TABLE OF CONTENTS Page:

APPENDIX A
SIBAS ON NORD-10

APPENDIX A

SIBAS ON NORD-10

Machine and SINTRAN III dependent description of the SIBAS program package and its use.

Note that this description is not static and will change as SINTRAN III and the SIBAS implementation on NORD-10 develop. Deviations from this description will be given in the "NORD Software Library" Description.

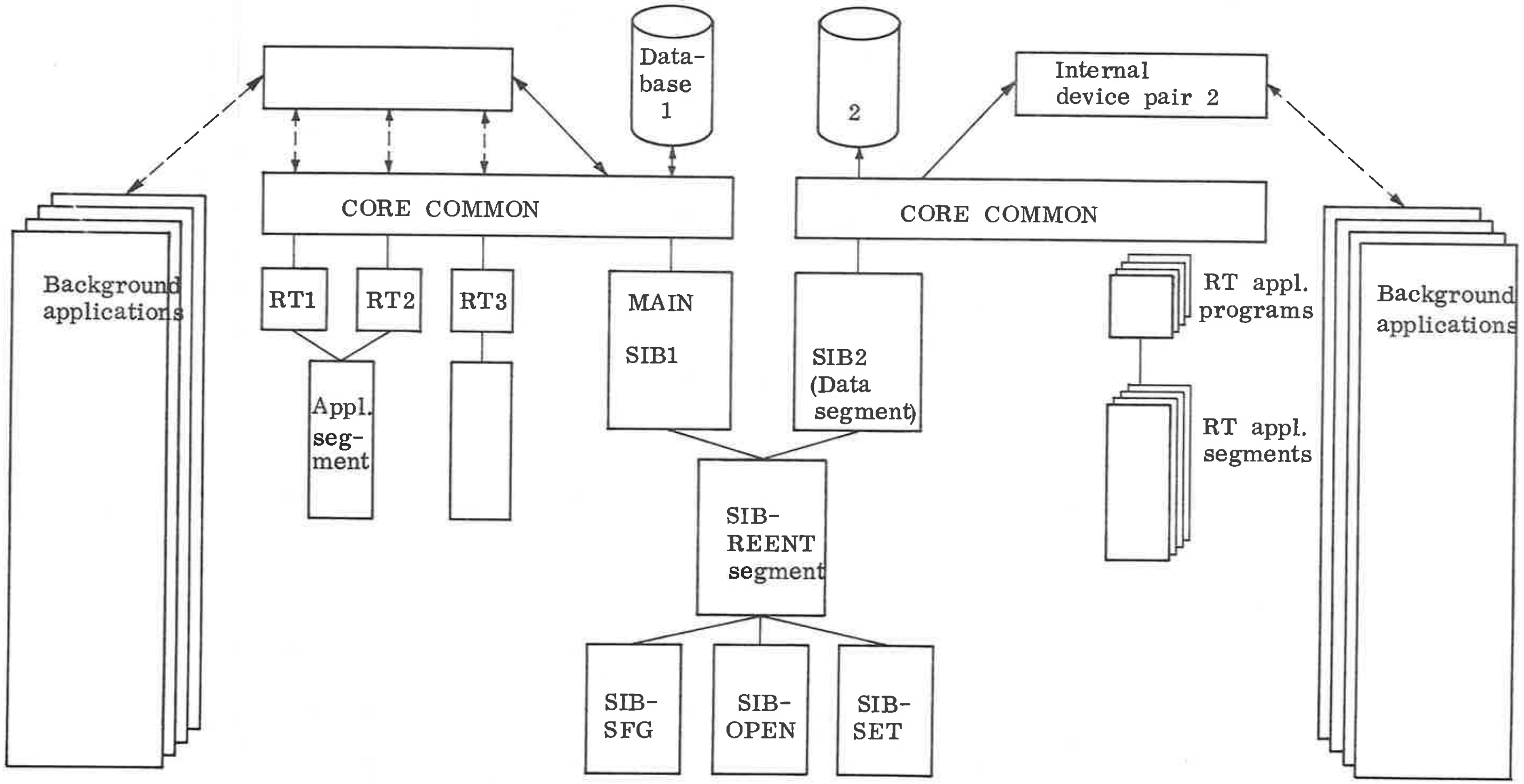
A.1 Introduction

SIBAS is now present in NORD Software Library as a program package consisting of the following modules:

SIBAS-INITIATION	- SIB 1927	(binary)
SIBAS-SYSTEM	- SIB 2120	(symbolic)
- SIB-MAIN	- SIB 2113	(binary)
- SIB-REENT	- SIB 2114	(binary)
- SIB-SFG	- SIB 2115	(binary)
- SIB-OPEN	- SIB 2116	(binary)
- SIB-SET	- SIB 2117	(binary)
SIB-DML-C	- SIB 2118	(BRF)
SIB-DML-RT	- SIB 2119	(BRF)
SIB-DML-B	- SIB 2121	(BRF)
SIBINTER	- SIB 1985	
SIBAS-DATABASE-ADMINISTRATOR	- SIB 1921	(binary)

By use of this package one can create a database and run SIBAS SYSTEM and application programs as separate processes communicating through internal devices as shown in Figure 1.

Two SIBAS systems running on shared code. Application programs as background or real time processes.



