Chapter 8

STANDARD DATA MANAGEMENT

8.1 INTRODUCTION

This chapter discusses the file types handled by SDM, and how to create them. The indexing mechanism for an indexed file structure for SDM is explained. In addition, details of file enlargement under SDM and notes on file maintenance are given. A list of the file handling instructions supported by SDM and a survey of the Status Word and Return Status values that may be returned by SDM are also included.

8.2 SDM INSTRUCTION SET

The following instructions are supported by SDM:

File handling instructions:

OPEN .DOUT	Create a new standard or indexed file of S-type and open for direct output
OPEN .EXT	Open and Extend an existing standard file for sequential output
OPEN .IN	Open an existing file for input only
OPEN .INOUT	Open an existing file for input and output
OPEN .SOUT	Create a new standard file and open for sequential output
CLOSE	Close file
CLOSE .DROP	Close and delete file
POSIT .DIR	Set Current Record Number on specified record
POSIT .IXDIR	Set Current Record Number on record with specified key
DSC X'19'	Read File Parameters

Record handling instructions

READ .DIR	Read record with specified relative key
READ .SEQ	Read next record using the relative key
READ .IXDIR	Read record with specified symbolic key
READ .IXSEQ	Read record with next symbolic key
WRITE .DIR	Write record with specified relative key
WRITE .SEQ	Write next record using the relative key
WRITE .IXDIR	Write record with specified symbolic key
REWRITE .CUR	Rewrite current record
REWRITE .DIR	Rewrite record with specified relative key
REWRITE .IXDIR	Rewrite record with specified symbolic key
DISCARD .CUR	Delete current record
DISCARD .DIR	Delete record with specified relative key
DISCARD .IXDIR	Delete record with specified symbolic key

Transaction Control Instructions

COMMIT	Release the records accessed during the current transaction
COMMIT .REL	Release the records accessed during the current transaction, on the specified files only

Record Protection

When a file is opened with Sharability "Protected", any records accessed will be under exclusive access for the task. A second task requiring access to a record which is already under exclusive access will have the message "record protected" (bit 11) in the Status Word, and the other records held protected for this task will be released.

This is called automatic rollback, because the task should then go back to the previous COMMIT.

As a consequence, an application design is recommended where all records used during one transaction are read before any record is updated.

8.3 FILE TYPES

SDM handles standard files and indexed files of S-type.

It is possible to handle common files (with a common file code) in SDM but then the Assembler interface must be used for the Open, Close and I/O functions. Common files can not be handled by the CREDIT interface.

8.3.1 Standard Files

A standard file is a file where the records are identified by the relative key. The records have a fixed length and are grouped into blocks that always start on a logical sector boundary. Each record has a status byte indicating if the record is "used" or "free". There is no free record chain. The Last Record Number (LRN) defines the logical end of the file (see section 8.4.3). Up to 64 file extents are allowed per volume, and up to 4 file sections. The file name consists of one alphabetic ISO-7 character followed by up to 7 alphanumeric characters.

8.3.2 Indexed File of S-type

An indexed file of S-type is a data file with the same characteristics as a standard file. In addition to the relative key, the records are identified by up to four symbolic keys. Each key must be contained in a separate index. For each symbolic key there is an Index file and Master Index file of S-type.

8.3.3 Index File of S-type

An Index file of S-type has the same characteristics as a standard file. It contains one index of an indexed data file of S-type. Each record contains the symbolic key of a data record together with the relative key of that record in the data file. The index records are is ascending order of the binary value of the symbolic keys. An Index file of S-type can not have more than one file section.

Index files are sequentially searched during an indexed file access. The index file name must consist of the first 6 characters of the name of the data file to which it belongs, with the prefix "In" where n is the sequence number of the index. The prefix "I1" must always denote the primary index.

Index Record

Each index record has the following layout:

Where:

Key	: The symbolic record key.
Reserved	: 2 bytes containing binary zeroes.
Duplicate key	: 1 byte binary value representing the minimum number of leading characters of the symbolic key which is identical with the symbolic key in the next index record.
Record number	: 3 bytes containing the relative key of the data record identified by this symbolic key.
Status byte	: One byte indicating if this index record is "used" (status X'FF') or "free" (status X'00').

