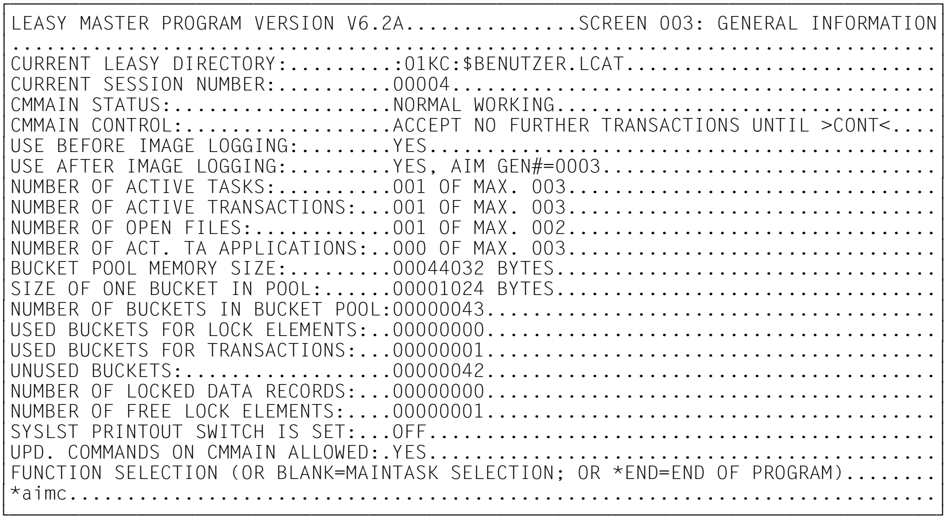
```
LEASY MASTER PROGRAM VERSION V6.2A..................SCREEN 009: TOTAL FILE TABLE
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
................................................................................
LOCK ON FILE: 1=UPDATE LOCK; 2=READ LOCK; 3=NO LOCK.............................
................................................................................
..FILE.LOGICAL..FCB-.LEASY.AIM.NUMBER.LOCK ON.DMS FILENAME (FIRST 35 CHARACTERS)
NUMBER.FILENAME.TYPE.TYPE..BIM..OF SI..FILE...OR MODEL INSTANCE NAME............
................................................................................
...001.MITABDAT.ISAM...S...A+B....002....3....:01KC:$BENUTZER.LCAT.MITABDAT.....
...002.PROTDAT .SAM ...T...A+B....000....3....:01KC:$BENUTZER.LCAT.PROTDAT......
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*opft...........................................................................
```

(10) Table of files opened by transactions



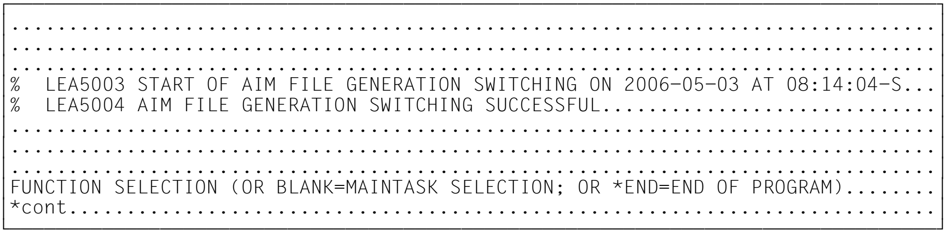
```
LEASY MASTER PROGRAM VERSION V6.2A...................SCREEN 010: OPEN FILE TABLE
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
NUMBER OF OPEN FILES:............001 OF MAX. 002................................
................................................................................
LOCK: 1=MASTER UPDATE LOCK; 2=MASTER READ LOCK; 3=YES; 4=NO.....................
................................................................................
..FILE.LOGICAL..FCB-.LEASY.AIM.+──────NUMBER OF OPENS───────+.LOCKED BY..RECORD.
NUMBER.FILENAME.TYPE.TYPE..BIM.TASKS.RETR-TRANS.NONRETR-TRANS.ITN/MASTER..LOCK..
................................................................................
...001.MITABDAT.ISAM...S...A+B...001........000...........001...............4...
...002..........................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
*quie...........................................................................
```

(11) General information following QUIE



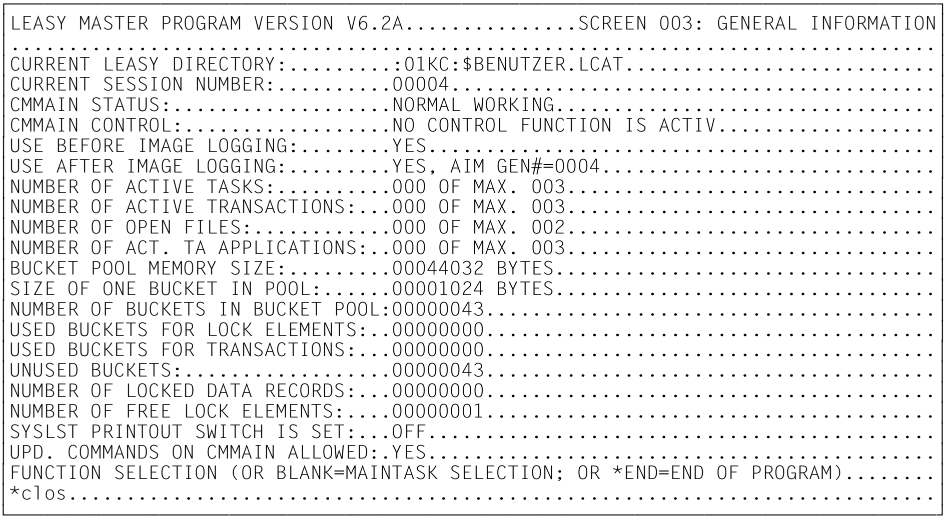
```
LEASY MASTER PROGRAM VERSION V6.2A...............SCREEN 003: GENERAL INFORMATION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
CURRENT SESSION NUMBER:..........00004..........................................
CMMAIN STATUS:...................NORMAL WORKING.................................
CMMAIN CONTROL:..................ACCEPT NO FURTHER TRANSACTIONS UNTIL >CONT<....
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........YES, AIM GEN#=0003.............................
NUMBER OF ACTIVE TASKS:..........001 OF MAX. 003................................
NUMBER OF ACTIVE TRANSACTIONS:...001 OF MAX. 003................................
NUMBER OF OPEN FILES:............001 OF MAX. 002................................
NUMBER OF ACT. TA APPLICATIONS:..000 OF MAX. 003................................
BUCKET POOL MEMORY SIZE:.........00044032 BYTES.................................
SIZE OF ONE BUCKET IN POOL:......00001024 BYTES.................................
NUMBER OF BUCKETS IN BUCKET POOL:00000043.......................................
USED BUCKETS FOR LOCK ELEMENTS:..00000000.......................................
USED BUCKETS FOR TRANSACTIONS:...00000001.......................................
UNUSED BUCKETS:..................00000042.......................................
NUMBER OF LOCKED DATA RECORDS:...00000000.......................................
NUMBER OF FREE LOCK ELEMENTS:....00000001.......................................
SYSLST PRINTOUT SWITCH IS SET:...OFF............................................
UPD. COMMANDS ON CMMAIN ALLOWED:.YES............................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*aimc...........................................................................
```

(12) Switching the AIM file generation



```
................................................................................
................................................................................
................................................................................
%  LEA5003 START OF AIM FILE GENERATION SWITCHING ON 2006-05-03 AT 08:14:04-S...
%  LEA5004 AIM FILE GENERATION SWITCHING SUCCESSFUL.............................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*cont...........................................................................
```

(13) General information following CONT



```
LEASY MASTER PROGRAM VERSION V6.2A...............SCREEN 003: GENERAL INFORMATION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
CURRENT SESSION NUMBER:..........00004..........................................
CMMAIN STATUS:...................NORMAL WORKING.................................
CMMAIN CONTROL:..................NO CONTROL FUNCTION IS ACTIV...................
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........YES, AIM GEN#=0004.............................
NUMBER OF ACTIVE TASKS:..........000 OF MAX. 003................................
NUMBER OF ACTIVE TRANSACTIONS:...000 OF MAX. 003................................
NUMBER OF OPEN FILES:............000 OF MAX. 002................................
NUMBER OF ACT. TA APPLICATIONS:..000 OF MAX. 003................................
BUCKET POOL MEMORY SIZE:.........00044032 BYTES.................................
SIZE OF ONE BUCKET IN POOL:......00001024 BYTES.................................
NUMBER OF BUCKETS IN BUCKET POOL:00000043.......................................
USED BUCKETS FOR LOCK ELEMENTS:..00000000.......................................
USED BUCKETS FOR TRANSACTIONS:...00000000.......................................
UNUSED BUCKETS:..................00000043.......................................
NUMBER OF LOCKED DATA RECORDS:...00000000.......................................
NUMBER OF FREE LOCK ELEMENTS:....00000001.......................................
SYSLST PRINTOUT SWITCH IS SET:...OFF............................................
UPD. COMMANDS ON CMMAIN ALLOWED:.YES............................................
FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)........
*clos...........................................................................
```

(14) General information following CLOS

```
LEASY MASTER PROGRAM VERSION V6.2A...............SCREEN 003: GENERAL INFORMATION
................................................................................
CURRENT LEASY DIRECTORY:.........:01KC:$BENUTZER.LCAT...........................
CURRENT SESSION NUMBER:..........00004..........................................
CMMAIN STATUS:...................NORMAL WORKING.................................
CMMAIN CONTROL:..................TERMINATE MAINTASK.............................
CMMAIN CONTROL:..................ACCEPT NO FURTHER TRANSACTIONS.................
USE BEFORE IMAGE LOGGING:........YES............................................
USE AFTER IMAGE LOGGING:.........YES, AIM GEN#=0004.............................
NUMBER OF ACTIVE TASKS:..........000 OF MAX. 003................................
NUMBER OF ACTIVE TRANSACTIONS:...000 OF MAX. 003................................
NUMBER OF OPEN FILES:............000 OF MAX. 002................................
NUMBER OF ACT. TA APPLICATIONS:..000 OF MAX. 003................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
................................................................................
FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)...........................
* ..............................................................................
```

(15) Common memory selection

```
LEASY MASTER PROGRAM VERSION V6.2A................SCREEN 001: MAINTASK SELECTION
................................................................................
PLEASE TYPE IN NAME OF LEASY DIRECTORY..........................................
(*END=END OF PROGRAM)...........................................................
**end...........................................................................
```

(16) Termination of the utility routine

```
 % LEA0510 NORMAL TERMINATION OF LEASY MASTER PROGRAM
```

Explanation of the example

(1) After calling the utility routine, the user is requested to enter the name of the LEASY catalog and the catalog password.

(2) The general information is then output.

(3) The *HELP* function effects the output of all functions of the utility routine on 3 screen pages.
Scrolling to the next page is effected by entering "+".

(4) Output of the general counters of the current session using the *GENC* function after the start of an application program.

(5) *TRAT* function. The transaction table is displayed.

The transaction takes place in timesharing mode under the TSN 2098 and the ID *BENUTZER* (user ID). Two file IDs have been opened. The number of the transaction is 1; it is in the active state.

(6) With *TRAC*, users are requested to specify the transaction for which they want to output counters. The number or the identifier of the transaction must be specified, as listed in the preceding table (*TRAT* function).

(7) The counters of the specified transaction (1) are output.

(8) The task table indicating the number (out of the possible maximum) of active tasks identified by TSN and task type is output (*TSKT* function).

(9) Information concerning all files in the current LEASY catalog is displayed (*FILT* function).

(10) The names of and the information concerning open files are output (*OPFT* function).

(11) The general information mask is displayed after entering the *QUIE* function (blocking of new transactions. The *QUIE* function is indicated in the *CMMAIN CONTROL* line.

(12) *AIMC* can be input following *QUIE*. Switching of the AIM file generation is acknowledged.

(13) The *CONT* function cancels suspension initiated by the *QUIE* function. This is shown in *CMMAIN CONTROL* in the information mask. The AIM file generation has been switched (*AIM-GEN#=4*).

(14) The *CLOS* function terminates MAINTASK

(*CMMAIN CONTROL: TERMINATE MAINTASK*)

(15) Entering a blank terminates the activities for the selected common memory. They can be selected again for as long as the common memory exists. Entering **END* at this point terminates the LEASY-MASTER utility without returning to the main task selection.

(16) LEASY-MASTER is terminated.

9 LEASY-RECONST

The LEASY-RECONST utility routine enables destroyed LEASY files to be reconstructed with the aid of save files and the corresponding after-image file. In addition, it permits shadow files to be updated in parallel with an ongoing LEASY session using an already completed AIM file generation.

Additional functions enable a listing of the actions performed with LEASY to be produced from the AIM file.

Application program runs can be traced by extending the listing to include the contents of all records in the AIM file.

The start of processing of each session is logged to SYSOUT.

Other statements can be used to restrict processing to a certain section of the AIM file, thus providing selective use of the reconstruction program.

9.1 User-defined RECONST ENTER file

If a *RENTER* file does not already exist, one is created at the beginning of a session. This gives users the opportunity to adapt the *RENTER* file to their own requirements. However, users must ensure that the *RENTER* file specified in LEASY-MAINTASK (*REN* operand) contains control statements which are really valid.

Structure of the RECONST ENTER file

The ENTER file must at least contain the following control statements:

```
/SET-LOGON-PAR
 .
 .                                               (1)
 .
/START-LEASY-RECONST
*CAT filecatalog,COP=(Y,A)
 .                                               (2)
 .
*END
 .
 .                                               (1)
 .
/EXIT-JOB
```

(1) Other BS2000 commands can be inserted here, e.g. to change the assignment to SYSOUT etc.

(2) In addition to the mandatory control statements **CAT* and **END*, other control statements can be specified (see table below)

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

9.2 Input to LEASY-RECONST: special features

The LEASY-RECONST utility routine can run in interactive or batch mode. Except where otherwise specified, the following information applies to both modes.

- The **CAT* statement must precede the **FIL* statement. Inputs to the LEASY-RECONST utility routine must be concluded with an **END* statement.
- All other statements may be input in any order.
- The **CAT* and **END* statements are mandatory; all other statements have default values and are optional.
- All statements specified, except **COM* and **END*, must contain at least one operand.
- In contrast to the LEASY-CATALOG utillity routine, LEASY-RECONST does not permit repetition of correctly input statements for reasons of data security. However, statements incorrectly input during the dialog may be corrected.
- Input record chaining is permitted for all statements.

9.3 LEASY-RECONST statements

The statements are described below in alphabetical order.

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 6: Overview of statements (LEASY-RECONST)

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

[2] Mandatory; must be the last statement specified

*CAT Access LEASY catalog

The **CAT* statement selects the LEASY catalog, the AIM file generation(s) to be used and the common memory CMMAIN of the main LEASY task.

Operation	Operands

`[*]CAT[ALOG]`	`[:catid:][$userid.]file-catalog`
	`[,GEN[]=[-]generation]`
	`[,FRO[]=[-]generation1]`
	`[,TOG[]=[-]generation2]`
	`[,COP[]={Y[] / (Y[],A[]) / N[]}]`

catid — Specifies the catalog identifier of the public volume set which contains the LEASY catalog. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-RECONST was started. *catid* may comprise up to 4 characters, enclosed in colons.

userid — Specifies the user ID under which the LEASY catalog is entered in the public volume set with the catalog identifier *catid*. If *userid* is omitted, LEASY uses the user ID under which LEASY-RECONST was started.

file-catalog — Specifies the name of the LEASY catalog.

GEN[ERATION]=[−]generation

Specifies which generation of the AIM file is to be used.

generation	absolute generation number
-generation	relative generation number

The permitted values are numeric, and within the range:

1 ≤ *generation* ≤ 9999.

If a minus sign (-) is specified, *generation* is the relative generation number, relative to the current generation.

If this operand is omitted, the highest existing generation is used for the program run when *TRANS=A* is specified for **MOD*.

With **MOD TRANS=C* or *TRANS=V*, a search is carried out in descending order from the highest existing generation to the one containing the start of session.

All specifications of this statement are used to form the file name of the AIM file. The file is then named:

```
$userid.file-catalog.LEASYAIM(*generation)
```

Specification of *GEN* is not permitted in conjunction with *FRO* and *TOG*.

FRO[MGEN]=[]generation1

AIM files are processed from the specified generation to the highest existing generation.

TOG[EN]=[]generation2

AIM files are processed beginning from the lowest generation to the one specified (absolute or relative). If no value is specified for *TOG*, the current AIM *generation1* is used for reconstruction.

i If *FRO* and *TOG* are specified, all files are processed from *generation1* to *generation2* (both may be specified as absolute or relative generations).
generation2 < *generation1* is not permitted.

AIM files may be processed from the first file up to the number allowed in the command

```
/CREATE-FILE-GROUP GROUP-NAME=filename,GENERATION-PAR=(MAXIMUM=max)
```

for the file generation group or in the **AGE* statement of LEASY-MAINTASK.

COP[Y] specifies whether the original files or copy files (shadow files) are to be reconstructed.

=Y[ES] The shadow files are reconstructed.

During reconstruction the names of the copy files are formed from the DMS file names of the original files according to the *CPC/CPS* rules (**CAT* statement, LEASY-CATALOG); these copy files are then opened.

During a reconstruction run to copy files the relevant common memory CMMAIN must be available, i.e. the LEASY MAINTASK must have been started. The CMMAIN can either be in the state for AIM reconstruction (LEASY-MAINTASK, **USE=R*) or be released for the LEASY runtime system (LEASY-MAINTASK, **USE=N*).

This CMMAIN is only read by the reconstruction program and not modified. It can also be used simultaneously by LEASY user programs, i.e. reconstruction to the shadow files is also possible during the current session. This is necessary for round-the-clock operation.

The *COP=Y* operand is ineffective if the *UPD=N* operand is specified in the *MOD* statement because only the AIM file contents are printed out.

=(Y[ES],A[UTOMATIC])

The shadow files for which a corresponding definition was made using the LEASY-CATALOG utility routine are automatically reconstructed.
Filter specifications are prohibited. File selection is automatically performed by LEASY-RECONST. *MOD* statements are modified if necessary.

ℹ LEASY-RECONST is started by LEASY-MAINTASK using *COP=(Y,A)* if automatic reconstruction of shadow files has been specified. The user can specify *COP=(Y,A)* only if the RECONST task, started by MAINTASK, has been aborted on account of system or DMS errors. In this case the user can restart the RECONST task.

With *COP=(Y,A)*, the *GEN*, *FRO* and *TOG* operands are checked for correct syntax but are not modified.

=<u>N[O]</u> The original files are reconstructed.

The common memory CMMAIN must have been generated in the state for AIM reconstruction (**USE=R*, LEASY-MAINTASK).

*COM Comment text

The _*COM_ statement inserts comment lines at any point in the statement sequence.

It is used primarily if the utility routine is employed in procedures.

Operation	Operands
`[*]COM[MENT]`	`[text]`

text Any text

*DAT Date filter

The _*DAT_ statement limits reconstruction to the period between two dates.

Operation	Operands

`[*]DAT[E]`	`FRO[]=date1[,TOD[]=date2]`
	`TOD[]=date2`

FRO[MDATE]=date1

Starting date from which the AIM file is to be processed.

The date must be entered in the format _yyyy-mm-dd_ (_yyyy_=year, _mm_=month, _dd_=day). The format _yymmdd_ is supported for reasons of compatibility: internally it is replaced by 20_yy-mm-dd_ (if _yy_ is less than 60) or 19_yy-mm-dd_ (if _yy_ ≥ 60). If _FRO_ is not specified, the AIM file is processed from the beginning, unless there is a further restriction (see the _*SES_ statement).

TOD[ATE]=date2

Specifies the end date up to which the AIM file is to be processed

The date must be entered in the format _yyyy-mm-dd_ (_yyyy_=year, _mm_=month, _dd_=day). The format _yymmdd_ is supported for reasons of compatibility: internally it is replaced by 20_yy-mm-dd_ (if _yy_ is less than 60) or 19_yy_-_mm_-dd (if _yy_ ≥ 60). If _FRO_ is not specified, the AIM file is processed from the beginning, unless there is a further restriction (see the _*SES_ statement).

Only AIM record types _MTSK_ and _SESS_ are used in the selection of the dates, i.e. the records are processed for all sessions whose session record falls between the two dates.

*END Program termination

The **END* statement terminates the input for the LEASY-RECONST utility routine and starts the reconstruction run.

Operation	Operands
`[*]END[]`	

*FIL File selection

The **FIL* statement selects the files to be reconstructed.

Operation	Operands

`[*]FIL[ELIST]`	`A[]`
	`[-](dvsname[,...])`

A[LL]

All the files of the selected LEASY catalog are to be reconstructed.

[-](dmsname)

This operand enables files to be selected; here *dmsname* is the DMS name of the file in the form *[:catid:][$userid.]filename*. If a minus sign is entered before the file name(s), all other files in the LEASY catalog are reconstructed except the one or ones specified.
If the minus sign is omitted, only the specified files are reconstructed.

If the files are distributed over various public volume sets, the catalog identifier (*:catid:*) of the public volume set must be specified. *catid* may comprise up to 4 characters, enclosed in colons.

Each file name can be partially qualified, and must include the user ID if the file is created under a user ID other than that of the assigned LEASY catalog. Parentheses are also necessary where only one file is specified.

The names of the original files must be specified even if the copy files are to be reconstructed (**CAT* statement, *COP=YES*). The copy names are not formed until after file selection. If, for example, only all foreign files of a particular user ID are to be reconstructed, these are selected by entering **FIL ($userid.)*

*MOD Function selection

The _*MOD_ statement controls the sequence of operations of the LEASY-RECONST utility routine.