A.2 Program Modules

A.2.1 SIBAS-INITIATION

This module is used for database definition as described in the SIBAS Users Guide. Data definition "cards" are read from the given input file, and a description of the database is listed.

Definition of Files

On the NORD-10 a SIBAS database is stored on files defined under the NORD file system. The SIBAS system realm is always located on a file of the same name as the database, and this file must be of type DATA, and must lie on the default directory of the user running the initiation program. (The database owner.)

The remaining realms in a database can either be stored in the same file as the SIBAS system realm, or in other files, not necessarily on the users default directory. The rule is: If column 76 of an NREALM card is not 1, the realm will be stored on the same file as the previously defined realm and will automatically receive the same page size as this previous realm. If column 76 of an NREALM card is 1, the realm will be stored on an existing file of type DATA with the same name as the realm. In this case columns 77-80 may contain a unique abbreviation of a directory name.

Thus, a database containing N SIBAS realms may be stored on 1 to N directories (discpacks).

By defining dummy SIBAS realms the SIBAS system realm can easily be patched to store 1 SIBAS realm on more than one directory.

Note:

> We recommend as few SINTRAN files as possible grouping SIBAS realms of same page size.

SIBAS Realm Page Size

On NORD, one can choose the page size of the SIBAS system realm to be 32, 64, 128 or 256 words.

For all user defined realms stored in their own files a page size may be given on the NREALM card. The page size (PS) read in from the card is treated as follows.

Page Size Read	Page Size Used
PS = 0	256
0 < PS < = 64	64
PS > = 1024	1024
64 < PS < 1024	rounded up to 128, 256 or 512

For user defined realms containing records, it is important to remember that it may be necessary to read two pages into core if the record one is interested in lies across a page boundary.

Transfer of 512 word pages will be highly optimized and is therefore recommended.

Use of Common

SIBAS uses a common buffer as its work area. In the NEW-SIBAS the length of this buffer is initiated by start of the SIBAS RT program. Some of this buffer is taken up by fixed tables, and the rest is available as a dynamic buffer pool, and is divided into the following:

a) SIBAS tables,

b) working buffers used while SIBAS is executing a call,

c) a software paging area into which pages are read from the database, or from which pages are written to the database, i.e., prolongation into core of the most frequently used parts of the database.

A.2.2 SIBAS-DATABASE-DEFINITION

This module simplifies the production of data definition cards by reading them from a terminal in interactive mode.

Output from this process is a card file that can later be edited by QED if necessary. Remember always to remove tabs for output by QED (M to (0)).

A.2.3 SIBAS-SYSTEM

This module consists of one batch file to load 5 segments used by one SIBAS RT system (one data segment in addition for each extra SIBAS system). Each system can run in the following modes:

- Normal mode: SIBAS-SYSTEM executing DML statements for the application programs using the SIBAS internal device interface. The SIBAS interface functions basically as follows:

  The user side collects all the input to the desired SIBAS call from the application program, carries out a certain amount of checking, and then sends it to SIBAS.

  The SIBAS side of the interface collects the input, builds up the input to the desired call and calls the relevant routine. When the call is completed, the output (often only the status parameter IST) is collected, and sent back to the calling program.

  The user side of the interface then collects the return data and gives it back to the application program which then continues executing.

  Errors can arise during any of these parts of the process of executing a SIBAS call, and those not resulting in the normal negative IST values described in the manual are termed interface errors. The values of IST given for these errors are from -60 to -99.

  Every SIBAS call that can lead to a change of the database is logged on a routine log-file, also the resulting answer. That means that it is possible to simulate a run using a back-up of the database and use the log file as input instead of application programs. By this reprocessing, SIBAS normally compares the "new" answers and the logged "old" answers.

Note that SIBAS-SYSTEM uses core common for the interface buffer. That means that SIBAS system 1 uses the uppermost page in core common, and system 2 uses the next page, etc.

Note also that at system start you can give the name of the database owner (default RT). User RT must be "friend" and have Read and Write access to the owners files. The owner of the call log file must be given as part of the file name (default RT).

- Reset log file:

  The content of the DML call log file is reset.

- Recover and start:

  The DML call log file is reprocessed from a given checkpoint or from start of log file (checkpoint id = 8 zeros).

  Before reprocessing SIBAS asks if you will supress testing during the reprocessing of calls, and whether you want to reprocess the whole log file, or stop after reprocessing a number of calls. If the latter is chosen, then routine numbers, user etc. are printed out according to the following rules:

  - No printout when more than 100 calls left.

  - Printout of checkpoints and BSEQU/ESEQU when 11 to 99 calls left.

  - Printout of every call when 10 or less left.

- Listing checkpoints and Skip or Reset - uncompleted critical sequences:

  The call log is traced printing out all found checkpoints. All uncompleted critical sequences (see routines BSEQU, ESEQU) are removed from the call log, skipped sequences (names, data and user identification) are printed. By using this run mode once more those sequences are reset.

SIBAS-SYSTEM (SIB1, SIB2 or SIB3) must be started by using the command "RT" before any application program executes a SIBAS call.

SIBAS-SYSTEM is able to make the following types of log:

i) Page Log

Before and/or after image page log can be defined using the

SIBAS-DATABASE-ADMINISTRATOR.

Checkpoints are then taken automatically when opening or closing the database physically, i.e., first user opening or last user closing.

ii) DML Call Log

Every call that can change the database can be logged if one gives a call log file name when starting SIBAS. The log file is blocked into 1/2K pages.