8.3.4 Master Index File

A Master Index file of S-type has the same characteristics as a standard file. It contains the level 1 index or "master index" to one Index file of S-type. A Master Index file of S-type can not have more than one file section.

Each master index record corresponds with one partition in the Index file and contains the highest symbolic key and the relative record number of the first index record in that partition. The last record of the master index file contains the value X`FF' in every character position of the symbolic key field.

With the Master index file it can be determined which partition of the index file must be searched for the specified key. SDM performs a fast binary searchof the Master index.

The master index file name must consist of the first 6 characters of the data file name to which it belongs, with the prefix Mn where n is the sequence number of the corresponding index.

When an index file is opened, the corresponding master index file is read into memory. When the file is closed, the master index area in memory is released.

After the index and master index files have been built or reorganised by utility RIX, the number of index records per index partition is equal to:
The number of used index records divided by the total length of the master index expressed in number of records.

During dynamic use of the file structure, when records are added or deleted, the index file is updated but the master index is not and after some time the number of used index records in each partition will vary. Running RIX again with the same size of the master index, will result in a smaller or larger number of index records per partition.

It is also possible to have an empty master index, e.g. when there is no memory space available for it. After running RIX the master index file must then be deleted, and a new file of S-type with the same master index name must be created but left empty (LRN = 0). When the index file is opened this empty master index is read into memory, occupying only a few bytes. As a result, the index file will always be searched from the beginning (sequential search).

Both index files and master index files may be opened as standard files and the records read, written and updated if required. Consistency of the files is the user's responsibility.

For index files and master index files several file extents are allowed but all index and master index files belonging to one data file must reside on one volume.

8.3.5 File Structure

Data file, index files and master index files together constitute a file structure. A data file with the name NAMES may have the index files I1NAMES, I2NAMES, with their master indexes M1NAMES and M2NAMES.

Fig 8-1 is an example of such a file structure. Data record 11 is found via prime key "888" and via second key "Berry".