Operation	Operands
[*]MOD[E]	⎧N[ ]⎫ [PRI[]=⎨ ⎬] ⎩M[ ]⎭ ⎧Y[ ]⎫ [,UPD[]=⎨ ⎬] ⎩N[ ]⎭ ⎧Y[ ]⎫ [,SIU[]=⎨ ⎬] ⎩N[ ]⎭ ⎧Y[ ]⎫ [,UNL[]=⎨ ⎬] ⎩N[ ]⎭ ⎧A[ ]⎫ [,TRA[]=⎨C[ ]⎬] ⎩V[ ]⎭ ⎧Y[ ]⎫ [,FRE[]=⎨ ⎬] ⎩N[ ]⎭

PRI[NT] This operand specifies the extent of the listing.

=N[ORMAL] The routine compiles a listing of the records in the AIM file. Editing of the listing is controlled by the _*REP_ statement or its default values.

=M[INIMUM] The routine creates a mini-listing containing only SESSION records. The _*REP_ statement is rejected.

UPD[ATE] This operand enables the user to either implement or suppress reconstruction of LEASY files.

=Y[ES] The LEASY files are to be reconstructed.

=N[O] Reconstruction of LEASY files is suppressed. This entry is useful, for example, when only one listing of the AIM file is required (see operand _PRI=N_).

Entering _*MOD_...,_UPD=N_ and LEASY-MAINTASK with _USE=R_ is not permitted.

SIU[PDATE]
This operand can specify, for files with secondary indices defined in the LEASY catalog, whether the corresponding secondary index pointers are to be reconstructed at the same time as the primary file.

=Y[ES]
The SI files are to be reconstructed in the same run (default value if *UPD=Y*).

SIU=Y is a global specification. Whether the secondary index pointers for a file are in fact created depends on the entries of each secondary index in the LEASY catalog. The specification is permitted only in conjunction with *UPD=Y*.

=N[O]
The SI files are not to be reconstructed (default value if *UPD=N*). However, following reconstruction they can be recreated by means of LEASY-LOADSI.

UNL[OAD]
This operand serves to control storage and runtime. It is only effective in conjunction with *UPD=Y*.

=Y[ES]
When a SESSION entry occurs in the AIM file all LEASY files open at that time are closed, thus releasing the class 5 memory of the IOAREAs (see the "Introductory Guide to DMS" manual).

Specification of *UNL=Y* is recommended if different sessions repeatedly process different files. This helps avoid unnecessary inflation of the class 5 memory.

=N[O]
The LEASY files are only closed if they are to be opened in a different *OPEN* mode when an *OPEN* record is encountered.

UNL=N should be entered if most sessions process the same files. By thus avoiding unnecessary *OPEN-CLOSE* cycles, runtime can be reduced during reconstruction.

TRA[NSACTIONS]
This operand can specify whether the AIM file is to be processed normally, or whether the AIM elements of uncompleted transactions in the last session are to be declared invalid, or whether only records of completed transactions are to be processed.

=A[LL]
The valid AIM elements of all transactions are to be processed.

=C[LOSED]
TRA=C does not effect any reconstruction. All transactions not closed by *CLTR* calls in the last recorded session are recognized and the AIM elements therein are declared invalid. A log of the last session of the AIM file is then created, indicating, for example, that this operation has taken place (by means of "*" after the AIM record type).

This procedure can only be carried out with AIM files on disk.

If this operand is used, LEASY-RECONST checks or sets the following operand values:

```
*MOD PRI=N,UPD=N,SIU=N
*REP LIS=Y,REC=A
```

The statements **SES*, **DAT*, **FIL* and **RAN* are not permitted.

The operand *COP=YES* in the **CAT* statement becomes meaningless. If a file generation range has been specified, it must contain the last generation.

See page 249ff for special notes on the execution of LEASY-RECONST when *TRA=C* is specified.

=V[ALIDATED] All transactions not closed by *CLTR* calls in the last recorded session are recognized and listed in an internal table. If a session entry is found, the AIM records of all completed transactions are processed. If no session entry is found, only a list is created. In the REPORT list, AIM elements are marked by an asterisk and can therefore be distinguished from open transactions.

TRA=V has the same effect as *TRA=C* followed by *TRA=A* if a session entry has been found.

The following operand values are checked or set when this operand is used:

```
*MOD PRI=N
*REP LIS=Y,REC=A
```

The *COP=Y* operand in the **CAT* statement has no effect. If a file generation range was specified, it must contain the last generation.

See page 249ff for special notes on the execution of LEASY-RECONST when *TRA=V* is specified.

FRE[E] This operand controls further use of the common memory CMMAIN in the main LEASY task.

=<u>Y[ES]</u> The common memory CMMAIN is to be released for application programs after the reconstruction run. This only applies if the reconstruction run is error-free (**MOD..., UPD=Y*).

=N[O] Further reconstruction runs are planned after the LEASY-RECONST run. The common memory CMMAIN is to retain its present status and remain locked to user programs.

i Program runs with **MOD..., UPD=N* or with **MOD..., TRA=C* do not require a common memory CMMAIN, i.e. the LEASY MAINTASK need not be started. If the common memory already exists, its status is retained.

*RAN Control range to be listed

The **RAN* statement restricts the range of the AIM file which is to be listed. The statement is only accepted if *UPD=N* was selected in the **MOD* statement.

Operation	Operands

`[*]RAN[GE]`	`[FRO[]=time1][,TOT[]=time2]`
	`[,FIR[]=tsn1][,LAS[]=tsn2]`

FRO[MTIME]=time1
Specifies the time in the form *hhmm* (*hh*=hours, *mm*=minutes) when the listing of the AIM file is to be started.
Permissible values: 0000 ≤ *time1* < 2400.
If this operand is omitted, the AIM file is listed from 00:00 onwards, provided no other restrictions apply.

TOT[IME]=time2
Specifies the time in the form *hhmm* (*hh*=hours, *mm*=minutes) when the listing of the AIM file is to be stopped.
Permissible values: 0000 < *time2* ≤ 2400.
If this operand is omitted, the AIM file is listed until 24:00, provided no other restrictions apply.

FIR[STTSN]=tsn1
Specifies the TSN as of which the AIM file is to be listed.

Permissible values: AAAA ≤ *tsn1* ≤ 9999.

If this operand is omitted, the AIM file is listed from *tsn1=AAAA*, provided no other restrictions apply. If fewer than four characters are entered, the TSN inserts as many leading zeros as are required to make up the difference. The TSNs are arranged according to the EBCDI code, i.e. the sequence is as follows:
A,B,C,....,X,Y,Z,0,1,2,....,7,8,9

LAS[TTSN]=tsn2
Specifies the TSN up to which the AIM file is to be listed.

Permissible values: AAAA ≤ *tsn2* ≤ 9999.

If this operand is omitted, the AIM file is listed up to *tsn2=9999*, provided no other restrictions apply. If fewer than four characters are entered, the TSN inserts as many leading zeros as are required to make up the difference.

RAN applies to all AIM records (unlike the filter statements *DAT* and *SES*). The intersection of all filter statements is output.

*REP Control of listing

The **REP* statement controls the extent of the listing (SYSLST). The statement is rejected if *PRI=M* was specified in the **MOD* statement.

Operation	Operands
[*]REP[ORT]	[LEN[]={S[ ] M[ ] L[ ] A[ ][,EXT[]=(pos,len)]...}] [,LIS[]={Y[ ] N[ ]}] [,REC[]={A[ ] M[ ] [-](sa[,...])}] [,USE[RINFORMATION]={N[ ] Y[ ]}] [,PRO[]={N[ ] Y[ ]}]

LEN[GTH] This operand specifies the printout format.

=S[HORT] The program creates one print line for each AIM record (see "Printout of the reconstruction log" on page 260).

A print line of a reconstruction log contains the following information:

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

Field	Meaning	
TRANS	Transaction number within the session	
ITR	Internal LEASY transaction number	
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 TSN U User ID P User program name I "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 openUTM user ID for openUTM programs; user ID for DCAM applications, otherwise left blank. H Processor name for openUTM programs, "DCA-tsno" for DCAM applications. A "$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.

=M[EDIUM]

For the record types *OPEN* and *FILS*, a second line appears which contains the logical file names of the open files (max. 14 files). For the record types *PUTX*, *PUTS* and *STOR*, a second line containing the record contents from character 1 to 124 appears, provided the record is that long. All other record types have the same output as for the *S* option.

=L[ARGE]

This entry differs from the *M* entry only for the record types *OPEN* and *FILS*; each file entered has a separate print line containing:

```
FILE=filename
OPENMODE=openmode
```

If the file could not be accessed, the following also appears in the same line:

```
FILE ACCESS IN ERROR
```

=A[LL]

This specification is only valid for record types *STOR*, *DLET*, *SETS*, *STOD*, *PUTX*, *PUTS*, *ELIR* and *ELIF*. All other records are treated as if *L* were specified. Two variants are available:

– If the additional operand *EXT* is not specified, the complete record contents are presented continuously (always beginning at print position 7).

– If *EXT* is specified, the output is controlled by this operand.

EXT[RACT]

This operand is only valid in conjunction with the operand *LEN=A*. It specifies the position and length of the record sections to be printed out. It must not be entered more than 255 times, i.e. up to 255 record extracts are permitted.

pos

is the relative record position at which output of the record contents is to begin (with variable-length records it includes the record length field).
Values must be numeric within the range:
1 ≤ *pos* ≤ 32767.

If the specified record position is not within the data record, the following message is displayed:

```
RECORD IS TOO SHORT FOR THIS POSITION
```

len

is the length of the record contents to be printed out, beginning at record position *pos*. It must be a numeric value in the range:
1 ≤ *len* ≤ 105.

If specified, the *EXT* operand applies to all the files in a program run. Selective application is only possible in conjunction with the *FIL statement if only files with the same record format are used in each program run.

LIS[TINVALID]

Controls output of invalid *AIM records (see the *MOD statement, operand *TRA=C* or *TRA=V*).

=<u>Y[ES]</u>

Invalid records are to be printed out and flagged as such.

=N[O]

Printout of invalid records is suppressed.

REC[ORD] This operand allows selection of the record types to be printed out (it does not affect reconstruction).

=A[LL] All record types are to be listed.

=M[INIMUM] Only record types *MTSK*, *SESS*, *CATD*, *RLBK*, *OLDB*, *ENDA*, *CTSK* and *CSES* are listed.

=(sa[,...]) Only the specified record types *sa* and *MTSK*, *SESS*, *CATD*, *RLBK*, *OLDB*, *ENDA*, *CTSK* and *CSES* are listed.

=-(sa[,...]) The specified *sa* record types are not listed.

The following keywords are available for selection of the record types (the user must decide upon a practicable combination):

`OPEN[ ], CLOS[ ], OPTR[ ], CLTR[ ], STOR[ ], DLET[ ], STOD[ ], SETS[ ], PUTX[ ], PUTS[ ], ELIR[ ] and ELIF[ ].`

USE[RINFORMATION]
The listing of the USER text from the AIM file can be controlled by means of this operand.

=N[O] No *USER* information is output in the RECONST log.

=Y[ES] The *USER* information is output in the RECONST log as a record with a maximum of 1020 characters.

PRO[TOCOL] Controls the scope of AIM element logging.

Applies only if *UPD=N* is specified (**MOD* statement).
If *UPD=Y* is specified, *PRO=N* is set.

= N[O] Only update operations are logged.

= Y[ES] Read operations, the *LOCK* and *UNLK* operations, and the *CINF* information function are logged in addition to update operations.

Read accesses are logged without user information.

Read operations, the *LOCK/UNLK* operations and *CINF* can be logged only if the AIM elements are contained in the AIM file (**FIL ... AIM = Y|R* in LEASY-CATALOG for the relevant file and **LOG = Y|P* or **LOG = A|P* in LEASY-MAINTASK).

*SES Session filter

The **SES* statement restricts reconstruction to a specified range of session numbers.

Operation	Operands

`[*]SES[SION]`	`FRO[]=sessno1[,TOS[]=sessno2[,LAS[]=transno]]`
	`TOS[]=sessno2[,LAS[]=transno]`

FRO[MSESSION]=sessno1

specifies the session number at which processing of the AIM file is to begin.

The permitted range of values is:
0 ≤ *sessno1* ≤ 65535.

If this operand is omitted, the AIM file is processed from the start of the file, provided no other restrictions apply (see the **DAT* statement).

TOS[ESSION]=sessno2

specifies the session number up to which the AIM file is to be processed.

The permitted range of values is:
1 ≤ *sessno2* ≤ 65535.

If this operand is omitted, the AIM file is processed up to the end of the file, provided no other restrictions apply (see the **DAT* statement).

Only the records of those sessions whose session record lies within the session-number range are processed in session selection.

LAS[TTRANS]=transno

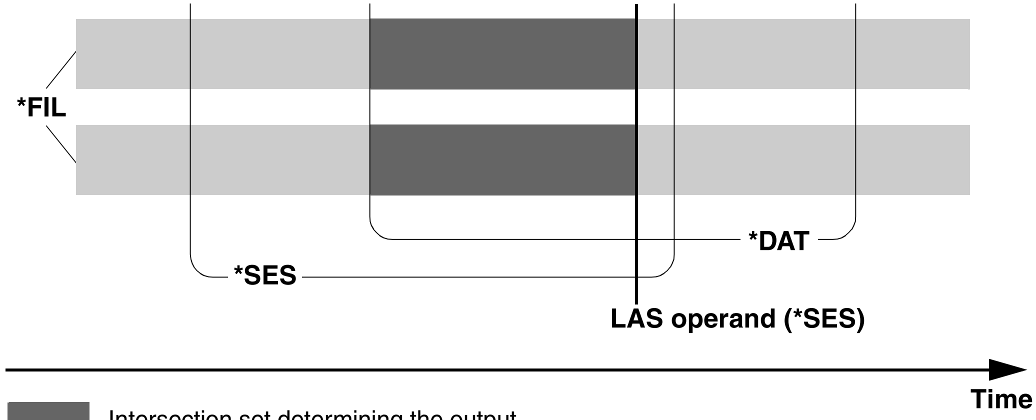
specifies a transaction number within the session specified in *TOS*, up to which the AIM file is to be processed.

The permitted range of values is:
1 ≤ *transno* ≤ 65535.

If *LAS* is used, the *TOS* operand is mandatory.

9.4 Filter effect

The **FIL*, **DAT* and **SES* statements have the effect of a multilayered filter on processing of the AIM file. The output generated (both the file reconstruction and the printout) is derived from the intersection set of these three statements:



▆	Intersection set determining the output
*FIL	File selection
*DAT	Date filter
*SES	Session filter
LAS	Specification of the last transaction to be processec

Figure 1: Filter effect

Only complete transactions are processed, with one exception: namely, when incomplete transactions are contained in the last recorded session of the AIM file but *ALL* is specified in the *TRANS* operand of the **MOD* statement.

9.5 Log output

Output to SYSOUT

Each session record processed in the reconstruction run is logged to SYSOUT. The output line contains the:

- session number
- TSN
- date
- time.

Output to SYSLST

A reconstruction log is output for each reconstruction run. This comprises two parts:

- The first part indicates the functions of the LEASY-RECONST utility routine selected for the reconstruction run (cover page).
- The second part can be determined by the user as explained in the description of the **REP* statement.

As a rule the contents of records are output without modification, i.e. without editing of non-readable data or formatting beyond that provided by the *EXT* operand. For this reason binary data generates its equivalent printable characters in accordance with the EBCDIC code.

When **truncated** AIM elements are used, the contents of the records are output in non-truncated form, provided the file reconstruction can be carried out without errors. However, if an error occurs, the record is listed in the form in which it is stored in the AIM file. Any *EXTRACT* output has no effect in this case.

Depending on the selected operand combination, six different types of listing can be output. Although differing greatly in size, they have a common basic format:

Operand PRI (*MOD)	**Operand LEN (*REP)**	**Operand EXT (*REP)**	**Output**
M	not specified	not specified	only *SESS* records
N	S	not specified	one line per record type
N	M	not specified	two-line printout for *OPEN, PUTX, PUTS, STOR* and *FILS;* single-line output for other record types
N	L	not specified	one line per file entry in the *OPEN* record; otherwise as for the combination *PRI=N, LEN=M*
N	A	not specified	the complete record contents for record types *STOR, PUTX, PUTS* and *STOD;* otherwise as for the combination *PRI=N, LEN=L*
N	A	specified	control of output via the *EXT* operand for record types *STOR, PUTX, PUTS* and *STOD;* otherwise as for the combination *PRI=N, LEN=L*

9.6 Reconstruction run

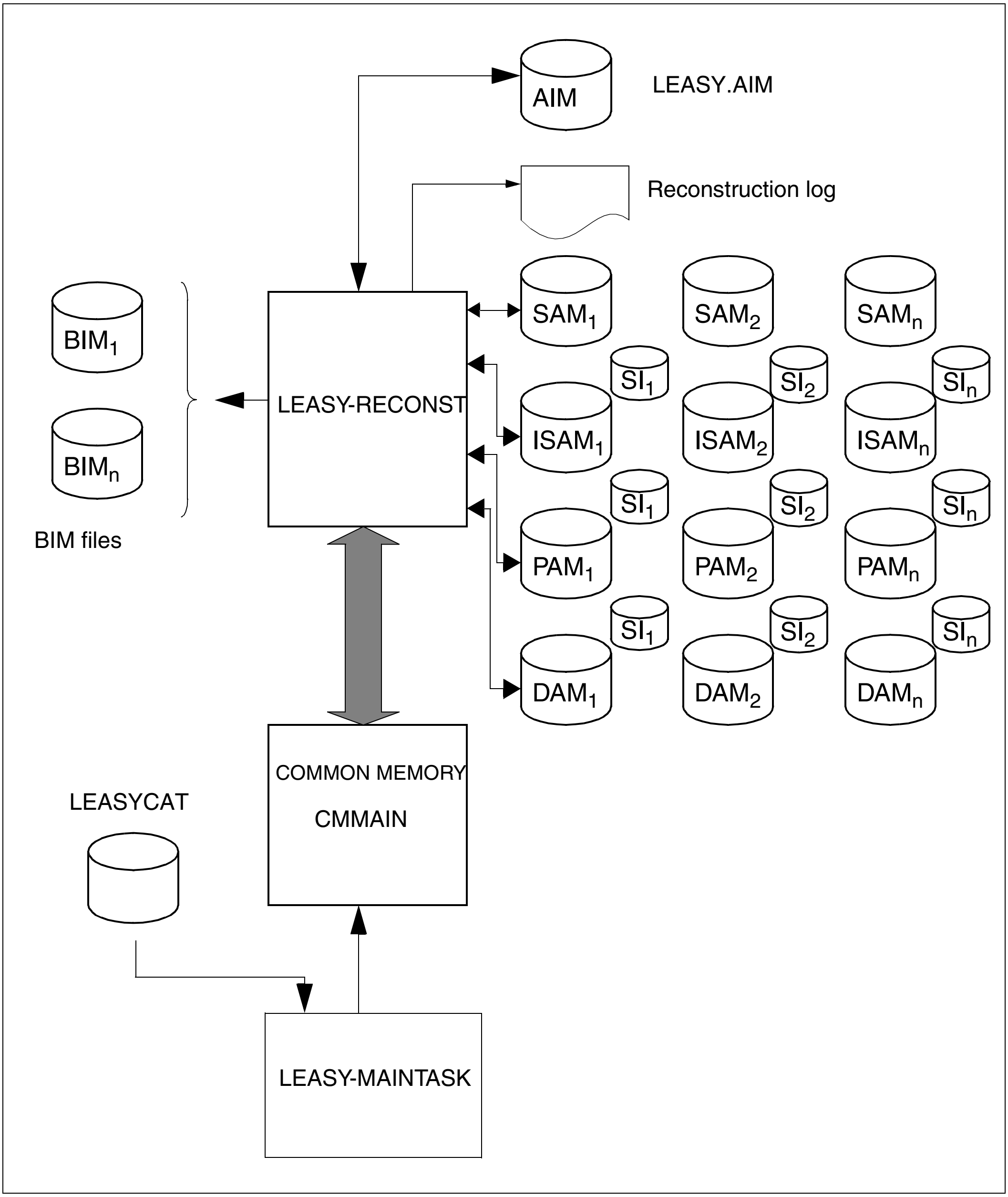


Figure 2: System environment of the LEASY-RECONST utility routine

General sequence of operations for reconstruction of original files

Before reconstructing destroyed files, the cause of the error (e.g. a defective disk storage pack), must first be located and eliminated. The original (error-free) status of the LEASY files must then be reconstructed using the save files. These save files can be created, for example, using the ARCHIVE utility routine (see the "ARCHIVE (BS2000/OSD)" manual) or by means of the *COPY-FILE* command (see the "Commands, Volume 1 - 5" manual).

The main LEASY task should then be started using the LEASY.MAINTASK utility routine and the statement **USE=R*. This statement readies the common memory CMMAIN created for the LEASY-RECONST run, while inhibiting concurrent LEASY application runs. The LEASY-RECONST utility routine run then reconstructs all updates made to the LEASY files up to a specified point in time.

Care must be taken to avoid errors in the reconstruction run. If errors do occur, the reconstruction run can be repeated as often as required. Consistency of LEASY files can only be guaranteed if the reconstruction run is error-free.
If reconstruction was error-free, the common memory CMMAIN is released for LEASY application runs; however, *FRE=Y* must have been specified in the **MOD* operand, otherwise the file status required for further reconstruction runs is retained.

Program runs with the statements **MOD...,UPD=N* or **MOD..., TRA=C* do **not** require the LEASY MAINTASK.