This log file should be allocated as a continous file (default type = LOGG).

Note that all checkpoints and all return parameters from SIBAS are also logged. Compared to earlier versions the log volume is 2 to 3 times greater for the same number of database calls.

iii) Terminal Log

Every DML call can also be written out on this terminal in the format:

<routine no.> <user ID>

Both numbers are 2 digits decimal, that means user ID>99 will give a peculiar printout.

Interaction between page and call log:

i) Running SIBAS-SYSTEM

Before starting SIBAS-SYSTEM in normal mode, the log can be reset. When SIBAS starts, it searches for the last logged call and continues logging from that point.

Remarks:

- When the call log is full, SIBAS-SYSTEM prints out "NO SUCH PAGE" from time to time; execution continues normally.
- When only one page image log file is defined and that log file is full, then SIBAS-SYSTEM is aborted. Recovery must consequently be done.

ii) Recovery after system or hardware failure

If SIBAS-SYSTEM is stopped leaving the database physically open, then the database is usually inconsistent.

Recovery including the most recently logged DMR calls can then be done using a combination of logs and a database security copy.

Every system stop without the printout "SIBAS STOPPED" on the database terminal is a

No image log defined:

i) Enter the last database security copy.

ii) Start SIBAS-SYSTEM.

iii) Use SIBAS-SYSTEM to read and execute the DMR calls on the log file. Checkpoint zero (8 zeroes) are used.

After image page log defined:

i) Enter the last database security copy.

ii) Use SIBAS-DATABASE-ADMINISTRATOR to RECOVER, using after image log up to the last checkpoint.

iii) Display log to get last checkpoint identification.

iv) Continue as if no image log were defined. That is using the page-recovered database as security copy and the routine log file from last checkpoint.

Before image page log defined:

i) Use SIBAS-DATABASE-ADMINISTRATOR to roll back, using before image log, to the last <u>close-database checkpoint.</u> (Not a user checkpoint or a reserved space on page log file checkpoint.)

ii) Continue as if no image log were defined.

Removing critical sequences from call log:

By using special user routine calls, sequences of DML calls could be marked on the call log.

Reprocessing of uncompleted sequences could be avoided by removing them prior to performing roll back/recover using page log and reprocessing. One should compare times of printed checkpoints and times for the omitted sequences to find the correct checkpoint to roll back/recover to.

A.2.4 SIBAS-DML-SIMULATOR

This module is to be loaded together with application programs to simulate the SIBAS data manipulation routines. This module and a similar one inside the SYSTEM program constitutes the SIBAS internal device interface.

The interface functions basically as follows:

- The user side (DML-SIMULATOR) collects input parameters from the user program, carries out a certain amount of checking and sends it through an internal device. The SIBAS side of the interface collects parameters through another internal device.

- The DML-SIMULATOR-CORE uses the most effective way of transferring information from RT applications using RT COMMON to SIBAS, that is RT COMMON.

- The DML-SIMULATOR-RT is used by RT programs not using RT COMMON.

- The DML-SIMULATOR-BACKGROUND is used by background programs.

We therefore access SIBAS via an interface instead of loading application programs and SIBAS together. Some of the reasons for doing this are:

- MULTIUSER SIBAS

  Up to 64 (at present) different application processes can share SIBAS. DML calls are executed one at a time for the different user programs.

- LARGE APPLICATION PROGRAMS

  Application program sizes greater than 60K words are possible, as the DML-SIMULATOR needs less than 2K.

- DATABASE SECURITY

  Erroneous user programs cannot destroy the database (writing out side array bounds for instance). Accordingly errors are separated into USER ERRORS, internal SIBAS ERRORS and INTERFACE ERRORS.

- BETTER MAINTENANCE

  Having only one SIBAS-SYSTEM identical for all customers means:

  - it is easier to find bugs, if necessary a patch can be sent to all customers;
  - each error is found only once.

- RUNNING SEVERAL DATABASES FROM THE SAME PROGRAM

  SIBAS SYSTEM needs two internal devices. Serving several internal device pairs by the same number of SIBAS SYSTEM programs, allows one user program to run against several databases. The program can simply switch internal device pairs using the routine SETDV (see below). This feature is convenient when your database system structure needs one level higher than the realm level. Building a new database from an old can easily be done in the same program by reading the old and storing into the new database. (Different database owners.)

- COOPERATING RT PROGRAMS WORKING AS ONE "SIBAS USER"

  Normally the RT DESCRIPTION address of the application process is used as user identification.

  The identification can be defined in another way: An assembly routine "IDGET" can be made and loaded before the DML-SIMULATOR. "IDGET" should leave all registers unchanged except the A register that returns the user identification.

  SIFLG should contain the SIBAS SYSTEM number (0, 1 or 2) + 1. SIFLG is used by SETDV and SIBOF.

  <u>Note:</u>
  The DML-SIMULATOR-CORE and RT refer to SIFLG as an external symbol. One should therefore define and set initial value of SIFLG by loading. If the DML-SIMULATOR-code is shared between several RT programs using the routines "SETDV" or "SIBOF" they should have their own SIFLG loaded on different segments in the same logical address.

Known disadvantages are:

- COMMUNICATION AND SYSTEM OVERHEAD

  When using 4-word-oriented internal devices there is about a 15 percent CPU overhead on each DML call. Other system overhead is less than 5 percent.