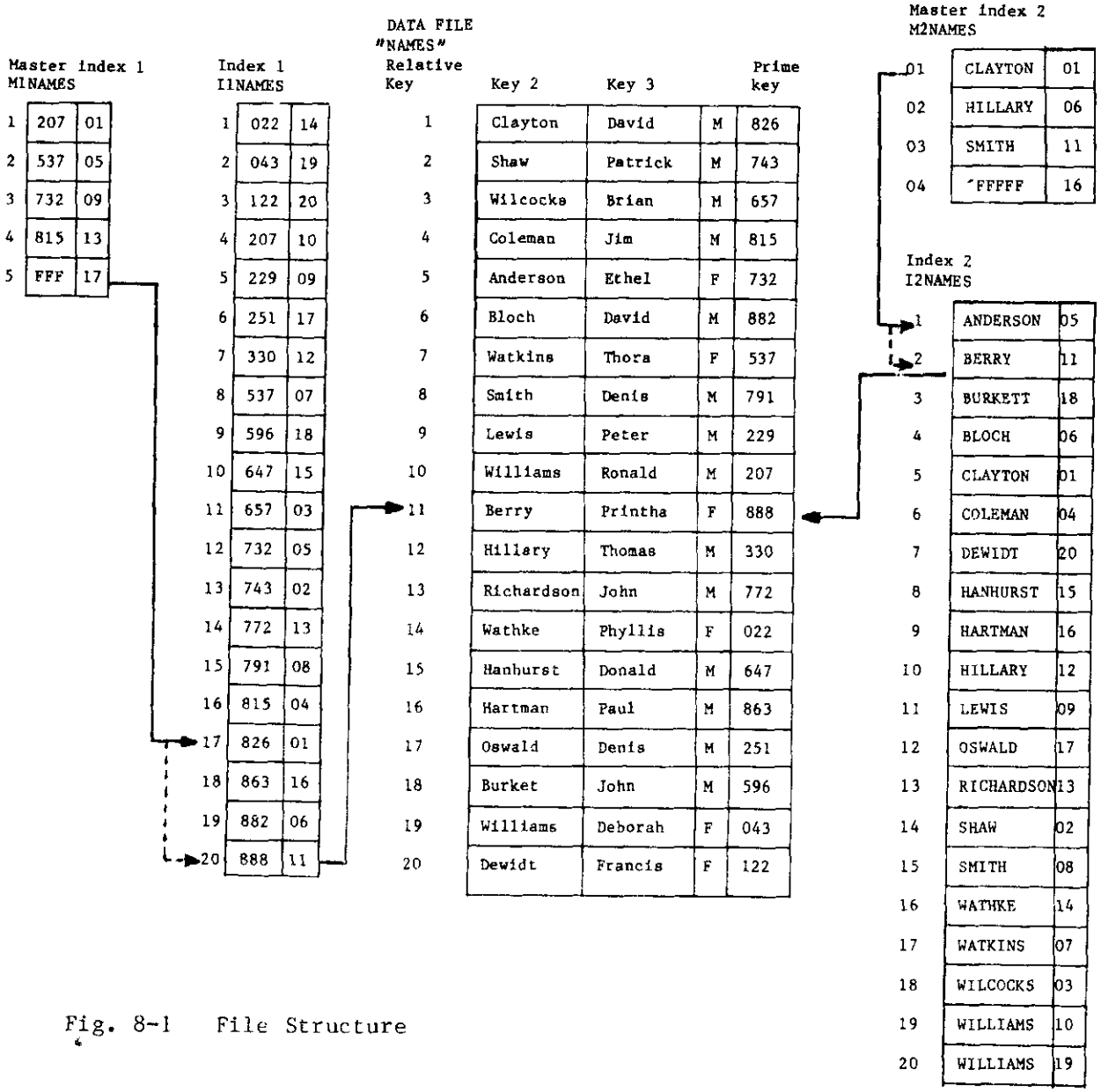


Fig. 8-1 File Structure

8.4 RECORD IDENTIFICATION

8.4.1 Record Keys

Symbolic record keys must consist of one key item or character string, and the key length must not exceed 64 bytes.

Prime Key

At least one of the keys must be unique for each data record. This is the prime key. The index containing the prime key must be defined as the first index when the file is created or opened.

The prime index must not contain duplicate keys. The records are identified by the prime key for the following instructions:

Write Indexed Sequential
Write Indexed Direct
Rewrite Indexed Direct
Delete Indexed Direct

The other keys are called alternate keys, and for those duplicates may exist. SDM does not check to see if there are duplicates. Note that the parameter "Index Type", indicating if duplicate keys are allowded for an index, is not significant for SDM but only for EDM.

Duplicate Keys

Duplicate keys are keys that have the same value in a number of data records. For indexed accesses on these records, the first one is found by an indexed direct access and the others are then accessed by indexed sequential instructions.

8.4.2 Currency

SDM holds a Current Record Number (the CRN) per data file for each task. This is the relative key of the current record for the task. The CRN is set by Read and Posit instructions. The CRN is used for record identification by Read Sequential, Rewrite Current and Discard Current instructions.

In SDM, the Posit instruction can not be used to set the CRN for Rewrite and Discard instructions. Rewrite Current and Discard Current after a Posit instruction will access the record last read.

Per task, SDM maintains the currency for one index at the time, that is the index specified for the last indexed instruction. The currencies for the other indexes associated with the data file will be zero, so that the data record associated with the first entry in such an index is accessed when a different index is used.

The Index currency is used for record identification by Read Indexed Sequential instructions.

8.4.3 Last Record Number (LRN)

Per S file (not per task), SDM holds a Last Record Number (the LRN). This denotes the logical end of the file, and the start point for subsequent Write instructions except Write Direct on a standard file.

For a Standard file, the LRN points to the last record written by Write Sequential instructions.

For an Indexed file of S-type, the data records are written to the data file sequentially with Write Indexed Direct instructions. The LRN points to the last record written by Write Indexed Direct instructions.

When the LRN is reached by Read Sequential instructions (CRN and LRN have the same value), the message "End of file" is returned.