Special cases in a reconstruction run

Using the *TRA* operand in the **MOD* statement

- During a normal run the BIM files ensure that the transactions are processed as a unit, i.e. that a transaction is either executed completely or is rolled back.
- Incomplete transactions (e.g. due to a system crash) are rolled back (by LEASY-MAINTASK) at the latest when warm-starting a new LEASY session.
- However, situations may occur in which this does not apply (e.g. if a DMS error occurs while processing the associated BIM file during a transaction rollback). LEASY files may then contain partially completed transactions. In such cases the LEASY-RECONST utility routine provides additional security.
- If a LEASY application has this status, a LEASY-RECONST run is required initially with the following statements:

  ```
  *CAT      file-catalog,GEN=generation
  *MOD      TRA=C
  *END
  ```

- If the highest existing generation of the AIM file is to be used for reconstruction, the *GEN* operand can be omitted.

- The **REP* statement with the *LEN* and *EXT* operands may also be used.
- This LEASY-RECONST run processes the whole AIM file from the end to the beginning, and declares invalid all the AIM elements of incomplete transactions (indicated by missing *CLTR* records) in the last session. All BIM files are then deleted, since they can no longer be processed. In the same run the AIM file is then read from beginning to end and a listing of the last session is generated. Invalid records are indicated in this listing by an asterisk (*) after the AIM record type.
- After the save files have been read in, the user must ensure in a subsequent reconstruction run (with the operand *TRA=A*) that at least all those files (and their associated SI files) that are marked in the listing described above are processed.
- In this way partially completed transactions in the last session are purged from the files, thus establishing a consistent state again.
- AIM files on tape cannot be processed with **MOD TRA=C*.
- The *TRA* operand in the **MOD* statement can have the value *V[ALIDATED]* in the LEASY-RECONST utility routine. If a generation range is specified (**CAT FRO=...,TOG=...*), it must contain the last generation.
- In this mode the AIM file is read from back to front until a session entry is found. All uncompleted transactions are listed in a table. The BIM files are then deleted since they can no longer be used. The AIM file (or AIM files - if more than one has been specified in the **CAT* statement) is then processed from the beginning, and the files are updated. Records with a transaction identification (which is contained in the table of uncompleted transactions) are not incorporated. These records are marked in the REPORT list by an asterisk (*) after the AIM record type.
- If no session element is found during reverse reading, message *LEA5404* is issued. The BIM files are not deleted and the files are not updated. Only a list of the processed AIM file(s) is created.

Reconstruction using several AIM files

- If the current save files of the LEASY files are lost, reconstruction is still possible so long as all old AIM files still exist in their entirety, beginning with the AIM file belonging to an old save file. These AIM files must be entered in the file in the correct order.
- If several program runs are executed by LEASY-RECONST, the operand *FRE=N* must be specified in the **MOD* statement to ensure that the common memory CMMAIN is not released for user programs after every reconstruction (necessitating deletion and a restart) and that the status is retained for reconstruction. The use of the operands *FRO* and *TOG* in the **CAT* statement is more practical.

Reconstruction using save files and incompatible AIM files

- AIM files older than the save files employed can be used if they meet the following two requirements:
  - This type of AIM file may be used if it is processed at the correct session number indicated by the session filter (**SES* statement), i.e. at the next-highest session number after the save file is created.
  - Although processing of files older than the save file without a session filter is pointless, this does not cause errors, since the records are either overwritten identically or are not located during the deletion attempt (both cases are ignored).
- The use of AIM files more recent than the copies would lead to inconsistent files because of the gap caused by the transactions which are not reconstructed.

Handling the LEASY catalog, state of the secondary index files

Since a common memory CMMAIN (containing mainly the contents of the LEASY catalog) set up by the main LEASY task must exist for the reconstruction run, particular attention should be given to the status of the LEASY catalog.

If the catalog current at the time of the error has not been destroyed, it should be used for reconstruction. Its use is non-critical, even if files or secondary index entries still effective for a section of the AIM file have been deleted from the catalog. LEASY-RECONST intercepts (and logs) such cases, but following this type of run the common memory is not released for further use by application programs.

However, if the LEASY catalog file is also destroyed and only a save file is available, all updates up to the time of the error must be entered in the LEASY catalog prior to reconstruction. Only then is a successful reconstruction possible.
The response of LEASY-RECONST is as described above.

In both cases it may happen that the status of the secondary index files is unsatisfactory because LEASY-LOADSI actions have not been incorporated in AIM saving (e.g. a secondary index of a file may have been deleted during the life of an AIM file, though its pointers still exist following reconstruction). The LEASY-LOADSI utility routine must be used to make these changes after the reconstruction run.

Suggestions for the chronological sequence of file saving and creation of new AIM files

It is always left to the discretion of the user to select the most suitable time for creating save files and opening a new AIM file, since to a large degree this depends on the user applications. However, saving should as a rule be performed periodically, e.g. at daily or weekly intervals.

Whenever possible, file saving (including the LEASY catalog) and creation of a new AIM file should be performed simultaneously.

Reconstruction of shadow files

Not only the original files but also the shadow files can be reconstructed by means of the LEASY-RECONST utility routine. This may be effected either parallel to a reconstruction run (LEASY-MAINTASK,**USE=R*) or parallel to an ongoing LEASY session (LEASY-MAINTASK,**USE=N*). The following applies in both cases:

- The operand *COPY=YES* must be specified in the **CAT* statement of the LEASY-RECONST utility routine.
- The names of the original files are used in the **FIL* statement (LEASY-RECONST) and not those of the shadow files to be reconstructed.
- The name structure for the shadow files is determined by way of the *CPC* and/or the *CPS* operands of the *CAT* statement in LEASY-CATALOG.

In the case of reconstruction parallel to the ongoing session:

- Switching from one generation to the next is effected by means of:
  - *AIMI*, *AIMC* or *AIMW* of the LEASY-MASTER utility routine or
  - **AIS* of the LEASY-MAINTASK utility routine, which determines the maximum size of the AIM file for switching.
- The common memory remains in the state **USE=N*.

Reconstruction of shadow files in the case of a DMS error

After the conclusion of an AIM file generation, in automatic reconstruction the LEASY-RECONST utility routine is started by the main task so that the AIM file generation is adapted to match the shadow files. If a DMS error occurs on a shadow file, all operations on this shadow file are marked in the log as invalid. Reconstruction of other shadow files, however, is continued.

Once the AIM file generation has been processed, the RECONST run is terminated abnormally, i.e. any further AIM file generations will not be processed and will remain in the status "READY FOR RECONST".

The user can have the RECONST run monitored by means of a job variable and can respond appropriately in the case of abnormal termination (e.g. by restarting LEASY-RECONST, terminating the main task and informing users and/or system administration, etc.).

9.7 Examples

Example 1 for LEASY-RECONST

The reconstruction run is to be monitored by a separate procedure. The name of the job variable is to be *RECOCHECK*. The RECONST-ENTER file is to be called *E.RECONST.START*. The ENTER file monitoring the reconstruction run is to be called *E.RECONST.MONITOR*. The log of the reconstruction run is to be written to the *W.RECO.LOG* file. If there is an error in the reconstruction run, the main task is to be terminated, but all transactions that have been started are to be terminated normally.

Creating the job variable

The job variable that is to monitor the reconstruction run is cataloged with:

```
/CREATE-JV JV=RECOCHECK
```

Monitoring procedure

The ENTER file with the name *E.RECONST.MONITOR* which is to monitor the reconstruction run contains the following calls:

```
/SET-LOGON-PAR
/WAIT-EVENT UNTIL=*JV(COND=((RECOCHECK,1,2)=C'$R'))               (1)
/.LOOP    REMARK
/WAIT-EVENT UNTIL=*JV(COND=((RECOCHECK,1,2)<>C'$R'),TIMEOUT=LOOP) (2)
/SKIP-COM TO-LABEL=NORMEND,IF=*JV(COND=((RECOCHECK,1,2)=C'$T'))   (3)
/START-LEASY-MASTER
TESTCAT
CLOS                                                              (4)
*END
/.NORMEND REMARK
/EXIT-JOB
```

(1) Wait until LEASY-RECONST is running.

(2) Wait until LEASY-RECONST is terminated. (Once the wait time for *WAIT-EVENT* has elapsed, the *WAIT-EVENT* command is repeated.)

(3) When LEASY-RECONST is terminated normally, there is no further action.

(4) When LEASY-RECONST is terminated abnormally, LEASY-MASTER is called and the main task is terminated by means of *CLOS* (no more new transactions are permitted but current transactions are continued).

RECONST ENTER procedure

In order to define a monitoring job variable for the reconstruction run, the ENTER file *E.RECONST.START* must be created by the user himself/herself.

```
/SET-LOGON-PAR
/ASSIGN-SYSOUT TO-FILE=W.RECO.LOG                        (1)
/MODIFY-JV (RECOCHECK,1,2),SET-VAL=C'  '                 (2)
/ENTER-JOB E.RECONST.MONITOR                             (3)
/START-LEASY-RECONST MONJV=RECOCHECK                     (4)
CAT TESTCAT,COP=(Y,A)
END
/SET-JOB-STEP
/ASSIGN-SYSOUT TO-FILE=*PRIMARY
/PRINT-FILE FILE-NAME=W.RECO.LOG                         (5)
/EXIT-JOB SYSTEM-OUTPUT=*NONE
```

(1) The log of the reconstruction run is written to the *W.RECO.LOG* file.

(2) The contents of the job variable *RECOCHECK* are deleted.

(3) The ENTER file *E.RECONST.MONITOR* is started.

(4) The LEASY-RECONST utility roputine is started; *RECOCHECK* is defined as the job variable monitoring the program.

(5) The log of the reconstruction run is printed.

Main LEASY task

Once the job variable and the two ENTER files have been created, the main task can be started.
The following control statements must be specified in the LEASY-MAINTASK utility routine:

```
/START-LEASY-MAINTASK
CAT=TESTCAT
   ...
LOG=Y,M   (bzw. LOG=A,M)
AUT=Y
REN=ENTER-JOB E.RECONST.START
   ...
END
```

The main task then automatically starts *E.RECONST.START*, which in turn starts *E.RECONST.MONITOR*.

Example 2 for LEASY-RECONST

Runtime log 1: Execution of LEASY-RECONST with input errors

```
/START-LEASY-RECONST ──────────────────────────────────────────────────  (1)
%  BLS0523 ELEMENT 'RECONST', VERSION '06.2A' FROM LIBRARY
   ’:2OSL:$TSOS.SYSPRG.LEASY.062’ IN PROCESS
%  BLS0524 LLM 'LEASY-RECONST', VERSION '06.2A' OF '2006-03-08 01:28:19'
LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS
RESERVED
%  LEA0401 LEASY AFTER IMAGE PROGRAM (RECONST) VERSION V6.2A STARTED
*COM ERRORED DEMONSTRATION EXAMPLE ────────────────────────────────────  (2)
*CAT LCAT,GEN=0 ───────────────────────────────────────────────────────  (3)
%  LEA6002 INVALID GENERATION-PARAMETER ──────────────────────────────  (4)
*MOD PRINT=NORMAL,UNLOAD=NO,PRI=N  ────────────────────────────────────  (5)
%  LEA2002 DUPLICATE PARAMETERS ARE NOT ALLOWED ───────────────────────  (6)
*END  ─────────────────────────────────────────────────────────────────  (7)
%  LEA2402 *CAT COMMAND MISSING OR INVALID  ───────────────────────────  (8)
%  LEA0411 ABNORMAL TERMINATION OF LEASY AFTER IMAGE PROGRAM (RECONST)   (9)
%  EXC0732 ABNORMAL PROGRAM TERMINATION. ERROR CODE ’NRT0101’: /HELP-MSG
NRT0101
```

(1) The LEASY-RECONST utility routine is called.

(2) Comment line.

(3) Assignment of the LEASY catalog LCAT and file generation 0 of the AIM file.

(4) Error message from the LEASY-RECONST utility routine: file generation 0 is not allowed.

(5) Function selection **MOD*: a list of the records in the AIM file is to be output; the file should only be closed if it is to be opened in a different OPEN mode.

(6) The statement is rejected because the operand *PRI* was specified twice.

(7) End of the statement input.

(8) Error message from the utility routine: no LEASY catalog is specified because the preceding **CAT* statement was rejected.

(9) The LEASY-RECONST utility routine is terminated, but no reconstruction run is started.

Runtime log 2: Reconstruction run without input errors

```
/START-LEASY-RECONST-------------------------------------------------------------- (1)
%  BLS0523 ELEMENT 'RECONST', VERSION '06.2A' FROM LIBRARY
   ':20SL:$TSOS.SYSPRG.LEASY.062' IN PROCESS
%  BLS0524 LLM 'LEASY-RECONST', VERSION '06.2A' OF '2006-03-08 01:28:19' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2006. ALL RIGHTS RESERVED
%  LEA0401 LEASY AFTER IMAGE PROGRAM (RECONST) VERSION V6.2A STARTED
*COM ERRORED DEMONSTRATION EXAMPLE ----------------------------------------------- (2)
*CAT LCAT,GEN=2 ------------------------------------------------------------------ (3)
*SES FRO=5,TOS=6 ----------------------------------------------------------------- (4)
*DAT FRO=2006-03-29,TOD=2006-04-29 ----------------------------------------------- (5)
*FIL (LCAT.MITABDAT) ------------------------------------------------------------- (6)
*MOD UNL=N,SIU=Y,FRE=Y ----------------------------------------------------------- (7)
*REP LEN=A,EXT=(1,105) ----------------------------------------------------------- (8)
*END ----------------------------------------------------------------------------- (9)
SESS#=00005 TSN#=2J4K D=2006-03-29 T=14:52:42-S     } ------------------------- (10)
SESS#=00006 TSN#=2NE2 D=2006-03-30 T=14:57:58-S     }
%  LEA0410 NORMAL TERMINATION OF LEASY AFTER IMAGE PROGRAM
```

(1) The LEASY-RECONST utility routine is called.

(2) Comment line.

(3) Assignment of the LEASY catalog LCAT. The second AIM file generation is to be used for the reconstruction run.

(4) Reconstruction is to be carried out from the session with session number 5 to the session with session number 6.

(5) Reconstruction is restricted to a specific date range.

(6) The MITABDAT file in the LEASY catalog LCAT is to be reconstructed.

(7) The **MOD* statement is then used to select the following functions:

- The files are to be closed only if they are to be opened subsequently in a different OPEN mode.
- The secondary index files are to be reconstructed in the same run.
- After the reconstruction run the common memory is to be released for application programs.

The default values are used for the other functions.

(8) The **REP* statement is used to define listing control:

Positions 1-105 in record types *STOR*, *DLET*, *SETS*, *PUTX*, *PUTS*, *STOD*, *ELIR* and *ELIF* are to be output.
All other record types are processed as with *LEN=L*.

(9) Termination of LEASY-RECONST and start of the reconstruction run.

(10) All session entries in the specified time span are output to SYSOUT, together with the session number, the TSN of the main task, and the date and time.

Printout of the selected functions of the LEASY-RECONST utility routine

This printout is generated automatically for each reconstruction run.

```
LEASY AFTER-IMAGE-RECONSTRUCTION           DATE OF RECONSTRUCTION: 2006-03-30   TIME: 15:18:22-S

SELECTED PARAMETERS:
--------------------

LEASY CATALOG  (*CAT):     SELECTED CATALOG NAME:............. :2OSC:$BS2000D4.LCAT
                           AFTER IMAGE FILE GENERATION NUMBER: 0002
                           UPDATING SHADOW FILES:............. NO

OPERATION MODE (*MOD):     PRINT:............................. NORMAL
                           UPDATING LEASY DATA FILES:......... YES
                           UPDATING LEASY SI-FILES:........... YES
                           TRANSACTIONS SELECTED:............. ALL
                           UNLOAD CLASS-5-MEMORY:............. NO
                           SET FREE CMMAIN FOR RUNTIME-SYSTEM: YES

LIST REPORT    (*REP):     REQUESTED LENGTH OF RECORD OUTPUT:. ALL
                           RECORD EXTRACTION:................. POSITION LENGTH
                                                                      1    105
                           LIST INVALID RECORDS:.............. YES
                           RECORD SELECTION:.................. ALL RECORDS
                           LIST USER-INFORMATION:............. NO
                           LIST PROTOCOL RECORDS:............. NO

DATE FILTER    (*DAT):     FROM DATE (YYYY-MM-DD):............ 2006-03-29
                           TO   DATE (YYYY-MM-DD):............ 2006-04-29

SESSION FILTER (*SES):     FROM SESSION-NUMBER:..............      5
                           TO   SESSION-NUMBER:..............      6
                           LAST TRANSACTION:.................. END   OF TO-SESSION

FILE FILTER    (*FIL):     SELECTED FILES:.................... :2OSC:$BS2000D4.LCAT.MITABDAT

RANGE FILTER   (*RAN):     NOT SPECIFIED
```

Printout of the reconstruction log

The meanings of the individual fields are explained in the description of the **REP* statement.

```
LEASY AFTER IMAGE RECONSTRUCTION, AIMFILE=:2OSC:$BS2000D4.LCAT.LEASYAIM(*0002)  NEW PAMBLOCK-LINK:BLOCK# = .....

OP  XYS POS SESSION TRANS ITR TSN  FILE                          TIME        RECORD
------------------------------------------------------------------------------------------------------------------
SESS      822 00005     0   0 2J4K                               14:52:42-S  D=2006-03-29
CATD      874 00005     0   0 2JEX                               14:53:17-S  T=2JEX U=BS2000D4 P=PERSDAT I=TSN-..
OPEN      968 00005     0   0 2JEX                               14:53:17-S  MITABDAT:2OSC:$BS2000D4.LCAT.MITABDAT
       FILE=:2OSC:$BS2000D4.LCAT.MITABDAT                  OPENMODE=INOUT
OPTR     1017 00005     1   1 2JEX                               14:53:23-S  B=         H=     A=$DIALOG  #=
STOR     1118 00005     1   1 2JEX :2OSC:$BS2000D4.LCAT.MITABDAT 14:54:15-S  KEY=X'0004463B'
   EXTRACT001=(00001,00105):  MUENCHEN  YZ989GANGGASSE 3A
STOR     1219 00005     1   1 2JEX :2OSC:$BS2000D4.LCAT.MITA     14:55:11-S  KEY=X'0006AC43'
   EXTRACT001=(00001,00105):    ENTENHSN. YZ989WALTSTRASSE 8
CLTR     1243 00005     1   1 2JEX                               14:55:13-S
MTSK     1307 00005     0   0 2NE2                               14:57:41-S  D=2006-03-30
SESS     1371 00006     0   0 2NE2                               14:57:58-S  D=2006-03-30
CATD     1423 00006     0   0 2NEV                               14:59:04-S  T=2NEV U=BS2000D4 P=PERSDAT I=TSN-..
OPEN     1517 00006     0   0 2NEV                               14:59:08-S  MITABDAT:2OSC:$BS2000D4.LCAT.MITABDAT
       FILE=:2OSC:$BS2000D4.LCAT.MITABDAT                  OPENMODE=INOUT
CLOS     1544 00006     0   0 2NEV                               14:59:08-S
CATD     1596 00006     0   0 2NEV                               15:03:16-S  T=2NEV U=BS2000D4 P=PERSDAT I=TSN-..
OPEN     1690 00006     0   0 2NEV                               15:03:17-S  MITABDAT:2OSC:$BS2000D4.LCAT.MITABDAT
       FILE=:2OSC:$BS2000D4.LCAT.MITABDAT                  OPENMODE=INOUT
OPTR     1739 00006     2   1 2NEV                               15:03:49-S  B=         H=         A=$DIALOG  #=
CLTR     1763 00006     2   1 2NEV                               15:04:22-S
OPTR     1812 00006     4   1 2NEV                               15:04:36-S  B=         H=         A=$DIALOG  #=
DLET     1843 00006     4   1 2NEV :2OSC:$BS2000D4.LCAT.MITA     15:04:47-S  KEY=X'000116B7'
CLTR     1867 00006     4   1 2NEV                               15:04:50-S
OPTR     1916 00006     5   1 2NEV                               15:04:53-S  B=         H=         A=$DIALOG  #=
STOR     2017 00006     5   1 2NEV :2OSC:$BS2000D4.LCAT.MITABD   15:05:47-S  KEY=X'000171F7'
   EXTRACT001=(00001,00105):  MUENCHEN  AB212AM KNACKEPUNKT 15
STOR     2118 00006     5   1 2NEV :2OSC:$BS2000D4.LCAT.MIT      15:06:20-S  KEY=X'00022997'
   EXTRACT001=(00001,00105): MUENCHEN  AB212RAUCHERSTEG 3
STOR     2219 00006     5   1 2NEV :2OSC:$BS2000D4.LCAT.MITA     15:07:23-S  KEY=X'000470CA'
   EXTRACT001=(00001,00105):    MUENCHEN  DP212SCHWABENSTRASSE 30
CLTR     2243 00006     5   1 2NEV                               15:07:29-S
MTSK     2307 00006     0   0 2NGK                               15:11:35-S  D=2006-03-30
```

10 LEASY-SAVE

LEASY-SAVE is a utility routine for saving

- the LEASY catalog
- all the files in the LEASY catalog

and, if available,

- the shadow files
- the occurrences of temporary files
- the status file between two sessions.

Saving is performed using the software product ARCHIVE. Information on the functions and operation of ARCHIVE can be found in the "ARCHIVE (BS2000/OSD)" manual.

The LEASY-SAVE utility routine offers guided dialog for input, but can also be used in batch mode.

Once LEASY-SAVE has been loaded, it issues the message:

```
LEA0601 LEASY SAVE PROGRAM VERSION V6.2A STARTED
```

Dialog with the user then begins. Each input is requested by a LEASY message, which includes the possible responses. Input lines must not exceed 255 characters (including all characters such as blanks etc.). Each input must begin in the first column of the input line.