  By using RT COMMON the total overhead is reduced to about 5 percent.

- CHANGED SPECIFICATIONS ON SOME DML CALLS

  Sometimes the internal device interface needs additional information about the length of array parameters (see below).

A.2.5 SIBAS-INTERACTIVE

This module is run as an application program, executing DML calls in interactive mode. For explanation see the manual: "NORD Interactive Information Processing and Report Generating System".

A.2.6 LOADING INSTRUCTIONS

- SIBAS-DML-SIMULATORS are in BRF code. The other modules are loaded as described on the library description sheet.

- SIBAS-SYSTEM
  After editing this batch input file to get good segment numbers, SIBAS-SYSTEM can be loaded using the resulting file.

A.3 Requirements for SIBAS on NORD-10

A.3.1 SINTRAN III

One must have SIBAS internal devices (at least 2) which can be reserved from background programs. They should have at least 64 words buffer size, i.e., SIBAS internal devices.
4 segments + one for each SIBAS RT program is necessary.
Also 1K RT COMMON for each SIBAS RT program.

A.3.2 MEMORY

SIBAS-SYSTEM has a working set of 40-60K. To run SIBAS effectively without thrashing one must have sufficient physical memory.

Below are some loose rules for the memory requirements (only SIBAS users logged on):

64K	One simple data base and 1-2 applications with OK throughput.
96K	2 applications with good throughput, or 3 applications with OK throughput. 2 SIBAS-SYSTEM programs (buffer size 10K)
128K	More than 4 applications with good throughput, 2 SIBAS-SYSTEM programs (buffer size > 15K).

If one requires good throughput for several applications with many identical routines, these should be real time programs sharing the code. Such configurations need knowledge of RT programs and the RT loader.

Also time critical applications should be RT programs using DML-SIMULATOR-CORE to achieve the most effective communication.

A.4 Changes Necessary to "STANDARD SIBAS" Application Programs

- OPEN DATA BASE (SOPDB)

To avoid the logging of calls which do not change the database, the user identification parameter IBRID is used on opening to distinguish between Retrieval or Load/Update. (This parameter IBRID is only used by the interface, SIBAS gets another identification.

IBRID = 0 (Retrieval)	No logg of DML calls is done. No realms can then be readied for load/update. If that is tried, the interface gives status IST = -89.
IBRID = 15473 (Load/Update)	The application program is allowed to change the data base. Every DML call is logged.

Application programs seldom changing the data base are recommended separated in sections for retrieval and for load/update (paranthesized by SOPDB-SCLDB).

The call logg volume and the time for doing recovery can thus be limited.

- EXTRA PARAMETER IN SFTCH, STORE, SMDFY, SFEBL

To transfer correct number of words in value buffers through the internal devices, these routines have an extra parameter (last parameter) called LENG.

LENG =	length in words of the key value for SFTCH, SFEBL,
LENG =	length in words of data items' value buffer for STORE, SMDFY.

Too small value of the LENG parameter gives incorrect parameter transmission to SIBAS-, sometimes error status -64 or -62.

To high values of LENG give time-consuming transmission.

We therefore recommend that LENG be properly set before each such call.

- EXTRA SUBROUTINES SETDV,UTBLK, SIBOF, RBLAN, ZTRB, SABOR, RESIB, RELSI, BSEQU, ESEQU, STOPS, GANTA

CALL SETDV (SIBSYSNO)
Changes the SIBAS SYSTEM now used by this application (internal device numbers).
Default value is 0. (SIFLG=SIBSYSNO+1)

CALL UTBLK (IST)
is implemented to ensure transaction consistency with respect to the log file. The current log buffer is written to the log file, that is, all previous calls are now logged. This call is automatically executed by BSEQU, ESEQU and STOPS, and also at every checkpoint and when SIBAS detects an internal SIBAS error.

CALL SIBOF
turns off calling the database. If SIBAS is "turned off", every call returns "IST" = 1, and no data is transmitted to SIBAS via the interface. SIBAS is turned on with SETDV.

CALL RBLAN (IND, VALUE, IST)
gets contents of SIBAS, common address IND.

CALL ZTRB (N, IND, VALUES, IST)
gets N words from SIBAS common address IND.

CALL SABOR (DBNAM, IST, IBRID)
closes database for IBRID.

RESIB/RELSI (IST)
Reserves/releases SIBAS SYSTEM given by SIFLG for exclusive use. That is the internal device pair is reserved.

RESIB puts -SIFLG into SIFLG. SETDV and SIBOF should therefore not be used between RESIB and RELSI.

STOPS (IST)
Stops the SIBAS SYSTEM RT program in a controlled way. Status IST=0 if somebody has left the database open.

GANTA (IFIL, key name, antfi, antindex, ist)
Counts no. of records in readied realm "IFIL" and index table "key name" in double words (antfi and antindex). Number in realm is only counted when antfi = 0.

Note:

The routines SABOR, STOPS, GANTA are by now implemented as part of the "GO-TO-USER-SUBROUTINE" in SIBINTER.

BSEQU/ESEQU8(NAVN, IST)

Critical sequences in one application program could be parenthesized by these calls. The effect is that the calls are logged with thei r identifying "NAVN" and time. Also UTBLK is automatically performed.