When a new file is created, the LRN is preset to zero.

With Read Direct and Write Direct instructions (non-indexed!) it is possible to write and read records after the LRN. The LRN is not updated by direct access instructions. However, only "used" records can be read and rewritten, and only "free" records can be written.

The LRN is stored in the VTOC record for the first file extent. When a file is opened the LRN is read into memory and updated when new records are written to the file, except by non-indexed Write Direct instructions. The LRN is written back to the VTOC record when the file is closed.

For an Index file of S-type, the LRN is the relative key of the last index record in the last used partition of the file. This may be a "free" record when index entries have been deleted afterwards, because the LRN is not updated when records are deleted.

8.5 FILE CREATION

Indexed files of S-type are indexed random files, where the data records need not be in sequence of any of the keys. The records are located internally by the relative key.

When an indexed file structure is created, the file size specified for the data file and the index files must allow for updates on the file because SDM does not allow re-use of deleted records, and indexed files can not be extended.

Standard files and indexed files of S-type may be created by SDM or by the TOSS utilities Create File (CRF), Build Index File (BIX) and Reorganise Index File (RIX). These are described in the TOSS Utility Reference Mnaual, module M8A.

8.5.1 Creating a File by SDM

New standard and indexed files can be created during runtime by SDM. To create a new file the file must be opened with open mode Output Sequential or Output Direct.

The number of index files and master index files specified in parameter "Number of Indexes" is created on the volume specified for "Index Volume Name". The index files will have the size needed to contain the number of index records equal to the number of records specified for the data file. The master index files will have the size needed to contain the number of master index records equal to the number of index records in the index file, divided by 8.

When an indexed file structure is created by SDM, the master index file will be created but it is not filled. After writing the data records to the file, the master index must be built by running the utility RIX as described in phase 7 in the next section.

When opened for Output Direct, the file will be formatted at the Open instruction (the records are filled with spaces and the status is set to "free"). When opened for Output Sequential, the part of the file after the LRN will be formatted when the file is closed.

8.5.2 Creating Files by the TOSS Utilities

Creation of a file structure is performed in the following phases:

1 The data file must be created by the TOSS utility Create File (CRF).
2 The index file and master index file must be created with utility CRF.
3 The data records are written to the file.
4 One or two intermediate files must be created as workfiles to sort the data records and build the index file.
One work file is needed if the records in the file are in sequence of the key for which the index file is built. If this is not so, one extra workfile is needed for the Sort utility.
5 The intermediate index file must be built on the first workfile, by the TOSS utility Build Index File (BIX).

6 The index records in the workfile must be sorted on key value, by the TOSS utility Sort (SRT). Output of SRT is the second workfile.
7 The index file is built from the sorted workfile, by the TOSS utility Reorganise Index File (RIX).
Free records are distributed over the sectors on the index file according to the load factor specified, and the master index is created.
8 Data file, index file and master index file are now available for use by the application. The workfiles can be deleted by the TOSS utility Delete File (DLF).

If the data records contain more than one symbolic key, steps 4 through 8 must be repeated for every index file of the file structure.
All index files and master index files belonging to the same data file must reside on one volume.

Detailed descriptions of the utilities CRF, BIX, SRT and RIX are found in the TOSS Utilities Reference Manual M8A.
Most of the parameters for the utilities are self-explanatory.
However, some that may need more explanation are discussed here:

Phase 1 - Create the data file with CRF.

- File organisation: "S" must be stated for all files.
- Number of records:
File size should allow for extension and updates of the file.
- Number of index files:
Up to 4 index files may be specified. If there are no indexes, answer zero.
- Key address in data record:
This question is only relevant when creating an index file. When creating the data file answer zero.

CRF now searches the volume(s) for free extents large enough to hold the stated file size. The file is created with the required number of records, all containing space characters and all with a status byte indicating "free" (X'00'). The LRN is set to zero in the VTOC record of the first file extent.

Phase 2 - Create the index file and master index file with utility CRF.