A comment statement can be issued at any point in the program:

```
*COM[any text]
```

In guided dialog the following inputs are requested by means of LEASY messages:

– type of volume and retention period
– volume serial numbers
– device type
– output medium for the REPORT list
– type of save (full or partial)
– name of the LEASY catalog
– if necessary, user IDs for temporary files

10.1 Defining the volume

The guided dialog begins after the utility routine has been started. First the type of volume for archival and the retention period are defined.

Prompt

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

Possible responses

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. The retention period must be specified together with the device type. If this parameter is omitted, there is no retention period.

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

If saving of the specified files and the associated catalog is terminated, the prompt is re-issued. This means that several LEASY catalogs can be saved to different volumes or in different modes during a single save run.

10.2 Assigning volume serial numbers

Once the type of volume has been defined, the volume serial numbers are specified.

Prompt

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

Possible responses

PO[OL] For saving to tape only.

The volume serial numbers are taken from the tape pool of the associated directory file in ascending order (see the "ARCHIVE (BS2000/OSD)" manual). If the pool does not contain sufficient free tapes, additional tapes are requested from the operator’s tape pool.

OP[ERATOR] For saving to tape only.

This entry requests a VSN at the console during the save operation. This message has the number *DMS0591*. The specified VSN is entered in the tape pool.

vsn For saving to disk or tape.

The volume with the VSN *vsn* is requested.

(vsn,...) Several volume serial numbers can be specified. The volumes (disks or tapes) are requested in ascending order as they are required. Additional tapes are requested from the operator’s tape pool.

Blank or DUE For saving to tape only.

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

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

ℹ An area comprising 160 characters is reserved for transferring the volume serial numbers. All VSNs in excess of 160 characters (without blanks) are ignored, but the utility routine continues to execute after issuing an appropriate informative message.

10.3 Assigning the device

Once a volume serial number has been defined, the device type is specified.

Prompt

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

Possible responses

device — defines the device type.

Possible entries are specified in the manual "Introductory Guide to DMS".

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.

10.4 Assigning the REPORT list

Once the device type has been defined, the output medium for the REPORT list is assigned.

Prompt

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

Possible responses

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.

10.5 Defining the type of save operation

Following the REPORT list specification, the user specifies whether a full save or an incremental save is to be performed.

Prompt

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

Possible responses

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.

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*.

10.6 Assigning the LEASY catalog

Once the type of save operation has been defined, the name of the LEASY catalog whose files are to be saved is specified.

Prompt

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

Possible response

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

where

catid
Specifies the catalog identifier of the public volume set which contains the LEASY catalog. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-SAVE was started.
catid may comprise up to 4 characters, enclosed in colons.

$userid
Is only permitted when LEASY-SAVE is running under the TSOS privilege.

file-catalog
Specifies the name of the LEASY catalog.

CATID=YES
Specifies that file names are to be stored under *catid* in the ARCHIVE directory file. This operand corresponds to the *CATID=YES* operand in the ARCHIVE utility routine (see the "ARCHIVE (BS2000/OSD)" manual).

Save runs for a catalog can only be carried out in the same mode (either with or without the operand *CATID=YES*). The *DIRCONV* conversion routine can be used to convert an ARCHIVE directory file without *catid* to one with *catid* (see the "ARCHIVE (BS2000/OSD)" manual).

If LEASY-SAVE was not started under a privileged user ID (cf. section 10.7 on page 268), saving of the catalog and its files begins as soon as the catalog name has been entered.

10.7 Specifying user IDs for temporary files

If LEASY-SAVE was started under a privileged user ID, it is now possible to enter user IDs under which occurrences of temporary files from the previously entered catalog are stored. If no entry is made, or if LEASY-SAVE is not executing under a privileged user ID, only occurrences under the user ID of the LEASY catalog are saved.

Prompt

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

Possible responses

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

Specifies identifiers under which occurrences of temporary files are stored. *catid* is the catalog identifier of the public volume set which contains the occurrences of temporary files. This operand must be specified if the occurrences of temporary files are on a different public volume set from LEASY-SAVE. If *catid* is omitted, LEASY uses the catalog identifier of the public volume set with the user ID under which LEASY-SAVE was started.
catid may comprise up to 4 characters, enclosed in colons.

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

10.8 Continuing the save operation

If saving of the files in a LEASY catalog has been terminated, at this point the user has the following options:

- Save a further LEASY catalog under the same conditions.
- Modify the ARCHIVE specifications.
- Terminate LEASY-SAVE.

Prompt

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

Possible responses

Y[ES]
A further LEASY catalog is to be saved. The specified operands remain set. If the previous LEASY catalog was saved to tape and the appropriate volume serial numbers were explicitly specified, it is not possible to specify *YES*. This is because the newly created save version would be overwritten by the save version of the next catalog.

N[O]
New save operands are requested. Subsequently, a prompt for input of the volume type appears.

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

After the LEASY files have been saved to magnetic tape or cartridge using the LEASY-SAVE utility routine, the tape or cartridge is rewound and unloaded. (Older LEASY versions left the save tape in its current position following the save run.)

10.9 LEASY-SAVE error handling

Syntax or semantics errors in the input data result in error messages.

In interactive mode it is possible to enter the correct data following a new prompt. In batch mode, the routine is terminated by means of

```
TERM MODE=ABNORMAL,UNIT=STEP
```

If other errors occur which jeopardize continuation of the save operation, LEASY-SAVE is likewise terminated with

```
TERM MODE=ABNORMAL,UNIT=STEP.
```

In this case, it is possible that not all the files in a LEASY catalog are saved, or that the operation for saving the files of a catalog has not even started.

10.10 Use of ARCHIVE

The LEASY-SAVE utility routine uses the BS2000 software product ARCHIVE, which is called via the *ARCHIVE* macro. LEASY-SAVE causes directory files to be generated by ARCHIVE for automatic management of the saved files. The names of these files are based on the following format:

```
file-catalog.LEASAVE.device-type
```

Meaning

file-catalog	Name of the LEASY catalog.
device-type	
=TAPE	For saving to tape.
=DISK	For saving to disk.

Users can generate the ARCHIVE directory files themselves, for example, in order to create a tape pool beforehand.

10.11 Carrying out the save operation

The following types of save operation must be distinguished when using LEASY-SAVE:

- Online save
- Offline save

The following requirements apply when an **online save** operation is performed for the files of a LEASY catalog:

- LEASY maintask has been started.
- The current LEASY catalog contains only master files with the access method ISAM or PAM.
- All master files (including the SI files) of the current LEASY catalog must be reserved for READ-ONLY mode (*ROM=Y* parameter in the LEASY-CATALOG statement **CAT* or **FIL*).
- When the catalog was configured, **no** specifications were made for shadow files (*CPC/CPS*).
- When the *ROMS* function is executed in the LEASY-MASTER utility routine, *ALL must be specified in screen mask 47 (add file for ROMS).
- The *ROMS* function must have been completed successfully; no transactions are open.

The following requirements apply when an **offline save** operation is performed for the files of a LEASY catalog:

- The session working with this catalog is terminated properly.
- Open transactions are rolled back before starting LEASY-SAVE by means of a LEASY-MAINTASK warm start (all the BIM files are empty).
- Only LEASY-SAVE accesses the LEASY catalog (no other applications have opened the catalog)

Only if the requirements which apply for the case type concerned are satisfied does LEASY-SAVE read the catalog entries and generate ARCHIVE statements for the files found. If entries relating to shadow files were made when generating the catalog (*CPC/CPS*), both file and shadow file are saved.

In the case of an offline save, the following actions are also performed:

- All existing occurrences of temporary files are established.
- Files which were defined in the LEASY catalog with *LEA=F* (foreign files) are saved if a DMS file was allocated via the file link name (*LINK-NAME*=logical file name).

Restriction

The conventions governing ARCHIVE mean that it is not possible to save files specified with a :*catid*:. Files specified with a foreign user ID can be saved by privileged users only. The LEASY utility routine LEASY-SAVE checks all the file names as to whether they satisfy the specified conditions. If the check is positive, the file names are not passed on to ARCHIVE and messages are displayed to indicate that saving is not possible.

The generated ARCHIVE commands are written to a 28-Kbyte statement area. If this storage space is too small to accommodate all the file names in a catalog, the save operation is performed in several ARCHIVE runs.

Using a tape pool

It is advisable to create a tape pool for each LEASY catalog large enough to meet the particular requirements (see the ARCHIVE command *POOL*). There is then no need to assign volume serial numbers during the LEASY-SAVE run. In this case, ARCHIVE uses the next free tape for saving. The ARCHIVE statement *PURGE* enables save versions that are no longer needed to be deleted. The tapes thereby made available again can subsequently be used for further save operations.

10.12 Restoring destroyed files

Destroyed files can be restored by means of ARCHIVE if they were saved with LEASY-SAVE. The following commands are specified:

```
FILES NA=filename
RESTORE DIR=file-catalog.LEASAVE.TAPE
```

when saving to tape, and

```
DRESTORE DIR=file-catalog.LEASAVE.DISK
```

when saving to disk.

Further operands can be found in the "ARCHIVE (BS2000/OSD)" manual.

i The LEASY catalog is protected by an internal password; consequently, restoration can be performed by privileged users only.

11 Messages of the LEASY utility routines

The messages of the LEASY utility routine are provided by message file *SYSMES.LEASY.061*.

More detailed information on each message can be called up with the *HELP-MSG* command:

```
HELP-MSG MSG-ID=msgid[,LANGUAGE=D / E]
```

Messages can be output in German (D) or English (E). The default value depends on the system generation. The task-specific default value can be modified with the following command:

```
MODIFY-MSG-ATTRIBUTES TASK-LANGUAGE = D / E
```

All utility routine messages consist of the three letters *LEA* and a four-digit code:

```
LEAxyzz
```

where

x	Code identifying the type of error
0	Headers
1	Messages based on DMS error recognition
2	Logical errors
3	Program inconsistencies
4	RDATA module messages
5	System status messages
6	Syntax errors
7	Macro messages
8	Other errors

Table 7: Code x - Identification of the type of error

y	Code identifying the utility routine which initiated the message
0	Messages that can be sent by several utility routines
1	LEASY-CATALOG messages
2	LEASY-LOADSI messages
3	LEASY-MAINTASK messages
4	LEASY-RECONST messages
5	LEASY-MASTER messages
6	LEASY-SAVE messages
7	LEASY-IOTASK messages
8	LEASY-CONVERT messages

Table 8: Code y - Identification of the utility routine

`zz` Two-digit code for a more exact identification

Note:

Messages LEA0097, LEA0098 and LEA0099 are output by the LEASY runtime system and not by the LEASY utility routines.

LEA0001 `PLEASE TYPE IN NAME OF LEASY DIRECTORY`

Response
Enter name of LEASY directory or *END.

LEA0097 `Warning: Calling program in 24-bit-mode, subsystem in upper address space. LEASY runtime system will be loaded into the class-6 memory.`

LEA0098 `LEASY TERM : RC-CC=  (&00), RC-LC=  (&01), RC-LCE=  (&02)`

Meaning
LEASY operation TERM failed with error. See returncodes for reason.

Response
Check returncode by use of manual

LEA0099 `LEASY STXIT: INTERRUPTION-CODE = (&00), PC = (&01)`

Meaning
LEASY application ended without TERM (INTERRUPTION-CODE <>90) or LEASY application ended with TERM but still open transactions (INTERRUPTION-CODE=90)

Response
No

LEA0101 `LEASY CATALOG PROGRAM VERSION (&00) STARTED`

Meaning
Program start. (&00): Version number of the utility routine.

LEA0110 `NORMAL TERMINATION OF LEASY CATALOG PROGRAM`

Meaning
Program termination without errors.

LEA0111 `ABNORMAL TERMINATION OF LEASY CATALOG PROGRAM`

Meaning
Program termination with serious errors.

LEA0112 `ABNORMAL INPUT TERMINATION OF LEASY CATALOG PROGRAM`

Meaning
Program termination due to error caused while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error that occurred.

Response
Correct error with the aid of the other error messages.

LEA0201 `LEASY PROGRAM FOR SECONDARY INDICES LOADSI VERSION (&00) STARTED`

Meaning
Program start. (&01): Version number of the utility routine.

LEA0203 `PLEASE TYPE IN FILE SPECIFICATION`

Meaning
Request to enter a primary file.

Response
Enter a file specification in accordance with the input description.

LEA0204 `PLEASE TYPE IN SPECIFICATION FOR SECONDARY INDEX`

Meaning
Request to enter SI specifications.

Response
Enter a secondary index specification in accordance with the input description.

LEA0210 `NORMAL TERMINATION OF LEASY LOADSI PROGRAM`

Meaning
Program termination without errors.

LEA0211 `ABNORMAL TERMINATION OF LEASY LOADSI PROGRAM`

Meaning
Program termination with serious errors.

LEA0212 `ABNORMAL INPUT TERMINATION OF LOADSI`

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA0301 `LEASY MAINTASK VERSION (&00) STARTED`

Meaning
Program start. (&01): Version number of the utility routine.

LEA0310 `NORMAL TERMINATION OF LEASY MAINTASK BECAUSE OF CLOSE DOWN FUNCTION`

Meaning
The CLOSE DOWN function has been initiated for LEASY-MAINTASK.

LEA0312 `ABNORMAL INPUT TERMINATION OF LEASY MAINTASK PROGRAM`

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA0401 LEASY AFTER IMAGE PROGRAM (RECONST) VERSION (&00) STARTED

Meaning
Program start. (&01): Version number of the utility routine.

LEA0410 NORMAL TERMINATION OF LEASY AFTER IMAGE PROGRAM

Meaning
Program termination without errors.

LEA0411 ABNORMAL TERMINATION OF LEASY AFTER IMAGE PROGRAM (RECONST)

Meaning
Program termination with serious errors.

LEA0412 ABNORMAL INPUT TERMINATION OF LEASY AFTER IMAGE PROGRAM

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA0501 LEASY MASTER PROGRAM VERSION (&00) STARTED

Meaning
Program start. (&01): Version number of the utility routine.

LEA0510 NORMAL TERMINATION OF LEASY MASTER PROGRAM

Meaning
Program termination without error.

LEA0511 ABNORMAL TERMINATION OF LEASY MASTER PROGRAM

Meaning
Program termination with serious errors.

LEA0512 ABNORMAL INPUT TERMINATION OF LEASY MASTER PROGRAM

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA0516 (*END=END OF PROGRAM)

LEA0518 FUNCTION SELECTION (OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM)

LEA0519 FUNCTION SELECTION (OR +=CONTINUE; OR BLANK; OR *END)

LEA0520 PLEASE ENTER PASSWORD

LEA0521 PLEASE ENTER NUMBER OF CYCLES (BLANK=(&00); MIN=0; MAX=9999)

LEA0522 PLEASE ENTER DISPLAY TIME (BLANK=(&00) SECONDS; MIN=0; MAX=9999)

LEA0523 PLEASE ENTER FUNCTION CODE OF TABLE YOU WANT TO DISPLAY CYCLICALLY

LEA0524 (BLANK=NO LOCK)

LEA0525 PLEASE ENTER SESSION SPECIFIC TRANSACTION NUMBER

LEA0526 OR IDENTIFICATION OF TRANSACTION YOU WANT TO LOCK

LEA0527 PLEASE ENTER LOGICAL NAME OF FILE YOU WANT TO LOCK

LEA0528 PLEASE ENTER TASK SEQUENCE NUMBER FOR TASK

LEA0529 YOU WANT TO DISPLAY STATISTICALLY

LEA0530 PLEASE ENTER AIM FILE GENERATION NUMBER YOU WANT TO ERASE

LEA0531 OR IDENTIFICATION YOU WANT TO UNLOCK

LEA0532 OR IDENTIFICATION YOU WANT TO ROLLBACK

LEA0533 OR IDENTIFICATION YOU WANT TO DISPLAY STATISTICALLY

LEA0534 PLEASE ENTER LOGICAL NAME OF FILE YOU WANT TO UNLOCK

LEA0535 (BLANK=NO UNLOCK)

LEA0536 (BLANK=NO ROLLBACK)

LEA0537 (BLANK=NO CYCLICALLY DISPLAY)

LEA0538 (BLANK=NO ERASE)

LEA0539 DO YOU REALLY WANT TO TERMINATE THE LAST IO TASK (Y=YES; N=NO)?

LEA0540 FUNCTION SELECTION (OR A=ADD RETCODE; OR D=DELETE RETCODE;

LEA0541 OR BLANK=MAINTASK SELECTION; OR *END=END OF PROGRAM):

LEA0542 CONTINUE WITH DUE.

LEA0601 LEASY SAVE PROGRAM VERSION (&00) STARTED

Meaning
Program start.
(&00): Version number of the utility routine.

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

Meaning

LEASY asks for input of the output medium for the ARCHIVE report list.

Response

SYSO:	Output of the report list to SYSOUT.
SYSL:	Output of the report list to SYSLST.
BOTH:	Output of the report list to SYSOUT and to SYSLST.
NONE:	No output of the report list.
*END:	Terminate program.

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

Meaning

LEASY asks for input of the type of saving.

Response

YES::	Save modified files.
YES,LARGE:	Save modified files with LARGE in the catalog.
YES,number of pages:	Save modified files as of the specified number of pages.
NO:	Save all files.
*END:	Terminate program.

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

Meaning

LEASY asks for input of the name of the directory.

Response

Reply in the form <:catid:> <$userid.> directory <,CATID=YES>.

catid:	pubset of the directory.
$userid:	user identification of the directory (only for $TSOS).
directory:	name of the directory.
CATID=YES:	filenames with catid in the ARCHIVE-directory.

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

Meaning
LEASY asks for input of the device type and the retention period in days for the version of save.
For DAYS use a value between 0 and 32767.
If RETPD is not given, no retention period is stated.

Response
DISK <,RETPD=DAYS>: Specify for disks with or without retention period.
TAPE <,RETPD=DAYS>: Specify for tapes with or without retention period.
*END: Terminate program.

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

Meaning
LEASY asks for input of the VSN.

Response
vsn: Volume serial number of the volume.
(vsn,...): Volume serial numbers of more than one volume.
PO: Volume serial numbers from tape pool of the ARCHIVE library.
OP: Volume is to be made available by the operator.
no specification:Same function as OP.
*END: Terminate program.

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

Meaning
LEASY asks for input of the name of the device.

Response
Name of device: See DMS manuals for possible specifications.
No specification: Same function as TAPE specification.
*EOF: Terminate program.

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

Meaning
LEASY asks for the user identifications under which occurrences
of temporary files of the last entered directory are stored.

Response
User identifications: 1 to 5 userids in the form:
<:catid:>$userid1, .. ,<:catid:>$userid5
No specification: Occurrences of the temporary files under catalog ID.

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

Meaning
LEASY asks whether or not a further LEASY directory is to be saved.

Response

Y:	A further LEASY directory is to be saved.
N:	New save operands are to be queried.
*END::	Terminate program.

LEA0610 NORMAL TERMINATION OF LEASY SAVE PROGRAM

Meaning
Program termination without errors.

LEA0611 ABNORMAL TERMINATION OF LEASY SAVE PROGRAM

Meaning
Program termination with serious errors.

LEA0612 ABNORMAL INPUT TERMINATION OF SAVE

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA0701 LEASY IO-TASK VERSION (&00) STARTED

Meaning
Program start.
(&00): Version number of the utility routine.

LEA0710 NORMAL TERMINATION OF LEASY IO-TASK BECAUSE CLOSE-DOWN FUNCTION

Meaning
The CLOSE DOWN function has been initiated for LEASY-IOTASK.

LEA0711 *** LEASY IO-TASK FINISHED WITH ERROR ***

Meaning
Program termination of run with serious error.

LEA0801 LEASY CONVERT PROGRAM VERSION (&00) STARTED

Meaning
(&00): Version number of the utility routine.

LEA0810 NORMAL TERMINATION OF LEASY CONVERT PROGRAM

Meaning
Program termination without errors.

LEA0811 ABNORMAL TERMINATION OF LEASY CONVERT PROGRAM

Meaning
Program termination with serious error.

LEA0812 ABNORMAL INPUT TERMINATION OF LEASY CONVERT PROGRAM

Meaning
Program termination due to error while reading in from SYSDTA.
The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA1001 DMS ERROR (&00) WITH FILE (&01)
(B) Routing code: U Weight: 70

Meaning
DMS error (&00) has occurred during processing of file (&01).

Response
Identify error with aid of the "System Messages" manual.

LEA1002 ERROR (&00) WITH JOB VARIABLE (&01)

Meaning
JV error (&00) has occurred during processing of JV (&01).

Response
Identify error with aid of the "System Messages" manual.

LEA1101 DMS ERROR WHILE TESTING SI-FILE FOR SI-NAME (&00)

Meaning
A DMS error was detected while checking the SI file for pointers for the secondary index (&00). The preceding message LEA1001 identifies the error.

Response
Identify error with aid of the "System Messages" Manual.

LEA1102 DMS ERROR WHILE TESTING SI-FILE FOR SI-NAME(S)

Meaning
A DMS error was detected while checking the SI file for pointers.

Response
Identify error with aid of the "System Messages" manual.

LEA1103 `DMS ERROR WHILE ERASING SECONDARY INDEX FILE`

Meaning
DMS error occurred while erasing SI file. The preceding message LEA1001 indicates which DMS error occurred.

Response
Identify error with aid of the "System Messages" Manual.

LEA1201 `LEASY DIRECTORY READ ERROR`

Meaning
DMS error has occurred while reading LEASY directory.

LEA1202 `ERROR WHILE HANDLING PRIMARY FILE (&00)`

Meaning
An error has occurred while processing the file (&00). The preceding message LEA1001 indicates the DMS error.

Response
Wait for prompt.