If the system goes down in the middle of sequences, it is possible by reprocessing to undo the partly executed sequences. This can be successfully done provided these sequences have no critical interactions with other sequences or calls, i.e., reprocessing without the uncompleted sequences will result in a database where all application program sequences except the uncompleted, have been processed.

Note: Before using this excellent technique to avoid garbage collection after a breakdown, your whole application system should be carefully analyzed to avoid or get control over interacting sequences with respect to reprocessing.

ACCID/ACCFD/ACCDD (ID, NAME, INCR/RINCR/DINCR, NEW-value, IST)

Accumulates integer/floating or double word data element given by name in record given by ID. An increment is added to the data element and the new value is returned.

Note: The accumulation is executed as an uninterrupted sequence.

Execution of these routines is therefore often necessary or convenient in an application system where recovery strategy is based on editing of critical sequences.

A.5 Errors

A.5.1 APPLICATION PROGRAM

Every programmed stop of applications should close the database. Otherwise one can use SIBAS-INTERACTIVE to close.

"Escape" from a background application is dangerous. If this is done while communication to SIBAS is going on, it may be impossible to close the database physically. Subsequently, recovery must be done if attempts to close the database only give errors.

A.5.2 SIBAS INTERNAL ERRORS

SIBAS-76-SYSTEM performs a great amount of checking. When recognizing fatal errors, SIBAS produces a dump of all possibly relevant data on a list file.

The reason for the dump could be:

i) Not corresponding sizes of SINTRAN files with SIBAS files given by database definition (too small SINTRAN file, i.e., "NO SUCH PAGE").

ii) Inconsistent database, SIBAS-76-SYSTEM is earlier started on a "not physically closed" database.

iii) Error in SIBAS. The complete dump should be sent to ND immediately.

Recovery can be done. Every DML call is then reprocessed from the logg file. If recovery also results in an internal error, the new dump is also important for the ND supporter.

A.5.3 INTERFACE ERRORS

i) Fatal Errors

If any fatal errors occur, then an illegal instruction (WAIT) is executed by the interface. Wait nos. are found by looking at the preceding memory cell. That contains 151000 + wait no. (octal).

Fatal errors in application part	Caused by	Wait no.
Release "to SIBAS internal device" did not work	SINTRAN	2
Release "from SIBAS internal device" did not work	SINTRAN	3

ii) Error Status Number Returned to Application Program

Errors detected	Caused by	Error no.
Interface buffer full (>512)	Appl. program	-62
Negative no. of el. in array	Appl. program	-64
Wrong routine no.		-83
Database not open for LOAD/ UPDATE by appl. program	Appl. program	-89

A.6 Table of Routine Numbers

The internal device interface is using one set of routine numbers. SIBAS is using another as seen in SIBAS Users Guide. The former is logged by the terminal logg. The latter is returned by calls to SDBEC.

SIBAS DML Routine	Logged No.	SIBAS Routine no.
SFTCH	1	1
SRFSM	2	13
SRNSM	3	11
SRLSM	4	14
SRPSM	5	12
SRSOW	6	15
SGET	7	20
SMDFY	8	32
STORE	9	31
SRASE	10	33
SREMB	11	60
SLOCK	12	62
SFORG	13	61
SUNLK	14	63
SINSR	15	45
SCONN	16	41
SREMB	17	46
SDCON	18	42
SRRLM	19	52
SOPDB	20	50
SFRLM	21	53
SCLDB	22	51
SFEBL	23	2
SRFIR	24	3
SRNIS	25	16
SCONB	26	44
SCONA	27	43
SCHPW	28	54

SIBAS DML Routine	Logged No.	SIBAS Routine no.
SDBEC	29	-
SEREL	30	34
SCHPO	32	-
SINIT	33	-
STERM	34	-
SLOG	35	-
RBLAN	36	-
ZTRA	38	-
SABOR	39	-
QTYP	40	-
STOPS	41	-
QPAG	42	-
GANTA	43	-
BSEQU	44	-
ESEQU	45	-
ACCID	46	20/32
ACCFD	47	"
ACCDD	48	"

A.7 Examples

A.7.1 INITIAL LOADING OF SIBAS-SYSTEM

See library description sheet and the comments in the SIBAS-SYSTEM batch input file.