- File name:
For the index file, the file name is the first six characters of the data file name, with the prefix In where n is the sequence number of the index file. The prefix "I1" denotes the primary index.
For the master index file, the file name is the first six characters of data file name with the prefix Mn where n is the sequence number of the associated index file.
- File organization :
"S" must be specified for index and master index files.
- Volume name:
For index files and master index files the same volume name must be specified.
- Index volume name:
Not relevant when creating index and master index files.

- Blocking factor:
  Number of records per block. This is not the number of index records per partition. The blocking factor is determined in the same way as for the data file.
- Record length:
  For both files, the record length depends on the length of the symbolic key of the data record.
  For the index file the record length must be keylength +6, and for the master index file the record length must be keylength +3, specified in bytes.
- Number of indexes:
  Not relevant, answer zero.
- Number of records:
  For the index file, a larger number of records must be specified than the number of records for which the data file was created, to allow for insertion of index records into the last sector of the index file.
  For the master index file, the most efficient file size must be chosen for the memory space available and the access times required. See also section 8.5.3, Master Index File Size.

Phase 3 - Write data records to the file.

In most cases the data records will be written to the file by the application. This may be performed by Sequential or Direct Write instructions.

The LRN is not updated by Direct Write instructions. However, TOSS utility BIX does not check the LRN but reads the entire data file until End of Medium, ignoring records with status "free".

Phase 4 - Create the intermediate files with CRF

CRF is run to create the work files. File names and volume names may be chosen as convenient.

- Record length:
  For both files record length must be the same as for the index file, which is keylength +6.
- Number of records:
  Number of records must be at least the (estimated) number of used data records now in the data file. This number is equal to the number indicated by the LRN of the data file if the records have been written with Sequential Write instructions.
  No free space is needed in the intermediate files.
- Key address in data record:
  Answer zero when creating the intermediate files.

Phase 5 - Build intermediate index file with BIX

- Address of key in record:
  Specify the position of the first character of the symbolic key in the data record. The first character position in the data record is counted as zero.
- Key length:
  Specify the key length in characters (max 64).

BIX then scans the data file and copies the specified fields to the workfile, together with the relative keys of the data records. The index records thus built are written to the workfile sequentially, irrespective of the value of the key.

Phase 6

Sort the index records, if necessary. The TOSS utility Sort File (SRT) is run to sort the index records. Sort is not necessary if the data records have been written to the file in key sequence. Sorting is done on the binary value of the symbolic keys. The input file for Sort is the intermediate index file as built by BIX, and the output file is the second workfile.
Parameters for Sort routine:

- Sub-key address in record:
  Answer zero. The intermediate index records start with the symbolic keys.
- Sub-key length:
  Specify the length of the symbolic key, in number of characters.
- Max number of records:
  This must be equal to the number of records on the data file.
- Effective record length:
  Answer zero to sort the complete index records. For duplicate keys the index records will be in sequence of the relative keys of the data records.
- Ascending order:
  Answer yes.

Phase 7 - Build the index file and master index file with RIX

This phase is also needed if a new indexed file structure has been created under SDM.
Utility Reorganize Index file (RIX) is run to build the index file and master index file. Input for RIX is the sorted intermediate file, either created by Sort or by BIX if the data records were already in key-sequence on the data file.

The size of the partitions of the index file is determined by the size of the master index file. Each master index record corresponds to one index partition. The number of records per index partition is equal to:
the number of used index records divided by the total number of master index records (the length of the master index file expressed as number of records).

The following questions will be output by RIX:

- Maximum number of records on the output index file:
  An estimation of the highest record number (the LRN) on the new index file. This can be derived from the LRN of the data file, and the load factor. The space in the new index file after the record indicated by the LRN is reserved for future extensions.
- Load factor:
  A decimal value indicating which percentage of each block in the index file must be used. See the discussion of the load factor in section 8.5.4.

The utility RIX returns the following information to the operator:

- Number of index records per partition: nnnn.

The next question: "OK?" can be answered with Yes if this number corresponds with what was estimated and with the requirements of the application (mainly regarding access times). If this is not so, answer 'No'. The utility will be aborted and a master index file with a different size can be created before running RIX again.