LEA1203 `ERROR WHILE HANDLING SECONDARY INDEX FILE (&00)`

Meaning
An error has occurred while processing the SI file (&00). The preceding message LEA1001 indicates the DMS error.

Response
Wait for prompt.

LEA2000 `VERIFY NOT POSSIBLE FOR FOREIGN USERID WITH FILE (&00)`

Meaning
It has been discovered while opening the file (&00) that this file has not been closed correctly. The file is not, however, cataloged under the user ID under which the LEASY-MAINTASK utility routine is executed.
LEASY-MAINTASK is unable for this reason to carry out VERIFY.

Response
Execute VERIFY for the file (&00) in the user ID of the file (&00).

LEA2001 `DUPLICATE COMMANDS ARE NOT ALLOWED`

Meaning
At least one statement in the statement sequence has been repeated.

Response
Correct input.

LEA2002 `DUPLICATE PARAMETERS ARE NOT ALLOWED`

Meaning
A parameter keyword was specified more than once.

Response
Correct error and repeat input.

LEA2003 `LEASY DIRECTORY IS IN USE`

Meaning
The addressed LEASY directory file has already been opened.

LEA2004 `FILE (&00) NOT FOUND`

Meaning
The file (&00) is not contained in the tables of the common memory CMMAIN.

LEA2005 `ERROR: DIFFERENT VERSIONS. MAINTASK: (&00). (&01): (&02).`

Meaning
The common memory CMMAIN was created by the version (&00) of LEASY-MAINTASK utility routine. The version (&02) of (&01) utility routine is incompatible with that.

Response
Start the same version of LEASY-MAINTASK utility routine and (&01) utility routine.

LEA2006 `WARNING: (&00) STATEMENT IN BS2000 VERSIONS LESS THAN (&01) IGNORED`

Meaning
The (&00) statement is used only in BS2000 versions (&01) or higher.
The default value of the statement is used.
(&00): Name of the statement
(&01): BS2000 version in the form 'vv.v'.

LEA2007 `SIZE OF FILE (&00)  GREATER THAN 32 GBYTE`

Meaning
File (&00) has filesize greater than 32 Gigabyte. LEASY cannot work with that file.

Response
Reduce file until filesize is smaller than 32 gigabyte

LEA2101 `MOD = NEW: FILENAME MUST BE FULLY QUALIFIED`

Meaning
The file name must be fully qualified when creating a LEASY file.

Response
Correct error and repeat input.

LEA2102 `(&00) WARNING: UPDATE PASSWORD IGNORED`

Meaning
Passwords must not be altered when creating a LEASY file. The input is ignored.
(&00): MODE = NEW or MODE = CATALOG.

LEA2103 `(&00): DELETE OF SECONDARY INDICES NOT ALLOWED`

Meaning
Secondary indices cannot be deleted when creating a LEASY file.
(&00): MODE = NEW or MODE = CATALOG.

Response
Omit deletion request and repeat input.

LEA2104 `(&00): NOT ALLOWED DMS PARAMETERS FOUND`

Meaning
(&00): MODE = UPDATE or MODE = CATALOG:
The DMS attributes of an existing LEASY file
must not be updated (exception: SHARE operand).
LEASY-FILETYPE = F: The DMS attributes are defined in a
ADD-FILE-LINK command (exception: FCBTYPE and BLKCTRL).
LEASY-FILETYPE = M: The DMS attributes are transferred from
the model (exception: SHARE operand).

Response
Omit DMS parameter and repeat input.

LEA2105 `MODE = UPDATE: PARAMETER COMBINATION NOT ALLOWED`

Meaning
The specified statement combination is not permitted when updating the attributes of a LEASY file

Response
Use a different statement combination.

LEA2106 `TYPE = NEW OR TYPE = ERASE: CHANGE OF PASSWORD NOT ALLOWED`

Meaning
Passwords must not be altered when creating or erasing a LEASY directory

Response
Correct error and repeat input.

LEA2107 `FCBTYPE = SAM OR LEASY FILETYPE = F: SECONDARY INDICES ARE NOT ALLOWED`

Meaning
Secondary indices are not permitted for SAM files or LEASY foreign files.

LEA2108 `MORE THAN 255 SECONDARY INDICES ARE NOT ALLOWED`

Meaning
The maximum number of secondary indices permitted per LEASY file is 255.

LEA2109 `SECONDARY INDEX (&00): DUPLICATE KEYNAMES NOT ALLOWED`

Meaning
The same secondary index name was specified more than once in a file specification.

Response
Delete duplicate names and repeat input.

LEA2110 `SECONDARY INDICES MUST NOT BE DEFINED FOR AN INSTANCE OF A MODEL GROUP`

Meaning
Secondary keys must not be specified for an instance of a model group; those defined for the model apply.

Response
Omit secondary indices and repeat input if necessary.

LEA2111 `FILENAME INVALID FOR LEASY FILE TYPE (&00)`

Meaning
The file name is invalid for the LEASY file type (&00).

Response
Correct file name and repeat input.

LEA2112 `WRONG PASSWORD`

Meaning
Password does not tally with information stored.

Response
Correct password and repeat input.

LEA2113 `NO FILE FOUND`

Meaning
None of the files specified in an information request could be found.

LEA2114 `SECONDARY INDICES: TOO MANY RECORD TYPES DEFINED`

Meaning
The space reserved for the record type definition is insufficient. Approximately 8 to 16 Kbytes are available for record type definitions in each file entry, depending on the number of secondary index definitions.

Response
Check record types and reduce number.

LEA2115 `ERROR: RECSIZE GREATER THAN BLKSIZE ALLOWES`

Meaning
The record length greater than the block length allows.

Response
Correct error and repeat input.

LEA2116 `NO M-FILE MODEL DEFINED`

Meaning
No instance can be defined for a non-model file.

LEA2117 `SECONDARY INDEX: MAXIMUM LENGTH OF RECORD TYPES GREATER THAN LENGTH OF RECORD TYPE FIELD`

Meaning
A record type is longer than the record type field.

Response
Shorten the record types that are too long and repeat input.

LEA2118 `LEASY DIRECTORY NOT EMPTY. DO YOU REALLY WANT TO ERASE NON EMPTY LEASY DIRECTORY (&00)? (Y=YES; N=NO)`

Meaning
The user wishes to delete the LEASY directory (&00) that still contains file entries.

Response
Input "Y" if the directory is to be deleted.
Input "N" if the catalog is not to be deleted.

LEA2119 `FILE (&00) ALREADY CATALOGED IN LEASY DIRECTORY`

Meaning
There is already an entry in the LEASY directory for the file (&00)

Response
Repeat input if necessary using the operand MODE = U.

LEA2120 `ERASE M-FILE MODEL NOT ALLOWED: INSTANCES ARE STILL CATALOGED`

Meaning
The model file in a model file group may only be erased when there are no more instances left in this group.

Response
Delete instances explicitly.

LEA2121 `FCBTYPE = PAM DEMANDS STANDARD BLKSIZE`

Meaning
Not standard blocking is not permitted for disk files.

Response
Correct error and repeat input.

LEA2122 `KEYLEN (SI-FILE) > 255`

Meaning
The sum of (1 + greatest SI length + length of primary index) exceeds 255.

Response
Reduce key length and repeat input.

LEA2123 `MAX. ((SI-KEYPOS) + (SI-KEYLEN)) GREATER THAN MAX. RECSIZE OF PRIMARY FILE`

Meaning
The secondary key is greater than the primary record.

Response
Correct error and repeat input.

LEA2124 `SECONDARY INDEX: MAXIMUM KEYLENGTH OF NEW SI-KEY > THAN MAXIMUM OF EXISTING KEY`

Meaning
An attempt has been made to add a secondary key longer than the longest existing key.

Response
If necessary, delete all existing keys and re-enter them.

LEA2125 `RECORDTYPE FIELD UPDATE NOT ALLOWED WHILE POINTERS FOR SI NAME (&00) STILL EXISTING`

Meaning
Updating the record type field is prohibited so long as pointers exist in the SI file for the key (&00).

Response
Delete any pointers which may exist using LEASY-LOADSI; then update the entry in the LEASY catalog.

LEA2126 `RECORDTYPE FIELD UPDATE NOT ALLOWED WHILE POINTERS FOR SI NAME (&00) IN INSTANCE (&01) FOR THIS M-FILE STILL EXISTING`

Meaning
Updating the record type field is prohibited so long as pointers exist in the SI file of the instance (&01) for the key (&00) of the model file group.

Response
Delete any pointers in the instance (&01) which may exist using LEASY-LOADSI; then update the entry in the LEASY directory.

LEA2127 `SECONDARY INDEX: DELETE SI NOT POSSIBLE. SI-NAME (&00) NOT IN LEASY DIRECTORY`

Meaning
The requested secondary index (&00) is not entered in the LEASY directory for this file.

LEA2128 `RECORDTYPE FIELD UPDATE NOT ALLOWED WHILE LEASY APPLICATION ACTIVE`

Meaning
Updating the record type field is forbidden during an ongoing session with the specified LEASY directory.

Response
Wait until the end of the current session.

LEA2129 `MORE THAN 512 SI KEYPARTS FOR ONE FILE ARE NOT ALLOWED`

Meaning
The maximum total number of key parts for all secondary indices is 512 per primary file.

Response
Reduce the number of key parts.

LEA2130 `MODE = CATALOG DEMANDS LEASYTYPE = S`

Meaning
Incorporation of already existing DMS files in the LEASY directory is restricted to master files.

Response
Correct error and repeat input.

LEA2131 `NAME PARAMETER DEMANDS LEASYTYPE = S`

Meaning
Assignment of arbitrary file names is restricted to LEASY master files.

Response
Correct error and repeat input.

LEA2132 `FCBTYPE = DAM ONLY FOR PAM-FILES`

Meaning
A DAM file to be imported (*FILE ..., MOD = C) is not an UPAM file.

LEA2133 `SECONDARY INDEX (&00): RECORD TYPE (&01)(&02)' ALREADY SPECIFIED`

Meaning
The record type (&01) has already been specified for the secondary index (&00) or is already contained in another record type field group value.

Response
Omit record type (&01) and repeat input.

LEA2134 SECONDARY INDEX: RECORD TYPES NOT ALLOWED WITHOUT RECORD TYPE FIELD DEFINITION

Meaning
Input of record type definitions is not permitted if a record type is not defined.

Response
Repeat input with a record type field definition.

LEA2135 RECORD TYPE FIELD DEFINITION NOT ALLOWED FOR AN INSTANCE OF A MODEL GROUP

Meaning
A record type field definition is not allowed for instances of a model file group. The record type field of the model is valid for all instances.

Response
Repeat input without record type field.

LEA2136 RECORD TYPE FIELD NOT WITHIN RECORD

Meaning
The record type field is not wholly within the record.

Response
Correct record type field and repeat input.

LEA2137 FOR FILE (&00)

LEA2138 SECONDARY INDEX: SI UPDATE NOT ALLOWED WHILE POINTERS FOR SI-NAME (&00) STILL EXISTING

Meaning
Updating the secondary index is prohibited so long as pointers exist in the SI file for the key (&00).

Response
Delete any pointers which may exist using LEASY-LOADSI; then update the entry in the LEASY directory.

LEA2139 SECONDARY INDEX: SI UPDATE NOT ALLOWED WHILE POINTERS FOR SI-NAME (&00) IN INSTANCE (&01) FOR THIS M-FILE STILL EXISTING

Meaning
Updating the secondary index is prohibited so long as pointers exist in the SI file of the instance (&01) for the key (&00) of the model file group.

Response
Delete any pointers in the instance (&01) which may exist using LEASY- LOADSI; then update the entry in the LEASY directory.

LEA2140 SECONDARY INDEX: SI UPDATE NOT ALLOWED WHILE LEASY APPLICATION ACTIVE

Meaning
Updating the secondary index is forbidden during an ongoing session with the specified LEASY directory.

Response
Wait until the end of the current session.

LEA2141 SECONDARY INDEX: SI DELETE NOT ALLOWED WHILE POINTERS FOR SI-NAME (&00) STILL EXISTING

Meaning
Deleting the secondary index is prohibited so long as pointers exist in the SI file for the key (&00).

Response
Delete any pointers which may exist using LEASY-LOADSI; then delete the entry in the LEASY directory.

LEA2142 SECONDARY INDEX: SI DELETE NOT ALLOWED WHILE POINTERS FOR SI-NAME (&00) IN INSTANCE (&01) FOR THIS M-FILE STILL EXISTING

Meaning
Deleting the secondary index is prohibited so long as pointers exist in the SI file of the instance (&01) for the key (&00) of the model file group.

Response
Delete any pointers in the instance (&01) which may exist using LEASY-LOADSI; then delete the entry in the LEASY directory.

LEA2143 SECONDARY INDEX: SI DELETE NOT ALLOWED WHILE LEASY APPLICATION ACTIVE

Meaning
Deleting the secondary index is forbidden during an ongoing session with the specified LEASY directory.

Response
Wait until the end of the current session.

LEA2144 THE LINKNAME (FOR POOL/SIPOOL) HAS MORE THAN 8 CHARACTERS.

Meaning
The link name may only have max. 8 characters.

Response
Correct link name and try again.

LEA2145 THE FIRST SIGN OF THE LINKNAME IS NOT A CHARACTER.

Response
Correct link name and try again.

LEA2146 WARNING! LINKNAMES (FOR POOL/SIPOOL) ARE IGNORED.

Meaning
Link names were specified despite BLKCTRL=PAMKEY (default for ISAM-OLD). These link names are not processed.

LEA2147 WARNING! POOLPARAMETERS ARE IGNORED.

Meaning
Pool parameters are only processed in case of ISAM files

LEA2148 DMS-ERROR (&00) WITH POOL (&01).

Meaning
DMS error (&00) has occurred during processing of pool (&01).

Response
identify error with aid of the "System Messages" manual.

LEA2149 POOL (&00) ALREADY EXISTS.

Meaning
The pool to be created already has an entry in the
LEASY-catalog.

LEA2150 POOL (&00) IS NOT IN CATALOG.

Meaning
The pool which should be corrected (or deleted) has no entry in the LEASY catalog.

LEA2151 WARNING: (&00) IGNORES (&01)

Meaning
(&00): MOD=U, MOD=C, LEASY-FILETYPE=M or LEASY-FILETYPE=F
(&01): RECFORM, BLKCTRL, BLKSIZE

LEA2152 (MODE=LINK) WARNING: DMS-PARAMETERS ARE IGNORED

Meaning
When MODE=LINK, the DMS parameters are specified in DMS commands. DMS parameters specified in the *FIL statement are ignored.

LEA2153 SHARUPD OR DUPEKY NOT ALLOWED FOR LEASY-FILES

Meaning
LEASY rejects files with SHARUPD=YES or DUPEKY=YES.

LEA2154 POOL-LINKNAME: INCONSISTENCY BETWEEN *FIL AND DMS-COMMANDS

Meaning
Two different pool link names were specified in DMS and in the *FIL statement.

Response
Remove one of the two link names.

LEA2155 `LEASYTYPE = T/M DEMANDS STANDARD FILENAMES`

Meaning
The file name that was specified in DMS does not correspond to the LEASY name structure "catalogname.linkname".

Response
Reissue the DMS commands with standard file names.

LEA2156 `INCONSISTENCY BETWEEN FILENAMES IN DMS AND *FIL`

Meaning
The file name that was stored in DMS does not correspond to the specification in the NAME operand of the *FIL statement.

Response
Correct DMS commands or NAME operand.

LEA2157 `MODE=LINK IS NOT ALLOWED FOR FOREIGN FILES AND INSTANCES OF A MODEL`

Meaning
MODE=NEW must be used for new foreign files and instances of models.

Response
Reissue the command with MODE=NEW.

LEA2158 `MODE=LINK: THE CHARACTERISTICS OF FILE (&00) ARE NOT DEFINED`

Meaning
With MODE=LINK, DMS commands that define the file attributes must have been issued beforehand. The link name must correspond to the logical file name in the *FIL statement.
(&00): logical file name = link name

LEA2159 `WARNING: FCBTYPE = SAM OR LEASY-FILETYPE = F IGNORES SIOPERF`

Meaning
With FCBTYPE = SAM and LEASY-FILETYPE = F, secondary indices are not allowed. Therefore, no I/O performance parameter can be specified for the SI file.

LEA2160 `WARNING: FCBTYPE = SAM OR LEASY-FILETYPE = F IGNORES RTF`

Meaning
Secondary keys are not permitted with FCBTYPE = SAM and LEASY-DATEITYP = F. As a result, no record type field can be specified.

LEA2161 `WARNING: LENGTH OF FILENAME GREATER THAN 54 CHARACTERS`

Meaning
Changing of filename entries in LEASY directory by using OLDL/NEWL causes a filename length greater than 54 characters. Conversion not processed, further information given by SYSLST-report.

Response
First delete filename entry in LEASY-directory by using *ERA-instruction, add file with correct length using the *FIL-instruction afterwards.

LEA2201 `SI NOT POSSIBLE FOR PRIMARY FILE WITH LEASY FILE TYPE "(&00)"`

Meaning
The specified file is a LEASY TYPE (&00) file. No secondary indices are possible for this LEASY type.

LEA2202 `SECONDARY INDEX (&00) NOT IN LEASY DIRECTORY`

Meaning
The specified secondary index name (&00) is not in the LEASY directory.

Response
Wait for prompt.

LEA2203 `SECONDARY INDEX (&00) ALREADY IN SECONDARY INDEX FILE EXPANDED`

Meaning
The secondary index (&00) given in an ADD specification already exists.

Response
Wait for prompt.

LEA2204 `WARNING: (&00) SI POINTERS ARE NOT GENERATED. SOME PRIMARY RECORDS SHORTER THAN HIGHEST SI KEY POSITION.`

Meaning
(&00) SI pointers are not generated for a primary file with variable length records as one or more secondary keys are outside the key length limits.
Secondary keys cannot be used to access such short records.

LEA2205 `FOR SECONDARY INDEX (&00) NO POINTERS IN SI FILE`

Meaning
The secondary key (&01) has been specified in DELETE but there are no pointers for this key in the SI file.

Response
Wait for prompt.

LEA2206 DUPLICATE SECONDARY KEYS ARE NOT ALLOWED BUT FOUND FOR SI (&00). ONLY THE FIRST SI POINTER IS GENERATED

Meaning
The definition of the secondary key (&00) in the LEASY directory requires that primary records do not contain duplicated secondary keys; duplicates have however been found. At most the first 5 primary key pointers are displayed. The subsequent message LEA2207 gives the total number of duplicates found.

LEA2207 (&00) DUPLICATE KEYS FOUND

Meaning
Duplicates not permitted for a secondary key have nevertheless been discovered. Details of the secondary index concerned are given in the previous message LEA2206. The total number of primary record pointers found for a particular secondary key value is (&00).

Response
Correct SI definition in LEASY catalog or primary file and repeat LEASY-LOADSI.

LEA2208 REQUESTED PADDING IMPOSSIBLE TO REALIZE

Meaning
LEASY-LOADSI is attempting to reserve space in the SI records in accordance with the PAD entry in the file specification, but the remaining available space is smaller than a single SI record.

Response
Wait for prompt.

LEA2209 DUPLICATE PRIMARY KEYS ARE NOT ALLOWED BUT MORE THAN ONE RECORD FOUND FOR KEY (&00)

Meaning
More than one record with primary key (&00) has been found in the primary file. Only one pointer has been set up for the key.

Response
Remove duplicate records from the primary file.

LEA2210 SECONDARY KEY = (&00)

Meaning
Second part of LEA2206.

LEA2211 PRIMARY KEY = (&00)

Meaning
Third part of LEA2206.

LEA2301 `COMMON MEMORY ALREADY EXISTS`

Meaning
The common memory CMMAIN cannot be initialized because one already exists for the LEASY directory specified.

LEA2302 `MEMORY OVERFLOW. INCREASE *MEM VALUE!`

Meaning
The number of files (*FIL) and/or transactions (*TRA) exceeds the value specified or preset in *MEM.

Response
Repeat run with higher *MEMORY specification.

LEA2303 `WARNING: START = WARM, BUT COLD START EXECUTED`

Meaning
A warm start has been requested, but a cold start executed because no BIM files exist or all BIM files empty.

LEA2304 `START = COLD, BUT BIM FILE (&00) NOT EMPTY!`

Meaning
A cold start has been requested but the BIM file (&00) is not empty.

Response
Execute warm start.

LEA2305 `FILE (&00) NOT IN COMMON MEMORY DEFINED`

Meaning
The file (&00) to be warm-started cannot be found in the common memory CMMAIN.

Response
Check using LEASY-CATALOG to ensure that the file (&00) is still defined in the LEASY directory and that its transfer to the common memory has obstructed by a lock (LOCK operand in the *FIL statement).

LEA2306 `WARNING: FILE (&00) LOCKED AND NOT AVAILABLE`

Meaning
The file (&00) has been locked by the LEASY-CATALOG utility routine and is not available.

LEA2307 `WARNING: "START = WARM, RESET", BUT NO PRELIMINARY ENDED TRANSACTIONS FOUND`

Meaning
WARM, RESET warm start option has been requested but no suspended transaction exists.

LEA2308 `ASPACE CONTRADICTS DEVICE`

Meaning
Either ASPACE = TAPE was entered and the AIM file is on disk or a storage allocation for disk was entered and the AIM file is on tape.

Response
Correct input and start MAINTASK again.

LEA2309 `WARNING: ENTERED CATID (&00) NOT EQUAL CATID (&01) BY (&02) FILE.`

Meaning
The catid (&00) entered as MAINTASK operand for (&02) file is not equal to the (&01) catid saved in the LEASY directory.
The given catid (&00) is used.