A.7.2 SIBAS-INITIATION AND BACKUP

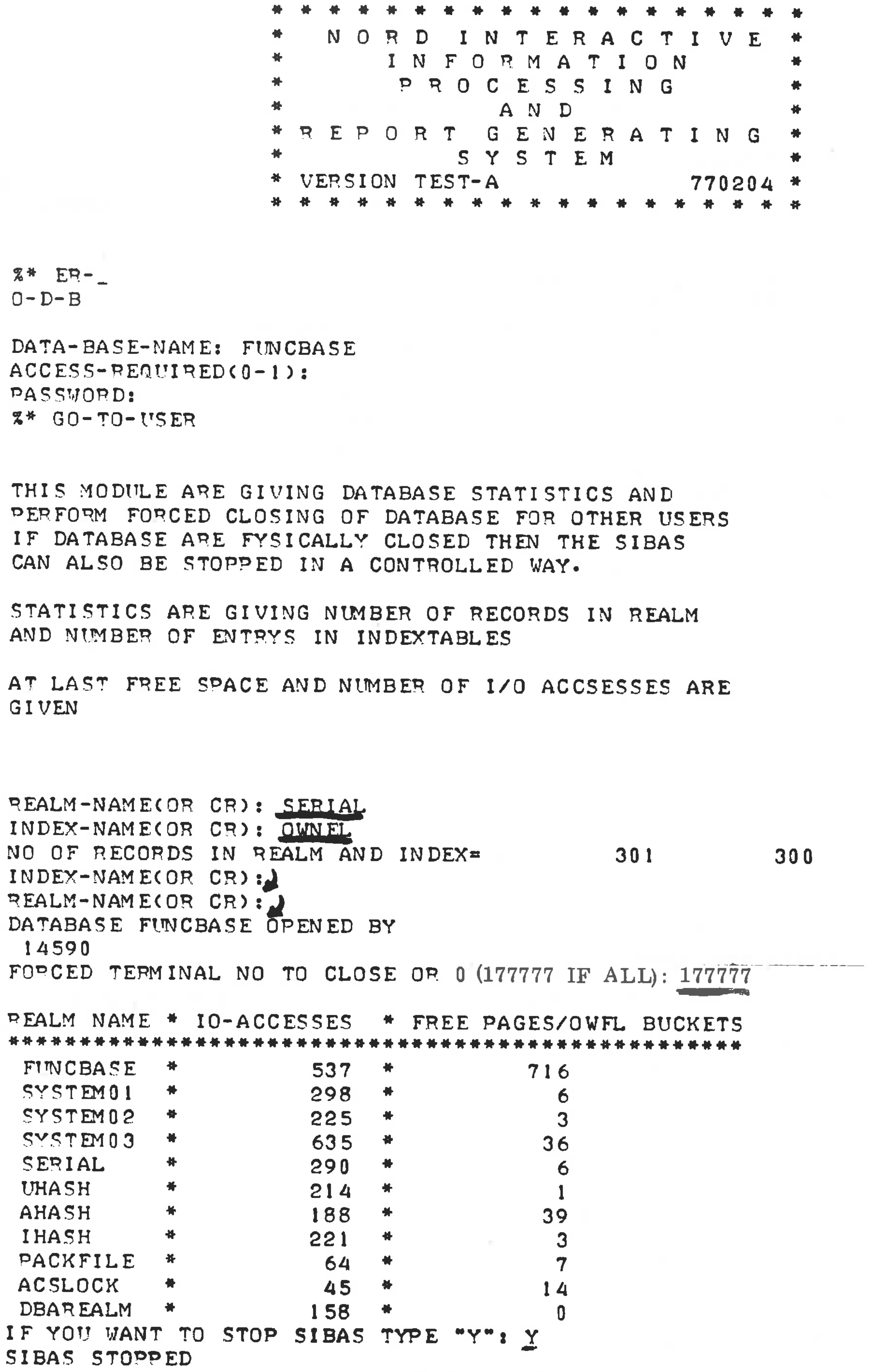
```
PASSWORD:
OK
@SIBAS-76-INIT

DO YOU WANT ADVICE ? IF YES TYPE "Y" RETURN
IF NOT,TYPE" ANY OTHER CHARACTER" RETURN: Y

SIB-1782B 02/12/75
A FILE OF TYPE DATA WITH THE SAME NAME AS THE SIBAS SYSTEM
REALM MUST EXIST.REMAINING REALMS ARE STORED ON THIS FILE
OR ON SEPARATE FILES OF TYPE DATA.THE RULE IS:
IF COLUMN 76 OF AN NREALM CARD IS NOT 1 THE REALM IS STORED
ON THE SAME FILE AS THE PREVIOUSLY DEFINED REALM AND WILL
HAVE THE SAME PAGE SIZE AS THIS REALM.
IF COLUMN 76 OF AN NREALM CARD IS 1 THE REALM WILL BE
STORED ON AN EXISTING FILE OF TYPE DATA OF THE SAME NAME.
COLUMNS 77-80 MAY CONTAIN AN ABREVIATED DIRECTORY NAME.
A DATA BASE CONTAINING N REALMS MAY THUS BE STORED ON FROM
1 TO N FILES.
ALLOWED PAGE SIZES ARE:
 SIBAS SYSTEM REALM: 32,64,128,256 WORDS
 USER REALMS: 64,128,256,512,1024 WORDS (DEFAULT 256 WORDS)
BOTH INPUT FILE AND LIST FILE MUST BE EXTERNAL DEVICES,OR
FILES OF TYPE SYMB.
CHOOSE PAGE SIZE FOR SIBAS SYSTEM REALM ,1,2,3,4 FOR 32,64,128,256: 2
INPUT FILE: TESTK
LIST FILE: L-P
001073 STOP    0
@MODE BACKUP-FBASE:MODE,TELE

@COPY-FILE XFUNCBASE:DATA,FUNCBASE:DATA
@COPY-FILE XSYSTEM01:DATA,SYSTEM01:DATA
@COPY-FILE XSYSTEM02:DATA,SYSTEM02:DATA
@COPY-FILE XSYSTEM03:DATA,SYSTEM03:DATA
@COPY-FILE XUHASH:DATA,UHASH:DATA
@COPY-FILE XAHASH:DATA,AHASH:DATA
@COPY-FILE XIHASH:DATA,IHASH:DATA
@COPY-FILE XSERIAL:DATA,SERIAL:DATA
@COPY-FILE XPACKFI:DATA,PACKFI:DATA
@COPY-FILE XACSLOC:DATA,ACSLOC:DATA
```