Index records are read from the sorted work file and written to the index file in the required format. Free records are added at the end of each sector according to the load factor.
Records are written to the master index file sequentially. RIX performs a check on the record sequence. If a key sequence error is detected, the utility is aborted and an error message is output on the operator's console.

8.5.3 Master Index File Size

The master index file size influences the performance.

If the master index is very large this may cause memory problems. The search time is not much influenced by the master index size because SDM does a binary search of the master index.

If the master index is too small, there is a large number of index records per partition. This means many disk accesses and sequential search of the index blocks, which will reduce the performance.

A master index file created during runtime will have the size needed to contain the number of master index records equal to the number of records specified for the data file, divided by 8.

8.5.4 Load Factor

SDM does not allow the index files to be extended. When the index file is created, two characteristics specified by the user must allow for future extension of the data file:

- Index file size
  The number of records in the index file must be larger than the number of records in the data file, to allow for index entries with a high key value to be added in the last partition of the index file.

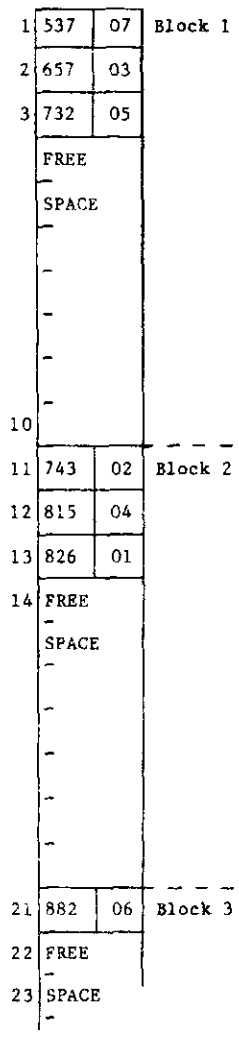
- Load factor
  When new records are written to the data file by the application, the corresponding index records must be inserted in the index file in the correct position. To make this possible free space is reserved in every block. The Load Factor determines which percentage of each block of the index file will be filled with index records when the index file is built.

Example 1

7 Data records have been written to the data file, and utility RIX is run with a load factor of 30 specified. The blocking factor of the index file is 10. RIX will build an index file where every block is filled for 30%.

DATA FILE "NAMES"

Relative Key	Key 2	Key 3		Prime key
1	Clayton	David	M	826
2	Shaw	Patrick	M	743
3	Wilcocks	Brian	M	657
4	Coleman	Jim	M	815
5	Anderson	Ethel	F	732
6	Bloch	David	M	882
7	Watkins	Thora	F	537
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				



The next data record written to the file has prime key 791. The data record gets the relative key 8. The new index entry is inserted in the index file at position 12 and the entries 12 and 13 are shifted to the positions 13 and 14 to make space.

After some time the data file has been filled to 90%. Index entries are inserted in the proper positions. The first block has no free space left, the second block has 40% and the third block has been needed for the highest keys.

DATA FILE "NAMES"

Relative Key	Key 2	Key 3		Prime key
1	Clayton	David	M	826
2	Shaw	Patrick	M	743
3	Wilcocks	Brian	M	657
4	Coleman	Jim	M	815
5	Anderson	Ethel	F	732
6	Bloch	David	M	882
7	Watkins	Thora	F	537
8	Smith	Denis	M	791
9	Lewis	Peter	M	229
10	Williams	Ronald	M	207
11	Berry	Printha	F	888
12	Hillary	Thomas	M	330
13	Richardson	John	M	772
14	Wathke	Phyllis	F	022
15	Hanhurst	Donald	M	647
16	Hartman	Paul	M	863
17	Oswald	Denis	M	251
18	Burket	John	M	596
19				
20				

1	022	14	Block 1
2	207	10	
3	229	09	
4	251	17	
5	330	12	
6	537	07	
7	596	18	
8	647	15	
9	657	03	
10	732	05	
11	743	02	Block 2
12	772	13	
13	791	08	
14	815	04	
15	826	01	
16	863	16	
17	FREE		
18	SPACE		
19			
20			
21	882	06	Block 3
22	888	11	
	FREE		
	SPACE		

1	022	14	Block 1
2	207	10	
3	229	09	
4	251	17	
5	330	12	
6	537	07	
7	596	18	
8	647	15	
9	657	03	
10	FREE		
11	732	05	Block 2
12	743	02	
13	772	13	
14	791	08	
15	815	04	
16	826	01	
17	863	16	
18	882	06	
19	888	11	
20	FREE		
21	FREE		Block 3
	SPACE		

To distribute the free space evenly over the blocks, utility RIX is run again. The data file is now filled to 90%, and for the index file a load factor of 90 is specified. Block 3 of the index file becomes free.