LEA2310 `TO DECREASE THE NUMBER OF GENERATIONS IS NOT POSSIBLE. GENERATION(S) NOT FREE.`

Meaning
It is not possible to decrease the number of AIM file generations, because the oldest generations are not yet reconstructed on the shadow files and therefore are not free.

LEA2311 `START = (&00) BUT CANNOT BE DONE BY LEASY-RECONST.`

Meaning
START = (&00) is requested using the automatic reconstruction of shadow files. This is not executed by LEASY-RECONST.

Response
Look at messages of LEASY-RECONST and try it again.

LEA2312 `NORMAL TERMINATION NOT POSSIBLE: NOT CLOSED TRANSACTIONS.`

Meaning
A LEASY session with AIM-writing by MAINTASK was requested to close with the CLOS command. But the reconstruction of all AIM generations is not possible because a transaction is opened and not closed between 2 hours after CLOS was entered.

Response
Start new session with a warm start (STA=W).

LEA2313 `ERROR: REN COMMAND MISSING OR INVALID.`

Meaning
A PASSWORD command is not possible before a RENTER command is given.

Response
Enter commands in the sequence REN=..., PAS=...

LEA2314 `WARNING: NOT FREE AIM GENERATIONS ERASED.`

Meaning
While comparing the AIM control record with the TSOSCAT entry it was detected that not free generations are erased. The shadow files are maybe not at the same level as the original files.

Response
If necessary terminate the session and bring the application in consistent state.

LEA2315 `WARNING: THIS OPERATING SYSTEM VERSION DOES NOT SUPPORT NKISAM FILES`

Meaning
There are NKISAM files (BLKCTRL<>PAMKEY) in the LEASY catalog, but they are not supported by this operating system version.

Response
It is not possible to work with the NKISAM files in this session.

LEA2316 `MAINTASK UNABLE TO START. GENERATION(S) NOT FREE`

Meaning
Automatic reconstruction of shadow files is used, but LEASY-RECONST is unable to reconstruct the oldest generation on the shadow files, so AIM file generations are not free.

Response
Look at messages of LEASY-RECONST and try it again.

LEA2317 `AIM-BUFFER TO SMALL.`

Meaning
Because of the large number of files that can be opened simultaneously, the OPEN element of the AIM record has already used up more than half of the AIM buffer.

Response
Increase the value for the AIM buffer in the MAINTASK by means of an AIB command.

LEA2318 `ERROR: PAS COMMAND NOT POSSIBLE.`

Meaning
It is not possible to specify a password because the RECONST ENTER file specified in the REN statement already existed and therefore LEASY does not change this ENTER file.

LEA2319 `WARNING: NO ACTIONS, SESSION WAS ALREADY CONSISTENT`

Meaning
Only empty BIM-files could be found.

LEA2320 `WARNING: STA=W IS SET BECAUSE OF USE=C`

Meaning
Cold start is not possible with USE=C. LEASY assumes STA=W.

LEA2401 `LEASY MAINTASK NOT READY OR AVAILABLE FOR RECONSTRUCTION`

Meaning
The reconstruction program cannot be executed because the LEASY MAINTASK has not, or has not yet, been started with the statement USE = R, or because it has not yet entered the wait state.

Response
Either wait for the wait state or start LEASY-MAINTASK, specifying the USE = R statement, and repeat the reconstruction run.

LEA2402 `*CAT COMMAND MISSING OR INVALID`

Meaning
In the *FIL statement, or after the statement input (*END statement), it has been discovered that the *CAT statement is either missing or invalid in the statement sequence.

Response
Correct input.

LEA2403 `PARAMETER MISSING`

Meaning
A statement must contain at least one parameter input.

Response
Correct input.

LEA2404 `*REP COMMAND CONTRADICTS *MOD PRINT = MINIMUM`

Meaning
If the *REPORT statement is used, the *MODE statement with PRINT = NORMAL is required.

Response
Correct input.

LEA2405 `(&00) WAS CHANGED TO (&01)`

Meaning
The operand (&00) was changed to (&01).

LEA2406 `ERROR IN FILE FILTER WITH FILE (&00). DUPLICATE FILENAMES ARE NOT ALLOWED.`

Meaning
File name (&00) is duplicated in the file filter selection.

Response
Correct input.

LEA2407 ERROR WITH RECORD SELECTION ELEMENT "(&00)": DUPLICATE RECORD SELECTIONS ARE NOT ALLOWED.

Meaning
The AIM record type (&00) has been specified more than once.

Response
Correct input.

LEA2408 ERROR WITH EXTRACTION ELEMENT EXT = ((&00), (&01)): DUPLICATE RECORD EXTRACTIONS ARE NOT ALLOWED.

Meaning
The record extraction element with position (&00) and length (&01) has been specified more than once.

Response
Correct input.

LEA2409 RECORD EXTRACTION DEMANDS LENGTH = ALL

Meaning
The record extract entry requires a length value ALL.

Response
Correct input.

LEA2410 SIUPDATE = YES CONTRADICTS UPDATE = NO

Meaning
SI files cannot be reconstructed without reconstructing the corresponding primary files.

Response
Correct input.

LEA2411 (&00) CONTRADICTS (&01)

Meaning
(&00) and (&01) contradicts one another. (&00) must be >= (&01).

Response
Correct input.

LEA2412 LASTTRANSACTION PARAMETER DEMANDS TOSESSION PARAMETER

Meaning
Use of the LAS parameter necessitates use of the TOS parameter.

Response
Correct input.

LEA2413 (&00) WARNING: FOR COMMAND (&01) FOLLOWING OPERANDS ARE CHANGED:

Meaning
The statement (&00) forces in the statement (&01) the change of operands as specified in the following message.

LEA2414 (&00): USE OF FILE FILTER, SESSION FILTER, RANGE FILTER AND DATE FILTER IS NOT ALLOWED.

Meaning
The *FIL, *SES, *RAN and *DAT statements are not allowed in conjunction with (&00).

LEA2415 GENERATION PARAMETER FORBIDS FROMGENERATION AND TOGENERATION PARAMETER

Meaning
The GEN parameter must not be used at the same time as the FROMGEN or TOGEN parameters in the *CAT statement.

LEA2416 *MOD TRANSACTIONS = (&00): RANGE OF AIM GENERATION DOES NOT INCLUDE LAST AIM GENERATION

Meaning
The last AIM generation must not be excluded from the generation number selection in the *CAT statement with *MOD TRANSACTIONS = CLOSED.

LEA2417 *MOD TRANSACTIONS = CLOSED: NO RECONSTRUCTION ALLOWED WHILE LEASY APPLICATION ACTIVE

Meaning
No common memory of the affiliated directory can exist for a reconstruction run with TRANSACTIONS = CLOSED.

LEA2418 *MOD TRANSACTIONS = CLOSED: NOT POSSIBLE TO REALIZE BECAUSE AIM FILE ON TAPE

Meaning
If AIM file on tape it is not possible to mark records of not closed transactions unguilty.

LEA2419 *MOD UPDATE = YES: PARAMETER RANGE NOT ALLOWED

Meaning
The RANGE filter is only allowed when LEASY-RECONST is used to show the entry of an AIM file.

Response
Either omit RANGE parameter or use *MOD UPD = N.

LEA2420 `COLD START NOT POSSIBLE BECAUSE AIM FILE GENERATION (&00) IN STATE (&01).`

Meaning
In the mode automatic reconstruction of shadow files is a cold start not possible because the aim file generation (&00) is in the state (&01) and (maybe) contains opened transactions.
(&01) = GENINUSE : The generation was last written to.
(&01) = GENWAIT : The generation was waiting for the end of all transactions.
(&01) = GENSWIT : The generation was being switched.

Response
Execute warm start.

LEA2421 `LEASY MAINTASK NOT SUITABLE FOR AUTOMATICALLY RECONSTRUCTION`

Meaning
The reconstruction program cannot be executed because the LEASY MAINTASK has not been started with the statement AUT=Y.

Response
Either start LEASY-MAINTASK again with AUT=Y or repeat the reconstruction run without automatically reconstruction.

LEA2422 `WARNING: COMMAND (&00) IGNORED BECAUSE OF AUTOMATIC RECONSTRUCTION`

Meaning
The command (&00) is not allowed with automatically reconstruction.
The command will be ignored.

Response
E

LEA2423 `WARNING: PARAMETER (&00) IN (&01) COMMAND IGNORED BECAUSE OF AUTOMATIC RECONST`

Meaning
The parameter (&00) in (&01) command is not allowed with automatically reconstruction.
The parameter will be ignored.

LEA2424 `WARNING: PARAMETER(S) IN (&00) COMMAND CHANGED BECAUSE OF AUTOMATIC RECONST:`

Meaning
Because of automatically reconstruction parameters in the (&00) command will be changed.
The message LEA2405 will show the changed parameter(s).

LEA2425 `NO AIM FILE GENERATION FOR RECONSTRUCTION FOUND`

Meaning
There is no AIM-FG found, that can be used for reconstruction.
Possible reasons:

1) TO generation number less than FROM generation number
2) The only AIM-FG is in the USE state and the MAINTASK is running normally (USE=N).

Response
Correct input.

LEA2501 `NO CMMAIN FOR THIS LEASY DIRECTORY IN SYSTEM`

Meaning
The selected common memory CMMAIN is not available or is no longer available.

LEA2502 `TRANSACTION (&00) IS ALREADY LOCKED`

Meaning
The requested lock has already been set for the transaction (&00).

LEA2503 `TRANSACTION IS LOCKED PERMANENTLY AND CANNOT SET UNLOCKED`

Meaning
The transaction cannot be unlocked for reasons relating to the system.

LEA2504 `QUIET COMMAND IS NOT ACTIVE`

Meaning
The system has to be put in the QUIET state by the QUIET function before the AIMC/AIMW function is used.

LEA2505 `THERE ARE ACTIVE TRANSACTIONS`

Meaning
The AIMC function is only accepted when active transactions no longer exist.

LEA2506 `OWN TASK IS NOT MAIN MASTER TASK: UPDATE ON CMMAIN IS NOT ALLOWED`

Meaning
The user is not authorized to alter the CMMAIN.

LEA2507 `AIM FILE GENERATION SWITCHING NOT POSSIBLE. MAIN TASK NOT/NO LONGER ACTIVE OR STARTED WITHOUT AIM SAVING`

Meaning
Further possible error:
MAINTASK is started by means of the operand USE=R and the reconstruction is not yet finished.

Response
Check, whether the MAINTASK is active and has been started with the correct operands.

`LEA2508     AIM FILE GENERATION SWITCHING IS ALREADY IN ACTION`

Meaning
AIM file switching is rejected as long as an earlier switching routine is still active.

`LEA2509     REQUESTED TRANSACTION(S) NOT FOUND`

Meaning
The requested transaction cannot be found in the transaction table.

`LEA2510     INVALID SELECTION COMBINATION`

Meaning
Invalid combination of specifications for the transaction identifier (e.g.: *,*,*).

`LEA2511     FILE (&00) IS ALREADY LOCKED`

Meaning
The requested lock has already been set for the file (&00).

`LEA2512     FILE (&00) IS MODEL OF MODEL GROUP`

Meaning
The file (&00) is the model of a model file group; the function (&00) (message LEA2513) cannot be set for this reason. The selected function is valid only for instances of model file groups.

`LEA2513     AND CANNOT BE LOCKED`

Meaning
Second part of LEA2512.

`LEA2514     FILE (&00) IS NOT M-FILE`

Meaning
The selected file(s) is (are) not model file instance(s). The suffix specification is invalid.

`LEA2515     REQUESTED TASK SEQUENCE NUMBER NOT FOUND IN TASK TABLE`

```
LEA2516     OWN TASK IS NOT MAIN MASTER TASK: ERASE OF AN AIM FILE GENERATION RESP. IOTASK
            IS NOT ALLOWED
```

Meaning
The user is not authorized to erase AIM file generations or I/O tasks.

`LEA2517     TRANSACTION (&00) IS ALREADY SET TO ROLLBACK`

Meaning
The requested rollback has already been set for the transaction (&00).

`LEA2518     TRANSACTION (&00) IS ALREADY LOCKED OR SET TO ROLLBACK`

Meaning
The requested rollback or lock has already been set for the transaction (&00).

`LEA2519    FILE (&00) IS ALREADY UNLOCKED`

Meaning
The requested unlock has already been set for the file (&00).

`LEA2520    AND CANNOT BE UNLOCKED`

Meaning
SECOND PART OF LEA2512.

`LEA2521    TRANSACTION (&00) IS ALREADY UNLOCKED`

Meaning
The requested unlock has already been set for the transaction (&00).

`LEA2522    NO AIM FILE EXISTS`

Meaning
There is no AIM file generation group relating to the selected LEASY directory. Therefore deleting of file is not possible.

`LEA2523    NO MEMORY POOL FOR IO TASKS IN SYSTEM`

Meaning
A memory pool for IO tasks accordingly to the selected CMMAIN is not available or is no longer available.

`LEA2524    NO IO TASK ACTIV`

Meaning
The command IOTE was entered. But there is no active IO-Task.

`LEA2525    ERASE NOT POSSIBLE. GENERATION(S) NOT FREE.`

Meaning
LEASY is running with automatic reconstruction of shadow files.
The specified generation(s) can not be erased because it (they) is (are) not free.

Response
Use function AIMS to detect which generations are free.

`LEA2526    DELETE NOT POSSIBLE. MAINTASK DOES NOT EXIST OR NOT READY`

Meaning
AIM file generation can only be erased, if the MAINTASK is still active.
Further possible error:
The MAINTASK has been started my means of the operand USE=R and the reconstruction has not yet finished.

Response
Start the MAINTASK or finish the reconstruction and try again the erase attempt.

LEA2527 `TERM NOT POSSIBLE, BECAUSE OF AIM-WRITING BY MAINTASK.`

Meaning
Because AIM writing is performed by the MAINTASK, it is not possible to cancel the MAINTASK with TERM.

Response
Use CLOS or SHUT to terminate the MAINTASK.

LEA2528 `OWN TASK NOT MAIN MASTER TASK. INPUT IN RETURN CODE TABLE NOT ALLOWED`

Meaning
The user is not authorized to enter a return code into the dump return code table. Only the main master task is allowed to enter a return code.

LEA2529 `MAXIMUM NUMBER OF RETURN CODES ALREADY REACHED`

Meaning
It is not possible to add a return code, because the table is already full.

LEA2530 `SPECIFIED RETURN CODE UNKNOWN`

Meaning
The specified return code is not known to LEASY.

LEA2531 `SPECIFIED RETURN CODE ALREADY IN TABLE`

Meaning
Addition of this return code is not possible, because it was already entered in this session.

LEA2532 `SPECIFIED RETURN CODE NOT FOUND IN TABLE`

Meaning
Delete not possible, because return code cannot be found in return code table.

LEA2533 `NO RETURN CODE IN TABLE`

Meaning
The desired function (display or delete) cannot be executed.

LEA2534 `AN ERROR OCCURRED DURING ROLLBACK PROCESSING OF TRANSACTION (&00)`

Meaning
(&00) Session specific Transaction Number

LEA2535 RETURNCODE (&00) WAS ENTERED TO THE RETURNCODE-TABLE

LEA2536 RETURNCODE (&00) WAS ERASED FROM THE RETURNCODE-TABLE

LEA2537 FOR THIS RETURNCODE NO DUMP CAN BE GENERATED

LEA2538 LIST OF LOCK ELEMENTS IS LOCKED BY RUNTIME SYSTEM

Meaning
The lock elements can not be shown, because a new lock element is added or an existing lock element is deleted by the runtime system in this time.

Response
Select function SHLE once more.

LEA2539 ROLLBACK ERROR (&00) WITH FILE (&01)

Meaning
(&00) Rollback error code has the same meaning as the LEASY runitme error code
(&01) filename

LEA2540 FILE (&00) NOT DEFINED IN COMMON MEMORY

Meaning
File (&00) was not found in the common memory

LEA2541 ROLLBACK NOT POSSIBLE. TRANSACTION (&00) WITHOUT BIM-LOGGING

Meaning
Rollback for the transaction (&00) is not possible, because the transaction is used without BIM-Logging.
(&00) Session specific transaction number

Response
Define transaction with BIM-Logging

LEA2601 LEASY FILE TYP F CAN'T BE SAVED, BECAUSE FILE (&00) IS NOT SPECIFIED

Meaning
File (&00) of LEASY type F cannot be saved as no DMS file has been assigned to it by means of a ADD-FILE-LINK command.

LEA2602 SAVE ON DISK REQUESTS SPECIFICATION OF DISK

Meaning
Disk has been specified as the storage medium, but no VSN has been specified.

Response
Wait for prompt.

LEA2603 `SAVE ON DISK REQUESTS SPECIFICATION OF DEVICE`

Meaning
Disk has been specified as the storage medium, but no device was specified.

Response
Wait for prompt.

LEA2604 `SAVE OF A REMOTE DIRECTORY OR A DIRECTORY OF A FOREIGN USERID IS NOT ALLOWED`

Response
Wait for prompt.

LEA2605 `LEASY FILE (&00) (&01) CAN ONLY SAVED BY SYSTEM ADMINISTRATOR`

Meaning
LEASY file (&00) with DMS name (&01) can only be saved by a privileged user.

LEA2606 `SAVE OF LEASY FILE (&00) (&01) IS NOT POSSIBLE BY ARCHIVE`

Meaning
The LEASY file (&00) with DMS name (&01) cannot be saved by the BS2000 ARCHIVE utility routine.

LEA2607 `SAVE OF LEASY SHADOW FILES IS NOT POSSIBLE WITH (&00) PARAMETER (&01)`

Meaning
The LEASY shadow files with parameter (&00) cannot be saved, as (&01) either contains a :catid: or contains a $userid and LEASY-SAVE is not executing under a privileged user ID.

LEA2608 `SAVE NOT ALLOWED, BECAUSE BIM FILE (&00) NOT EMPTY`

Meaning
When checking the BIM files, it has been determined that file (&01) is not empty.

Response
Roll back any open transactions by means of a warm start.

LEA2609 `NO LEASY DIRECTORY OR INCONSISTENCY BETWEEN FILE NAME (&00) AND FOUND DIRECTORY NAME (&01)`

Meaning
File (&00), which was opened on the basis of the name entered in response to the message LEA0604, is not a LEASY directory, or the name found in the identification record of the opened file differs from the DMS name.

LEA2610 `SAVE OF A SECOND DIRECTORY ON TAPE(S) WITH SAME VSN(S) IS NOT ALLOWED`

Meaning
Y has been entered in response to the message LEA0609, although the last LEASY directory was saved to an explicitly specified tape (or tapes).

Response
Wait for prompt.

LEA2701 `ERROR: DIFFERENT INTERNAL VERSIONS`

Meaning
The internal version of an already active IO-Task is different from the own version.

Response
Start IO-Task with equal version.

LEA2702 `ERROR: INITALIZATION OF IO-TASK MEMORY-POOL NOT COMPLETED`

Meaning
Initialization of common memory by another IO-Task is not completed

LEA2703 `DIFFERENT OPERANDS WITH ACTIV IO-TASK`

Meaning
One or more statements in this IO-Task are different from an already

Response
Start IO-Task with equal operands.

LEA2704 `MAXIMUM NUMBER OF IO-TASK ALREADY STARTED`

Meaning
There more IO-Task activated than specified in *IOT statement.

LEA2705 `LEASY MAINTASK FOR DIRECTORY (&00) NOT ACTIV.`

Meaning
The LEASY maintask for directory (&00) is not active in system.

LEA2706 `FILE (&00) NOT IN LEASY DIRECTORY`

Meaning
The file (&00), specified in *OPF statement, cannot be found in LEASY directory.

LEA2707 `ERROR AT LEASY CALL (&00). DB-PARAMETER IS (&01). ERRORCODE OF LEASY RUNTIME SYSTEM IS (&02).`

Meaning
During LEASY Call (&00) an error has occurred. Details are provided by the runtime system error code (&02) and DB operand (&01).

LEA2708 `WARNING: MAXIMUM RECORDSIZE OF FILE (&00) IS (&01) AND IS GREATER THAN ARLEN (&02).`

Meaning
In *ARL statement a moving size from IO-Task to usertask is specified that is (&00), but the file (&00) has a maximum record size of (&02).

LEA2801 `FILE '(&00)' NOT CATALOGED IN LEASY DIRECTORY`

Meaning
File name (&00) not found in present directory.
(&00): file name.

Response
Correct name of file or name of directory and retry

LEA2802 `WARNING: DIRECTORY '(&00)' HAS ALREADY (&01)-FORMAT, NO CONVERSION`

Meaning
The Directory has already the new format, a conversion is not necessary.
(&00): name of directory.
(&01): KEY / NONKEY.

LEA2803 `FILE '(&00)' NOT ON SAME USERID`

Meaning
The file (&00) is not on the userid, where LEASY-CONVERT was started, but this is necessary for a conversion.
(&00): file name.

Response
Start convert program from the same userid.

LEA2804 `WARNING: INCONSISTENCE BETWEEN DIRECTORY AND FILE IN BLKCTRL-VALUE`

Meaning
The value of BLKCTRL is not the same in the directory and the file.

LEA2807 `OVERWRITE OF FILE '(&00)' NOT POSSIBLE`

Meaning
Overwrite of file (&00) is not possible, because in the CONVERT statement operand OVERWRITE is set to NO.
(&00): file name.

LEA2808 `LEASY DIRECTORY NOT SPECIFIED`

Meaning
In FIL-statement a cataloged file is given, but no catalog is assigned previously.

LEA2809 `BLKSIZE=(&00) OF FILE '(&01)' SMALLER THAN MINIMUM SIZE=(&02), NO CONVERSION`

Meaning
The block size of the output file is too small for taking the records of the input file.
(&00): Block size of the output file.
(&01): File name.
(&02): Minimum block size of output file.