A.7.3 SIBAS-INTERACTIVE GIVING STATISTICS AND CLOSING DOWN THE SYSTEM



```
                         * * * * * * * * * * * * * * * * * * *
                         *   N O R D  I N T E R A C T I V E  *
                         *       I N F O R M A T I O N       *
                         *        P R O C E S S I N G        *
                         *              A N D                *
                         * R E P O R T  G E N E R A T I N G  *
                         *           S Y S T E M             *
                         * VERSION TEST-A            770204  *
                         * * * * * * * * * * * * * * * * * * *


%* ER-_
O-D-B

DATA-BASE-NAME: FUNCBASE
ACCESS-REQUIRED(0-1):
PASSWORD:
%* GO-TO-USER


THIS MODULE ARE GIVING DATABASE STATISTICS AND
PERFORM FORCED CLOSING OF DATABASE FOR OTHER USERS
IF DATABASE ARE FYSICALLY CLOSED THEN THE SIBAS
CAN ALSO BE STOPPED IN A CONTROLLED WAY.

STATISTICS ARE GIVING NUMBER OF RECORDS IN REALM
AND NUMBER OF ENTRYS IN INDEXTABLES

AT LAST FREE SPACE AND NUMBER OF I/O ACCSESSES ARE
GIVEN



REALM-NAME(OR CR): SERIAL
INDEX-NAME(OR CR): OWNFL
NO OF RECORDS IN REALM AND INDEX=          301          300
INDEX-NAME(OR CR):↵
REALM-NAME(OR CR):↵
DATABASE FUNCBASE OPENED BY
 14590
FORCED TERMINAL NO TO CLOSE OR 0 (177777 IF ALL): 177777

REALM NAME * IO-ACCESSES  * FREE PAGES/OWFL BUCKETS
****************************************************
 FUNCBASE  *          537  *          716
 SYSTEM01  *          298  *            6
 SYSTEM02  *          225  *            3
 SYSTEM03  *          635  *           36
 SERIAL    *          290  *            6
 UHASH     *          214  *            1
 AHASH     *          188  *           39
 IHASH     *          221  *            3
 PACKFILE  *           64  *            7
 ACSLOCK   *           45  *           14
 DBAREALM  *          158  *            0
IF YOU WANT TO STOP SIBAS TYPE "Y": Y
SIBAS STOPPED
```

A.7.4 INITIATION OF BEFORE IMAGE LOG ON LOGG:DATA AND CALL LOG ON LOGG:LOGG

```
@(SIBAS)DBA-MOD

 : START FUNCBASE.
 : DEFINE DBA-REALM DBAREALM 20.
 : DEFINE LOG-FILE LOGG DISC 10^^200000 10000 S-S 512.
 : DEF LOG-TYPE DATABASE BEFOR LOGG.
 : DISPLAY LOG.
  ** LOG  DEFINITIONS ON DATABASE FUNCBASE  MAXBUFFER =      1 **
  ***************************************************************


  **  LOG-FILE LOGG      **
  *************************
  MEDIUM      2  (1=TAPE,2=DISC,3=DRUM); STATUS 0
  FILE-SIZE IN WORDS                  200000
  USED SIZE IN WORDS                       0
  RESERVED-LENGTH                      10000
    SECTOR-SIZE IN WORDS                 512
    CURRENT SECTOR                         0
    NEXT WORD IN SECTOR                    0
   LOG-TYPES DEFINED :
  --------------------
     BEFORE-LOOK
  LAST USED CHECKPOINT-ID :  0  0  0  0     0  0    0 0
 : STOP.
010147 STOP     0
@RT SIB1

@RT SIB1

@LOG
 08.54.38     19 JULY   1977
--EXIT--


SIBAS SYSTEM START

MODUS =
 0 - NORMAL START
-1 - RESET LOG-FILE
-2 - RECOVER AND START
-3 - LIST CHECKPOINTS AND SKIP/RESET UNCOMPLETED SEQUENCES
MODUS: -1
NO OF ROUTINE-LOG PAGES OR 0: 100
GIVE ROUTINE-LOG FILE: LOGG


READY ? Y


SIBAS RUNNING



SIBAS STOPPED
```

A.7.5 REMOVING UNCOMPLETED SEQUENCES AND LISTING CHECKPOINTS

```
S I B A S   S Y S T E M   S T A R T

MODUS =
 0 - NORMAL START
-1 - RESET LOG-FILE
-2 - RECOVER AND START
-3 - LIST CHECKPOINTS AND SKIP/RESET UNCOMPLETED SEQUENCES
MODUS: -3
SIBAS SYSTEM (0-2): 0
GIVE ROUTINE-LOG FILE: LOG


READY ? Y


S I B A S   R U N N I N G


CHECKPOINT TIME,NR=    19    34    59     8    19     7 1977     0
CHECKPOINT TIME,NR=    23    35    59     8    19     7 1977     0
CHECKPOINT TIME,NR=    30    39    59     8    19     7 1977     0
CHECKPOINT TIME,NR=    34    43    59     8    19     7 1977     0
CHECKPOINT TIME,NR=    40    47    59     8    19     7 1977     0
SEQUENCE: USER=14694 ID=SEKVENS3 TIME:     14     6     0     9    19     7 1977
```