Response
Enlarge the block size of the output file.

LEA2810 `WARNING: RECORDS OF INPUT FILE '(&00)' SHORTED`

Meaning
While converting KEY to NONKEY even the maximum block size of the output file is not sufficient to take up the record of the input file in their total length.
(&00): File name.

LEA2811 `BLOCK IN PAM FILE '(&00)' CONTAINS X'FF' AS FIRST CHARACTER, NO CONVERSION`

Meaning
While converting KEY to NONKEY a block is encountered in a PAM file that contains X'FF' as first character. That is the mark for erased blocks in NONKEY format.
(&00): File name.

Response
This PAM file can't be converted by LEASY-CONVERT.

LEA2812 `WARNING: FILE '(&00)' HAS ALREADY (&01)-FORMAT, NO CONVERSION`

Meaning
The file (&00) has already the new format, a conversion is not necessary.
The SI file is not converted too.
(&00): file name
(&01): KEY / NONKEY / 4K-NONKEY

LEA2813 `WARNING: CONVERT-COMMAND IGNORED`

Meaning
A CONVERT statement is only allowed as first statement.

LEA2814 `CONVERSION OF FILE '(&00)' WITH BLKCTRL=NO NOT POSSIBLE`

Meaning
(&00): file name.

LEA2815 `SAME TAPE FOR IN- & OUTPUT-FILE '(&00)' NOT POSSIBLE`

Meaning
(&00): file name.

LEA2816 `ONE RECORD OF FILE '(&00)' > MAX. POSSIBLE RECORD-SIZE (&01)`

Meaning
The maximum block size of the output file is not sufficient to take up the records of the input file in their total length, and shorting of records is not allowed (TRUNCATE = NO).
(&00): file name.
(&01): Maximum possible record-size of output file.

Response
Conversion of file only with record-shorting (TRUNCATE=YES) possible.

LEA2817 `BLOCK SIZE NOT EQUAL TO (STD,2) OR MULTIPLE OF IT`

Meaning
When converting to 4K-NONKEY for the block size of the target file a value not equal to (STD,2) or a multiple of it is explicitly specified or SAME is given and the block size of the source file is not equal to (STD,2) or a multiple of it.

LEA2818 `WARNING: DIRECTION OF CONVERSION CHANGED TO 4K-NONKEY`

Meaning
In the statement CONVERT the operand TO=NONKEY was specified, but the target file is to put on a NK4 pubset.

LEA2819 `WARNING: MAXIMUM RECORD SIZE OF THE INPUT FILE SHORTENED`

Meaning
The maximum record size of the source file was determined implicitly from the block size. The record size of the target file is reduced due to the block size.

LEA3001 `STRUCTURE ERROR ON FILE (&00) IN BLOCK STARTING WITH HALFPAGE NUMBER (&01) AT POSITION (&02)`

Meaning
A structure error has been discovered in file (&00) in the block starting with the PAM block number (&01) at position (&02).

Response
Generate diagnostic documentation and inform system administrator.

LEA3101 `BECAUSE RESET ROUTINE COULD NOT BE EXECUTED LOGICAL INCONSISTENCIES BETWEEN CATALOG FILE AND DATA FILES MAY APPEAR`

Meaning
Due to errors which occurred when updating the LEASY directory, LEASY-CATALOG attempted to restore the old status. As errors also occurred during rollback, the status of the LEASY directory is undefined.

Response
Inform LEASY administrator, and if necessary reconstruct directory from system save file.

LEA3201 `SYSTEMERROR IN MAIN MODULE: SORTEXIT (&00) CALLED "(&01)" COULD NOT BE LOADED`

Meaning
The system cannot be informed of the sort exit module for exit (&00) named (&01) via a table macro.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3203 `SYSTEMERROR IN MAIN MODULE: SORT COULD NOT BE LOADED`

Meaning
The BS2000 SORT could not be linked to the LEASY-LOADSI utility routine.

Response
Check whether the BS2000-SORT is available under $TSOS.

LEA3401 `SYSTEM ERROR IN AIM FILE ACCESS MODULE: INVALID CALL FUNCTION CODE (&00)`

Meaning
Incorrect call parameter (&00) in the read module for the AIM file.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3402 `SYSTEM ERROR IN OPEN MODULE: FILE (&00) NOT FOUND IN COMMON MEMORY`

Meaning
The file (&00), which were required when the error occurred, cannot be found in the common memory.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3403 `SYSTEM ERROR IN (&00) MODULE: REQUESTED FILE ID (&01) NOT FOUND IN FAST TABLE`

Meaning
The file (&01) required at the time of the error cannot be found in an internal table by module (&00).

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3404 `SYSTEM ERROR IN TABLE ACCESS MODULE: UNDEFINED TABLE (&00) REQUESTED`

Meaning
Access to (undefined) table (&00) requested in the table access module.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3405 SYSTEM ERROR IN TABLE ACCESS MODULE: OPEN FUNCTION WITH NULL ELEMENTS REQUESTED

Meaning
When opening a table in the table access module, zero has been specified as the number of elements.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3406 SYSTEM ERROR IN TABLE ACCESS MODULE: ACTUAL TABLE END ADDRESS FOR TABLE (&00) IS IN ERROR

Meaning
An invalid table end address for the table (&00) has been detected in the table access module.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3407 SYSTEM ERROR IN TABLE ACCESS MODULE: DELETE FUNCTION FOR TABLE (&00) REQUESTED

Meaning
The erase of an internal table was requested in the table access module.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3408 SYSTEM ERROR IN TABLE ACCESS MODULE: UNDEFINED TABLE ACCESS FUNCTION (&00) REQUESTED

Meaning
The undefined function (&00) was requested in the table access module.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3409 SYSTEM ERROR IN TABLE ACCESS MODULE: UNDEFINED TABLE ACCESS MODE (&00) REQUESTED

Meaning
The undefined access mode (&00) was requested in the table access module.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3410 SYSTEM ERROR IN AIM FILE OPEN MODULE: TSN (&00) NOT FOUND IN USERID TABLE

Meaning
The TSN (&00) required at the time of the error cannot be found in an internal table.

Response
Generate diagnostic documentation and inform LEASY maintenance.

LEA3411 `ERROR IN AIM FILE: FILENAME (&00) INVALID`

Meaning
Executing an AIM OPEN entry an invalid file name was detected.

Response
Generate diagnostic information and inform LEASY maintenance.

LEA3412 `ERROR IN AIM FILE: RECORD INVALID SHORTENED FOR FILE (&00)`

Response
Generate diagnostic information and inform LEASY maintenance.

LEA3413 `SYSTEM ERROR IN DESHORTENING MODUL: LENGTH PARAMETER IN ERROR`

Response
Generate diagnostic information and inform LEASY maintenance.

LEA3414 `ERROR IN MODULE (&00): (&01) NOT FOUND IN TABLE (&02).`

Meaning
A LEASY internal table is wrong. The displayed search element cannot be found.

Response
Generate diagnostic information and inform LEASY maintenance.

LEA3501 `NOT IMPLEMENTED ENLARGEMENT IN (&00)`

Meaning
The version of the utility routine is not compatible with that of the LEASY-MAINTASK utility routine.

LEA3601 `SYSTEMERROR IN ARCHIVE EXECUTE MODULE: ACTUAL END ADDRESS OF ARCHIVE COMMAND LIST IS IN ERROR`

Meaning
The current end address of the ARCHIVE statement list is to high.

Response
Generate diagnostic information and inform LEASY maintenance.

LEA4001 `**** END OF FILE ON RDATA ****`

Meaning
The EOF condition for SYSDTA has taken effect during input of control data from a file.

Response
Complete input.

LEA4002 `PERMANENT HARDWARE READ ERROR`

Meaning
The RDATA macro has issued error code X"04".

Response
Inform system administrator.

LEA4003 `INPUT DATA RECORD TOO LONG`

Meaning
The RDATA macro has issued error code X"0C".

Response
Shorten input records.

LEA5001 `VERIFY FOR FILE (&00) EXECUTED`
(B) Routing code: U Weight: 30

Meaning
A VERIFY macro was issued when file (&00) was opened.

LEA5002 `SHADOW DIRECTORY NAME: (&00) SHADOW SUFFIX NAME: (&01)`

Meaning
Specifications for generating the name of the shadow files as defined with the program LEASY-CATALOG with *CAT CPC = and CPS =.

LEA5003 `START OF AIM FILE GENERATION SWITCHING ON (&00) AT (&01)`

Meaning
The program section for switching the AIM file generation has been started at (&01) on (&00).

LEA5004 `AIM FILE GENERATION SWITCHING SUCCESSFUL`

Meaning
AIM file generation switching has been carried out correctly.

LEA5005 `AIM FILE GENERATION SWITCHING IN ERROR`
(B) Routing code: U Weight: 70

Meaning
AIM file generation switching has not been carried out correctly.
The error which occurred can be identified from the preceding message.

LEA5006 `UNRECOVERABLE ERROR WITHIN AIM FILE SWITCHING ROUTINE`

(B) Routing code: U Weight: 70

Meaning
An unrecoverable error has occurred within AIM switching routine.
The state of the AIM file generations is unpredictable.

Response
Verify status with /SHOW-FILE-ATTRIBUTES and the LEASY-RECONST utility routine and restore consistency by means of manual intervention.

LEA5007 `ERROR WITH FILE (&00). FILENAME OF SHADOW FILE CANNOT BE BUILD`

Meaning
It is not possible to form a valid shadow file name for file (&00) with the CPC and CPS entries of the *CAT statement in the LEASY-CATALOG routine.

Response
Modify the CPC and CPS entries.

LEA5008 `ERROR WITH SHADOW FILE (&00). FILENAME TOO LONG TO BUILD SI-FILENAME`

Meaning
The name of the shadow file is too long for "-SI" to be added.

Response
Shorten the CPS entry.

LEA5009 `(BUT ONLY WITH "REPAIR = NO")`

Meaning
Second part of LEA5001.

LEA5010 `WARNING: AIM FILE (&00) CREATED WITH ONLY MINIMUM PRIMARY ALLOCATION`

Meaning
While switching to the next AIM file generation a storage shortage was detected. A second switch with minimum primary allocation was successful.

Response
If necessary ensure that there will be enough space by deleting other files or switch to an AIM file generation located on an other disk.

LEA5011 `SWITCH NOT POSSIBLE. NO GENERATION FREE.`

Meaning
LEASY is running with automatic reconstruction. A switch is not possible because there is no free AIM generation.

Response
None.

LEA5101 `LAST SESSION: NUMBER = (&00), DATE = (&01), TIME = (&02).`

Meaning
The user entered INF = Y to request output of information on the last LEASY session. The session last generated was number (&00) and was started on (&01) at (&02) hours.

LEA5102 `****** NO SESSION CREATED ******`

Meaning
No session has yet been generated for the LEASY directory specified (LEASY-MAINTASK has not yet been started).

LEA5103 `MISSING CHARACTERISTICS OF EXISTING FILE. FILE CANNOT BE CATALOGED IN LEASY DIRECTORY`

Meaning
MODE = CATALOG was used to attempt to import a file into the LEASY directory. The TSOS directory does not however contain any file attributes for this file.

Response
Use /FSTAT to check the attributes and correct the error.

LEA5104 `LEASY DIRECTORY NOT ERASED`

Meaning
The user wished to erase a directory with file entries. He responded to LEA2118 with "N" or an invalid entry. The directory was not erased.

LEA5105 `DMS FILENAMES WITH CATID`

Meaning
DMS functions are called with full qualified file names.

LEA5106 `DMS FILENAMES WITHOUT CATID`

Meaning
DMS functions are called with file names without catid.

LEA5201 `SORT ABORTED BECAUSE OF SERIOUS ERROR`

Meaning
Sort run aborted due to error in SORT. The error responsible was output by SORT prior to this message.
Warning: The SI file may already be partially created.

Response
Wait for prompt.

LEA5202 SORT ABORTED BECAUSE OF SERIOUS ERROR WHILE ACCESS TO PRIMARY OR SECONDARY INDEX FILE

Meaning
Sort run aborted due to a DMS error while accessing the primary or SI file or a logical error during SORT exit routines. The error involved was output by the SORT exit routine prior to this message.
Warning: The SI file may be only partially created.

Response
Wait for prompt.

LEA5203 STRUCTUR ERROR ON PRIMARY PAM FILE (&00) IN BLOCK STARTING WITH HALFPAGE NUMBER (&01)

Meaning
A structure error has been detected in the data block beginning with PAM block number (&01) while reading the primary file (&00).

Response
Check the file structure.

LEA5204 LAST PAGE POINTER OF PRIMARY FILE (&00) DOES NOT REFER TO END OF BLOCK

Meaning
The last page pointer has to be a multiple of the block size.
(&00): Name of primary file.

Response
Please check the last page pointer of the specified file and change it to a valid value.

LEA5301 VIRTUAL MEMORY OVERFLOW

Meaning
The virtual memory space is exhausted.

Response
Inform system administrator.

LEA5302 *** WARM/COLD START FINISHED WITH ERROR ***

Meaning
Errors have been detected during a warm or cold start.

Response
Correct error, repeat run.

LEA5303 WARM/COLD START SUCCESSFUL

Meaning
The warm or cold start has been terminated without error.

LEA5304 `*LEASY MAINTASK (&00) INITIALIZATION COMPLETED`

(B) Routing code: U Weight: 30

Meaning

All statements have been processed. The MAINTASK of the directory (&00) is ready for the LEASY runtime system and enters a wait state.

LEA5305 `*LEASY MAINTASK (&00) INITIALIZATION FINISHED WITH ERROR*`

(B) Routing code: U Weight: 70

Meaning

At least one error has been detected in the operand check for the MAINTASK of the directory (&00) during generation of the common memory CMMAIN.
The program is terminated.

Response

Correct error and repeat run.

LEA5306 `*LEASY MAINTASK (&00) READY FOR RECONSTRUCTION*`

(B) Routing code: U Weight: 30

Meaning

All statements have been processed. The common memory CMMAIN is ready for after-image reconstruction and enters a wait state.

LEA5307 `NEW LEASY SESSION CREATED: SESSION NUMBER = (&00), DATE = (&01), TIME = (&02)`

Meaning

The new LEASY session just created has session number (&00) and was created on (&01) at (&02) hours.

LEA5308 `READY FOR PETR HANDLING`

Meaning

A warm start is executed for suspended transactions.

LEA5309 `WARM START WITH PETR HANDLING SUCCESSFUL`

Meaning

A warm start for suspended transactions has been successfully completed.

LEA5310 `WARM START WITH PETR HANDLING FINISHED WITH ERRORS`

Meaning

A warm start for suspended transactions has been terminated with errors.

LEA5311 `AUTOMATIC RECONSTRUCTION ONLY WITH LOG=Y,M`

Meaning

Automatic reconstruction (AUT=Y) is only allowed if AIM records are written by MAINTASK.

Response

Change LOG statement and repeat MAINTASK run.

LEA5312 `AIM-FILE-GENERATION IS NOT IN A CONSISTENT CONDITION.`

Meaning
AIM-file generations are not included in shadow-files (old session was started with automatic reconstruction, new session without).

Response
Start MAINTASK with AUT=Y and REN statement.

LEA5313 `REN-INSTRUCTION MISSING.`

Meaning
AUT=Y requires a following REN statement.

Response
Start MAINTASK with AUT=Y and REN statement.

LEA5314 `USE=R WITH AUT=Y NOT ALLOWED`

Meaning
MAINTASK for reconstruction (USE=R) and automatic reconstruction of shadow files (AUT=Y) not allowed

Response
Start MAINTASK without AUT=Y

LEA5315 `SESSION (&00) FINISHED CONSISTENT`

Meaning
(&00): Session-Number
The Session No. (&00) was finished consistent.

LEA5316 `SESSION (&00) COULD NOT BE FINISHED CONSISTENT`

Meaning
Errors were detected during rollback.
The session could not be finished consistent.
(&00): Number of session

Response
Correct error and restart MAINTASK with USE=C.

LEA5317 `USE=C: ROLLBACK INITIATED FOR TRANSACTION (&00)`

Meaning
(&00): Session-specific transaction-number

LEA5401 `AIM FILE (&00) IS EMPTY`

Meaning
The AIM file is empty.

Response
If necessary repeat reconstruction run with correct AIM generation.

LEA5402 `VALIDATION FINISHED. RECONSTRUCTION STARTING NOW.`

Meaning
The LEASY-RECONST utility routine executes with the TRA=V mode. All AIM records are checked wether they are guilty or not. Now the guilty records are reconstructed.

LEA5403 `FOLLOWING PARAMETERS ARE RESET:`

Meaning
This message follows the message LEA5402. For reconstruction it is necessary to change the following parameters.

LEA5404 `*MOD TRANSACTIONS = (&00). WARNING: NO AIM SESSION RECORD FOUND`

Meaning
No session entry has been found in the specified AM generations while reconstructing with *MOD TRANSACTIONS = (&00).

Response
Repeat run with all AIM generations if not all of them were processed.

LEA5405 `*MOD TRANSACTIONS = (&00). WARNING: FOR FOLLOWING TRANSACTIONS NEITHER AIM OPTR RECORD NOR AIM CLTR RECORD FOUND`

Meaning
Neither CLTR nor OPTR entries were found for the listed transactions while reconstructing with *MOD TRANSACTIONS = (&00).

Response
Repeat run with all AIM generations if not all of them were processed.

LEA5406 `AIM UPDATE RECORD FOUND WITHOUT PRECEDING AIM FILS RECORD OR AIM OPEN RECORD`

Meaning
Neither a FILS nor an OPEN entry has been found prior to the first AIM update entry.

Response
Start reconstruction with an AIM file with a lower generation number.

LEA5407 `MAYBE AIM FILE STILL OPENED BY A TRANSACTION OF RUNNING APPLICATION`

Meaning
The AIM file is probably still open via a transaction of an ongoing application.

`LEA5408` `.................(&00)............(&01)`

Meaning
Second part of LEA5405.

`LEA5409` `NO FILES FOR AUTOMATIC RECONSTRUCTION.`

Meaning
The tables of CMMAIN contain no files which are automatically to reconstruct.

Response
If necessary change the file attributes with LEASY-CATALOG and start the application again.

`LEA5501` `AIM FILE(S) (BEFORE/AFTER) (&00) ERASED`

Meaning
The specified AIM file generation group or the specified generation was erased.

`LEA5502` `TERMINATION OF AN IO TASK INITIATED`

Meaning
The termination of an IO task is initiated.

`LEA5503` `NO LOCKED RECORDS`

`LEA5504` `ROLLBACK OF TRANSACTION (&00) IN ACTION`

Meaning
(&00) Session specific Transaction Number

`LEA5505` `ROLLBACK OF TRANSACTION (&00) PROCESSED`

Meaning
(&00) Session specific Transaction Number

`LEA5601` `NORMAL TERMINATION OF ARCHIVE`

Meaning
The generated statements have been successfully processed by ARCHIVE.

`LEA5602` `NORMAL TERMINATION OF ARCHIVE WITH WARNINGS`

Meaning
Warning messages have been issued, but this has not prevented the generated statements from being executed.

Response
Check whether the desired save operation has been performed.

LEA5603 `NORMAL TERMINATION OF ARCHIVE WITH ERRORS`

Meaning
ARCHIVE has detected an error, but this has not prevented the generated statements from being executed.

Response
Check whether ARCHIVE has executed the desired save operation, and correct the input if necessary.

LEA5604 `ABNORMAL TERMINATION OF ARCHIVE`

Meaning
ARCHIVE has detected an error, and execution of the generated statements has been aborted.

Response
Correct the error with the aid of the ARCHIVE messages.

LEA5605 `ABNORMAL TERMINATION OF ARCHIVE: EXECUTION NOT STARTED!`

Meaning
ARCHIVE has detected an error, and execution of the generated statements has not started.

Response
Correct the error with the aid of the ARCHIVE messages.

LEA5606 `ERROR WHILE LOADING ARCHIVE OR ARCHIVE NOT AVAILABLE`

Meaning
A error has occurred while loading ARCHIVE or ARCHIVE is not available.

Response
Inform the system administrator.

LEA5704 `*LEASY IO-TASK (&00) INITIALIZATION COMPLETED*`

Meaning
All statements have been processed. The IO-Task of the directory (&00) is ready and enters a wait state.

LEA5705 `*LEASY IO-TASK INITIALIZATION FINISHED WITH ERROR*`

Meaning
At least one error has been detected in the operand check for the IO-Task or during generation of the common memory. The program is terminated.

Response
Correct error and repeat run.

LEA5801 CONVERSION OF DIRECTORY '(&00)' INTO (&01)-FORMAT STARTED

Meaning
Conversion of Directory (&00) into (&01) format has started
(&00): Name of directory
(&01): KEY / NONKEY

LEA5802 CONVERSION OF DIRECTORY '(&00)' COMPLETED

Meaning
Conversion of Directory (&00) was completed successfully.
(&00): Name of directory.

LEA5803 CONVERSION OF FILE '(&00)' INTO (&01)-FORMAT STARTED

Meaning
Conversion of file (&00) into (&01) format has started.
(&00): file name.
(&01): KEY / NONKEY

LEA5804 CONVERSION OF FILE '(&00)' WITH (&01) RECORDS COMPLETED

Meaning
Conversion of file (&00) was completed successfully.
(&00): file name.
(&01): number of records.

LEA6001 INVALID COMMAND BEGINNING WITH (&00)

Meaning
There is no valid statement in columns 1 to 4 of the input line. The input line contains (&00) in columns 1 to 8. Valid statements are:

*CAT	for LEASY directory specification
*FIL	for file specification
*COM	for comments
*ERA	for erase specification
*END	for terminating the program.
*HAL	for terminating the LEASY-LOADSI utility routine

Response
Correct statement keyword and repeat input.

LEA6002 INVALID (&00)PARAMETER

Meaning
The operand (&00) is invalid or incorrect.

Response
Correct error and repeat input.

LEA6003 `INVALID PARAMETER KEYWORD`

Meaning
Incorrect parameter keyword.

Response
Correct error and repeat input.

LEA6004 `RECORD CONTINUATION IN INPUT SENTENCE IN ERROR`

Meaning
A read error occurred during reading in of a statement using record chaining. The preceding message LEA7001 or LEA400x identifies the error.

Response
Correct error with the aid of the other error messages.

LEA6005 `SECONDARY INDEX: SI-NAME MISSING OR INVALID`

Meaning
A secondary index name contains invalid characters or is too long. Valid characters are letters, $, #, @ and digits.

Response
Correct error and repeat input.

LEA6006 `ILLEGAL CHARACTER(S) BEHIND LEASY DIRECTORY NAME`

Meaning
Invalid characters follow the LEASY catalog name.

Response
Correct input.

LEA6007 `ILLEGAL CHARACTER(S) BEHIND PARAMETER`

Meaning
Invalid characters after an operand or a secondary index name.

Response
Correct input.

LEA6008 `INVALID CHARACTER(S) AT END OF LINE`

Meaning
Invalid characters at the end of the input line.

Response
Correct input.

`LEA6009` `FILE NAME MISSING OR INVALID`

Meaning
The file name is missing from the current statement or is syntactically invalid.

Response
Correct input.

`LEA6010` `LEASY DIRECTORY NAME MISSING OR INVALID`

Meaning
A catalog name has either not been specified in the *CAT statement or is syntactically invalid.

Response
Correct input.

`LEA6011` `WARNING: ILLEGAL CHARACTER(S) BEHIND PARAMETER`

`LEA6012` `INVALID DEVICE`

Meaning
The specified device is not supported by the DMS or is not compatible with the requested device type (disk/tape).

`LEA6101` `UNDEFINED LEASY DIRECTORY`

Meaning
A attempt has been made to issue a statement other than *CAT as the first statement.

Response
Enter the *CAT statement and then repeat the original statement.

`LEA6102` `INVALID PASSWORD`

Meaning
The password is invalid or too long or there is a syntax error in the password input.

Response
Correct error and repeat input.

`LEA6103` `INVALID FCBTYPE`

Meaning
Only SAM, ISAM, PAM and DAM access methods are permitted for LEASY files at present.

Response
Correct error and repeat input.

LEA6104 `INVALID LEASY FILETYPE`

Meaning
An invalid LEASY file type has been specified. The following are valid:

S	for master files
M	for model files
T	for temporary files
F	for foreign files.

Response
Correct error and repeat input.

LEA6105 `SUFFIX IN FILENAME TOO LONG (MORE THAN 20 CHARACTERS)`

Meaning
The maximum file suffix length is 20 characters.

LEA6106 `WARNING: "SHARUPD" OR " DUPEKY" IGNORED`

Meaning
The SHARUPD parameter is ignored because LEASY controls multiple access to a file via the OPEN mode. DUPEKY is ignored because duplicates at primary key level would not be subject to BIM saving.

LEA6107 `VOLUME AND/OR DEVICE PARAMETER NOT ALLOWED FOR EXISTING LEASY DIRECTORY`

Meaning
The LEASY-CATALOG utility routine cannot relocate an existing LEASY directory on another disk: the VOLUME and DEVICE parameters are therefore only permitted when creating a new directory.

Response
A LEASY directory can only be relocated on another disk by appropriate use of BS2000 commands.

LEA6108 `INF = YES ONLY WITH TYPE = OLD`

Meaning
The TYPE parameter specification contradicts that of the INF parameter.

Response
Correct input.

LEA6109 `SECONDARY INDEX (&00): SYNTAXERROR`

Meaning
Secondary index specification contains an unidentified syntax error.

Response
Correct error and repeat input.

LEA6110 `SECONDARY INDEX (&00): KEYPOSITON WRONG, NOT NUMERIC OR INVALID OR POSSIBLE ERROR AT INDEX SUPPRESSION BYTE`

Meaning
The specified secondary index position is invalid, or the error is related to the index suppression byte specification.

Response
Correct error and repeat input.

LEA6111 `SECONDARY INDEX (&00): KEYLENGTH OR PART OF IT WRONG, NOT NUMERIC OR INVALID`

Meaning
An invalid key length or partial length has been specified for the secondary index (&00).

Response
Correct input.

LEA6112 `SECONDARY INDEX (&00): INVALID DUPKEY PARAMETER`

Meaning
An invalid DUPEKEY parameter was specified for the secondary index (&00).

Response
Correct input.

LEA6113 `SECONDARY INDEX (&00): INVALID DISTANCE FOUND`

Meaning
The displacement of a partial key of the secondary index (&00) is < 0 or > 32767.

Response
Correct input.

LEA6114 `SECONDARY INDEX (&00): INVALID KEYUPDATE PARAMETER`

Meaning
An invalid KEYUPDATE parameter has been specified for the secondary index (&00).

Response
Correct input.

LEA6115 `SECONDARY INDEX (&00): SYNTAXERROR AT INDEX SUPPRESSION BYTE`

Meaning
On specification of an index suppression byte for the secondary index (&00) one of the following errors occurred:

- Suppression byte not in the form X’xx’ or C’c’
- Suppression byte specified for total and partial secondary key
- Suppression byte not specified for all partial secondary keys

Response
Correct error and repeat input.

```
LEA6116     ILLEGAL CHARACTER(S) BEHIND SECONDARY INDEX (&00)
```

Meaning
Illegal characters follow a secondary index name.

Response
Correct input.

```
LEA6117     ILLEGAL CHARACTER(S) BEHIND FILE NAME
```

Meaning
Illegal characters follow the file name.

Response
Correct input.

```
LEA6118     M-FILENAME TOO LONG TO BUILD SI-FILENAME
```

Meaning
When adding secondary indices to a model system a instance whose suffix is too long has been found. Generation of secondary indices is not possible.

```
LEA6119     TYPE = E: CPC- AND CPS-PARAMETER NOT ALLOWED
```

Meaning
The specification of CPC and CPS is not allowed if erasing of a LEASY directory is requested.

```
LEA6120     SECONDARY INDEX: INVALID REPETITION FACTOR
```

Meaning
Invalid repetition factor. The repetition factor must be > 0 and < 32769.

Response
Correct input.

```
LEA6121     SECONDARY INDEX (&00): RECORD TYPE MISSING OR INVALID
```

Meaning
A record type is missing for the secondary index (&00) or is syntactically invalid.

Response
Correct input.

```
LEA6122     DELETE OF SECONDARY INDEX (&00): REPETITION FACTOR NOT ALLOWED
```

Meaning
A repetition factor has been specified when deleting the secondary index (&00).

Response
Omit repetition factor and repeat input.

LEA6123 `RECORD TYPE FIELD (&00) INVALID`

Meaning
(&00) = POSITION: The start of the record type field is outside the range 1 to 32768.
(&00) = LENGTH: The record type field length is < 1 or > 255.

Response
Correct input.

LEA6124 `SECONDARY INDEX (&00): ILLEGAL CHARACTER(S) BEHIND RECORD TYPE (&01)(&02)’`

Meaning
An illegal character has been discovered following the record type (&01) of the secondary index (&00).

Response
Correct input.

LEA6125 `INVALID READ PASSWORD`

Meaning
The password (&00) is invalid or too long or there is a syntax error in the password input.

LEA6126 `INVALID WRITE PASSWORD`

Meaning
The password is invalid or too long or there is a syntax error in the password input.

LEA6127 `INVALID NEW PASSWORD`

Meaning
The password is invalid or too long or there is a syntax error in the password input.

Response
Correct input.

LEA6128 `SECONDARY INDEX: SYNTAXERROR`

Meaning
Secondary index specification contains an unidentified syntax error.

Response
Correct error and repeat input.

LEA6129 `WRONG VALUE IN BLKCTRL OPERAND`

Meaning
Only the following parameters are allowed: PAMKEY, DATA, DATA2K, DATA4K.

Response
Correct the BLKCTRL operand and try again.

`LEA6130` `LINKNAME (FOR POOL/SIPOOL) IS PRIVILEGED.`

Meaning
the following names are not allowed for user: $TASK01-16, $SYS01-16.

Response
give a new linkname and repeat the run.

`LEA6131` `BLKCTRL CAN NOT BE MODIFIED`

Meaning
The file attribute BLKCTRL can not be simply changed from PAMKEY to DATAor NO, or contrary, but only by conversion of the file with the utility LEASY-CONVERT.

`LEA6132` `POOLNAME MISSING OR INVALID`

Meaning
In the *POO-instruction the linkname is missing or invalid.

Response
Correct the *POO-instruction and try it again.

`LEA6133` `CATID TOO LONG`

Meaning
The catid may have max. 4 characters (from BS2000 V10 on).

Response
Correct the *POO-instruction and try again.

`LEA6134` `INVALID SIZE`

Meaning
There are only values between 32 and 524288 allowed.

Response
Correct the *POO-instruction and try again.

`LEA6135` `INVALID CATID`

Meaning
Give catid without colons.

Response
Correct the *POO-instruction and try again.

`LEA6137` `OLDL AND NEWL-OPERANDS ARE THE SAME; NO EXECUTION`

Meaning
The same is specified in the NEWL and OLDL operands. Nothing is done.

LEA6201 SYNTAXERROR

Meaning
Missing or invalid delimiter in the last input line.

Response
Wait for prompt.

LEA6202 WRONG SPECIFICATION

Meaning
Incorrect or missing keyword in SI specification.

Response
Wait for prompt.

LEA6203 ADD OR DELETE SPECIFICATION WITHOUT SI NAME

Meaning
An ADD or DELETE specification contains no SI name, or the blank between the specification keyword and the file name is missing

Response
Wait for prompt.

LEA6204 AT LEAST ONE BIM-FILE IS NOT EMPTY; OLDL-/NEWL-PARAMETER WILL NOT BE EXECUTED

Meaning
At least one BIM file is not empty. The statement is not executed. The user must first create a consistent state.

LEA6301 INVALID NUMERIC VALUE IN PARAMETER

Meaning
Invalid numeric value. Values must be integer and within the following ranges:

for *TRANS	1 <= x <= 255
for *TIME	0 <= x <= 999
for *FILES	1 <= x <= 999
for *MEMORY	1 <= x <= 99
for *KEYLEN	1 <= x <= 255.

Response
Correct error and repeat input.

LEA6402 MORE THAN 255 EXTRACT PARAMETERS ARE NOT ALLOWED

Meaning
More than 255 EXTRACT specifications have been made in the *REP statement.

LEA6403 `FILE FILTER OR ALL PARAMETER MISSING OR INVALID`

Meaning
In the FILE FILTER statement the ALL parameter or the file selection is invalid, or the entry has been omitted.

Response
Correct input.

LEA6404 `ILLEGAL CHARACTER(S) BEHIND FILENAME OR PARANTHESIS MISSING`

Meaning
Syntax error.

Response
Correct input.

LEA6405 `INVALID (&00)`

Meaning
(&00) is invalid.
(&00) = date: Is to specify in the form yymmtt or jjjj-mm-tt.
(&00) = time: Is to specify in the form hhmm.
(&00) = TSN : 1 to 4 alphanumeric positions.

Response
Correct input.

LEA6406 `EXTRACT PARAMETER: SYNTAX ERROR`

Meaning
Syntax error in the EXT parameter.

Response
Correct input.

LEA6407 `EXTRACT PARAMETER: POSITION MISSING OR INVALID`

Meaning
The position specification in an EXT parameter is missing or invalid.

Response
Correct input.

LEA6408 `EXTRACT PARAMETER: INVALID CHARACTERS BEHIND POSITION`

Meaning
The start position specification in an EXT parameter is followed by invalid characters.

Response
Correct input.

LEA6409 `EXTRACT PARAMETER: LENGTH MISSING OR INVALID`

Meaning
The length specification in an EXT parameter is missing or invalid.

Response
Correct input.

LEA6410 `EXTRACT PARAMETER: ILLEGAL CHARACTER(S) BEHIND LENGTH OR PARANTHESIS MISSING`

Meaning
Syntax error in an EXT parameter after the length specification.

Response
Correct input.

LEA6411 `AIM RECORD SELECTION MISSING OR INVALID`

Meaning
The record type selection is missing or invalid.

Response
Correct input.

LEA6412 `ILLEGAL CHARACTER(S) BEHIND AIM RECORD SELECTION OR PARENTHESIS MISSING`

Meaning
The AIM record type selection is followed by invalid characters or parentheses are missing.

Response
Correct input.

LEA6501 `NUMBER OF CYCLES WRONG, NOT NUMERIC OR INVALID`

Meaning
The numeric entry is invalid.

LEA6502 `SYNTAX ERROR IN TRANSACTION ID`

Meaning
A syntax error has occurred in the transaction identifier specification.

LEA6503 `SYNTAX ERROR IN FILE SPECIFICATION`

Meaning
A syntax error has occurred in the file specification.

LEA6504 `SYNTAX ERROR IN DISPLAY TIME`

Meaning
A syntax error has occurred in the display time specification.

`LEA6505` `SYNTAX ERROR IN TRANSACTION NUMBER`

`LEA6506` `NUMBER OF TRIALS WRONG, NOT NUMERIC OR INVALID`

Meaning
The specification of trials for end of transactions is no number or a number greater than 99.

`LEA6507` `WAITING TIME WRONG, NOT NUMERIC OR INVALID`

Meaning
The specification of the waiting time between two trials for end of transactions is no number or a number greater than 99.

`LEA6601` `INVALID CHARACTER(S) BEHIND VSN`

Meaning
Invalid characters following the VSN.

Response
Wait for prompt.

`LEA6602` `INVALID CHARACTER(S) BEHIND DEVICE`

Meaning
Invalid characters following the device.

Response
Wait for prompt.

`LEA6603` `INVALID SAVING OPTION`

Meaning
An invalid type of saving has been specified.

Response
Wait for prompt.

`LEA6604` `SECOND CHANGED PARAMETER MISSING OR INVALID`

Meaning
The second parameter of the CHANGED entry is missing or invalid.

Response
Wait for prompt.

`LEA6605` `VSN MISSING OR INVALID`

Meaning
No VSN or a syntactically invalid VSN has been entered.

Response
Wait for prompt.

LEA6606 `WARNING: SOME VSN ARE IGNORED`

Meaning
The area provided for the volume serial numbers entered is not large enough to accommodate them all.

LEA6607 `ERROR: FIRST CHARACTER EQUAL SPACE AND FOLLOWING CHARACTERS NOT EQUAL SPACE`

Meaning
The response to the message LEA0606 or LEA0607 begins with a blank followed by characters other than blanks.

Response
Wait for prompt.

LEA6608 `MISSING OR INVALID USERID`

Meaning
An invalid input has been made in response to the message LEA0608.

Response
Wait for prompt.

LEA6609 `WARNING: ILLEGAL CHARACTER(S) AFTER LAST USERID`

Meaning
The fifth user ID entered in response to the message LEA0608 is followed by the other characters. These characters are ignored.

LEA6610 `NO LEASY TEMPORARY FILES FOR USERID (&00)`

Meaning
No occurrences of temporary files have been found under user ID (&00), which was entered in response to the message LEA0608.

LEA6611 `WARNING: MAYBE NOT ALL LEASY TEMPORARY FILES OF USERID (&00) ARE SAVED`

Meaning
It is possible that not all occurrences of temporary files under user ID (&00) have been saved.

Response
Inform LEASY maintenance.

LEA6612 `UNKNOWN USERID (&00)`

Meaning
The user ID (&00), which was entered in response to the message LEA0608, does not exist.

LEA6613 CATID (&00) CANNOT BE FOUND

Meaning
The catid (&00) which was entered in response to the message LEA0608 cannot be detected.

LEA6701 INVALID NUMERIC VALUE IN PARAMETER

Meaning
Values must be integer and within the following ranges:

for *ARL	12 <= x <= 32678
for *DBL	12 <= x <= 32678
for *IOT	1 <= x <= 255
for *KBL	1 <= x <= 255
for *KEL	1 <= x <= 255
for *USE	1 <= x <= 1800
for *WAI	1 <= x <= 43200

Response
Correct error and repeat input.

LEA6702 LEASY FILE NAME MISSING OR INVALID

Meaning
A file name has either not been specified in the *OPF statement or is syntactically invalid.

Response
Correct input.

LEA6703 LEASY-OPEN-MODE MISSING OR INVALID

Meaning
The specified LEASY open mode is missing or wrong.
Allowed are: 1 (INPUT), 2 (INPUT,SHARUPD), 3 (INOUT), 4 (INOUT, SHARUPD), 5 (REVERSE), A (OUTIN) and B (OUTIN,SHARUPD)

LEA6801 UNDEFINED LEASY DIRECTORY

Meaning
The LEASY Directory does not exist.

Response
Correct Name of Directory and retry.

LEA7001 MACRO ERROR AT MODUL (&00). MACRO = (&01). CODE = X"(&02)".

Meaning
While using the macro (&01) an error (&02) has been detected in module (&00).

Response
Identify error with the aid of the "Executive Macros" manual.

LEA8001 `ERROR WITH FILE (&00). ERRORCODE OF LEASY RUNTIME SYSTEM IS (&01)`

Meaning
During the processing of file (&00) an error has occurred. Details are provided by the runtime system error code (&01).

LEA8002 `TIME LIMIT FOR PROGRAM RUN EXCEEDED`

The CPU time limit specified with the program call START-LEASY-<program> has expired. The program terminates abnormally.

Glossary

AIM

After-image saving:
- storage of the logical data contents after updating
- protection in the case of hardware faults (destruction of data fields)
- reconstruction after system crash
- implementation using AIM file and save copies of the original files.

BIM

Before-image saving:
- storage of data contents before updating
- protection in case of software errors (program abortion)
- no interruption of or interference with other transactions
- warm start capability (restart after software errors)
- implementation using BIM files.

common memory

Common memory for all tasks connected to a catalog.

DAM

Direct Access Method:
Derived from the relative file organization of COBOL, and designed in accordance with the KLDS standard, additionally supported by secondary indices (see chapter “File management”, in the partner manual “LEASY (BS2000) - Program Interface and Strategies”).

DCAM

Data Communication Access Method:
inquiry and transaction mode in BS2000.

deadlock

General state of waiting for system resources which, due to the particular configuration involved, can never terminate without outside intervention.

ISAM

Indexed Sequential Access Method as defined by the data management system (see the "Introductory Guide to DMS" manual), additionally supported by secondary indices (see chapter “File management”, in the partner manual “LEASY (BS2000) - Program Interface and Strategies”).

lock

Feature of the primary sort key of a data record for the logical separation of write accesses to one record by different transactions.

openUTM

Universal Transaction Monitor. UTM is used for implementing inquiry and transaction mode in BS2000.

PAM

BS2000 primary, block-oriented access method (see the "Introductory Guide to DMS" manual), additionally supported by secondary indices (see chapter “File management”, in the partner manual “LEASY (BS2000) - Program Interface and Strategies”).

primary index

The primary index is a unique sort key for files that permit direct access.

- The ISAM key of the file is the primary sort key of ISAM files.
- The block number (half-page number) of the PAM block is the primary sort key of PAM files, and is at the beginning of the user’s logical data block (the data block can extend over several PAM blocks of up to 2048 bytes each).
- A 4-byte value greater than or equal to zero, which is not part of the file record, is the sort key of DAM files.

SAM

Sequential files as defined by the data management system (see the "Introductory Guide to DMS" manual).

secondary index

Sort term for records of an ISAM, DAM or PAM file, which, like the primary sort key, enables direct accessing of the records in a file.

SI

Secondary index.

SI file

Secondary index file. This file contains the secondary key pointers to primary keys of the user file.

transaction

A series of file access operations which are processed as a single unit.

UPAM

See PAM.

UTM

See openUTM.

Related publications

The manuals are available as online manuals, see *http://manuals.fujitsu-siemens.com*, or in printed form which must be paid and ordered separately at *http://FSC-manualshop.com*.

LEASY (BS2000/OSD)
Program Interface and Strategies
User Guide

LEASY (BS2000)
Ready Reference

COBOL85 (BS2000)
COBOL Compiler
User’s Guide

COBOL2000 (BS2000/OSD))
COBOL Compiler
User’s Guide

Assembler (BS2000)
Reference Manual

SDF (BS2000/OSD)
Introductory Guide to the SDF Dialog Interface
User Guide

BS2000/OSD-BC
Commands, Volume 1 - 5
User Guide

BS2000/OSD-BC
Executive Macros
User Guide

BS2000/OSD-BC
Utility Routines
User Guide

SORT (BS2000/OSD)
User Guide

ARCHIVE (BS2000/OSD)
User Guide

BS2000/OSD-BC
System Messages, Volume 1 and Volume 2
User Guide

***open*UTM** (BS2000/OSD)
Generating and Handling Applications
User Guide

***open*UTM** (BS2000/OSD, UNIX, Windows NT)
Administering Applications
User Guide

***open*UTM** (BS2000/OSD, UNIX, Windows NT)
Programming Applications with KDCS for COBOL, C and C++
Core Manual

DCAM (TRANSDATA)
Program Interfaces
Reference Manual

DCAM (TRANSDATA)
COBOL Calls
User Guide

JV (BS2000/OSD)
Job Variables
User Guide

BS2000/OSD-BC
Security Handbook for Systems Support

DRIVE/WINDOWS (BS2000)
Programming System
User Guide

BS2000/OSD-BC
Introductory Guide to DMS
User Guide

BS2000/OSD-BC
DMS Macros
User Guide

FHS (TRANSDATA)
User Guide

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.