A.7.6 ROLL BACK USING BEFORE IMAGE LOG

```
  09.17.39      19 JULY   1977
 NORD 10/S - VERSION 77.06.07
 ENTER RT
 PASSWORD:
 OK
 @(SIBAS)DBA

  : START FUNCBASE.
  : ROLL CHAC^^ECK
 (BASIC-UNIT)
  : 40
 (SECOND)
  : 47
 (MINUTE)
  : 59
 (HOUR)
  : 8
 (DAY)
  : 19
 (MONTH)
  : 7
 (YEAR)
  : 1977
 (SEQUENCE-NUMBER)
  : 0
 (LOG-FILE)
  : LOGG
ADJUST
.
  : .
  : DISPLAY LOG.
   ** LOG  DEFINITIONS ON DATABASE FUNCBASE  MAXBUFFER =  4210 **
   ***************************************************************


   **  LOG-FILE LOGG     **
   ************************
   MEDIUM      2  (1=TAPE,2=DISC,3=DRUM); STATUS 0
   FILE-SIZE IN WORDS                 200000
   USED SIZE IN WORDS                    352
   RESERVED-LENGTH                     10000
     SECTOR-SIZE IN WORDS                512
     CURRENT SECTOR                        0
     NEXT WORD IN SECTOR                 440
    LOG-TYPES DEFINED :
   --------------------
      BEFORE-LOOK
   LAST USED CHECKPOINT-ID : 40 47 59  8     19  7 1977 0
 : STOP.
010147 STOP    0
@
```

A.7.7 REPROCESSING FROM START OF ROUTINE LOG (Checkpoint zero)

Note: When less than 100 of wanted no. of calls are left, then checkpoints and sequences are listed; when less than 10, then all calls are listed.

```
%* @PT SIBI

%%* @LOG
 11.00.52       22 AUGUST    1977
--EXIT--


S I B A S   S Y S T E M   S T A R T

MODUS =
 1 - NOPMAL START WITH OLD PARAMETERS
 0 - NORMAL START
-1 - PESET LOG-FILE
-2 - PECOVER AND CONTINUE
-3 - LIST CHECKPOINTS AND SKIP/RESET UNCOMPLETED SEQUENCES
MODUS: -2
SIBAS SYSTEM (0-2): 0
GIVE ROUTINE-LOG FILE: LOG
GIVE CHECKPOINT TO START OR 8 "0,"): ,,,,,,,,
TERMINAL LOG?
DATABASE OWNER:
SIBAS WORK AREA (10<32 K ): 32

READY ? Y
CHECKPOINT TIME    47    47    59    10    22     8 1977 NR     0
IF SUPRESS TEST ON REPROSESSING OF CALLS TYPE "-1": 0
REPROSESS WHOLE LOGFILE (0),STOP(-1) ELSE NO OF CALLS: 10
USER 034376 NEXT ROUTINE NO     20
USER 034376 NEXT ROUTINE NO     22
CHECKPOINT TIME    17    48    59    10    22     8 1977 NR     0
CHECKPOINT TIME    28    54    59    10    22     8 1977 NR     0
USER 034376 NEXT ROUTINE NO     20
USER 034376 NEXT ROUTINE NO     22
CHECKPOINT TIME    40    54    59    10    22     8 1977 NR     0
CHECKPOINT TIME    14    13     0    11    22     8 1977 NR     0
USER 034376 NEXT ROUTINE NO     20
USER 034376 NEXT ROUTINE NO     22
IF SUPRESS TEST ON REPROSESSING OF CALLS TYPE "-1": 0
REPROSESS WHOLE LOGFILE (0),STOP(-1) ELSE NO OF CALLS: 100
CHECKPOINT TIME    27    13     0    11    22     8 1977 NR     0
CHECKPOINT TIME    26    17     0    11    22     8 1977 NR     0
CHECKPOINT TIME    38    17     0    11    22     8 1977 NR     0
CHECKPOINT TIME    44    24     0    11    22     8 1977 NR     0

REPROSESSING FINISHED, NORMAL MODUS SET

CURRENT 1/2K ROUTINE-LOGG BLOCK     0 ,WORD 000366


      S I B A S   R U N N I N G
```

A.7.8 NORMAL SIBAS RUN

Database files owned by user SIBAS.

Note: Default user name for routine log file is RT.

```
S I B A S   S Y S T E M   S T A R T

MODUS =
 0 - NORMAL START
-1 - RESET LOG-FILE
-2 - RECOVER AND START
-3 - LIST CHECKPOINTS AND SKIP/RESET UNCOMPLETED SEQUENCES
MODUS: 0
SIBAS SYSTEM   (0-2): 0
NO OF ROUTINE-LOG PAGES OR 0: 1
GIVE ROUTINE-LOG FILE: (RT)LOGG
TERMINAL LOG? Y
DATABASE OWNER: SIBAS
SIBAS WORK AREA (10<32 K ): 32

READY ? Y


S I B A S   R U N N I N G


20M4 19M4 24M4 19M4 43M4 36M4 36M4 36M4 36M4 38M4
36M4 39M4 36M4 36M4 36M4 38M4 38M4 38M4 38M4 38M4
38M4 38M4 38M4 38M4 38M4 38M4 38M4 38M4 38M4 38M4
38M4 38M4 38M4 38M4 38M4 36M4 42M4


S I B A S   S T O P P E D
```