Fig 8-2 Load Factor

If the keys of the next data records that are added to the file have values lower than 888, the index entries are inserted in the free space in block 1 and block 2. But if they have high key values they must be inserted after the last entry and the index records will be shifted into block 3. For this reason, the index file must allow for more entries than would be needed for a 100% filled data file.

Example 2

A file structure is created and only few data records are as yet available. The load factor specified for the index file is 10%. This leaves 90% free space in every sector. After the application has written a number of records to the file and the corresponding index entries have been inserted in the index file, some sectors may still be almost empty and others may have little free space left.

To distribute the free space evenly over the sectors again, the TOSS utility Reorganise Index File must be run.

The load factor specified this time must reflect the new status of the data file. If the data file now has been filled for about three quarters, a new load factor of 70 or 75 may be specified. The index file will be rebuilt and every sector will contain 30% or 25% empty space.

8.6 ENLARGING FILES IN SDM

Only standard files can be enlarged by SDM. Files will be automatically extended by SDM when during Write Sequential instructions the end of the data file (indicated by the LRN) has been reached and the Growth Factor is not zero. Standard files are explicitly enlarged when opened with Open mode Extend.

A new extent is added to the file, with the size indicated by the Growth Factor in the File Parameter Block, rounded upwards to a file extent length which is multiple of three logical sectors and of the block length. SDM does not allow new file sections (on another volume) to be added to the file.

8.7 FILE MAINTENANCE

Discarded data records in an indexed file of S-type can not be re-used. When such a file has been much updated, it is necessary to run the TOSS utility Reorganise Index file (RIX) again, to reallocate the free space available and to update the master index. When RIX is rerun after some time, a new load factor can be specified, representing the real or estimated percentage of used records in the data file.

The TOSS utility RIX must be run in the following situations:

- The message End of File has been returned after a Write Indexed Direct instruction, to indicate that index records in the last block have been written after LRN. When a number of free blocks are available at the end of an index file, the message End of File (Condition Register set to 1) is returned and bit 3 is set in the Status Word each time when a free block is used. This may occur several times before it is necessary to reorganize the files.

- When index records have been shifted into the next partition, the master index file no longer represents a good picture of the index file. As searching of the index file for the specified key is done sequentially, starting at the record pointed to by the master index, the correct record will still be found, but search time increases.

- If the last block of the index file is filled completely the message End of Medium is returned (condition register set to 3) and the Write instruction is not completed. The files must be closed, and BIX-SORT-RIX must be run immediately.

- After the data file has been much updated and consists of many file extents or contains many discarded records, it may be reorganized by copying it into a newly created data file of S-type. BIX, SORT, RIX must then be run to build the index file and master index file for each index.

8.8 FILE RECOVERY

The possibilities for file recovery in SDM are limited. The following points must be taken into account when designing recovery procedures:

For files opened without the Delay option (see section 8.7.2, Delay option), each Write instruction issued by the application results in a disk access and the records are written to the file immediately.

When a file is opened with the Delay option, the information is written to the block buffer, and will only be written to the disk when another block of this file must be accessed, or when the file is closed.

During Write Sequential and Write Indexed Direct instructions, the LRN of the data file is updated in memory and not written to the VTOC on disk until the file is closed.

If a system failure occurs during updating of the file the value of the LRN will be lost. When the file is opened again the old value of the LRN will be read from the VTOC. Used records after the old LRN can be recovered with the instruction sequence:

- Read Sequential up to End of File (the old LRN is reached)
- Read Direct, reading the used record after the old LRN
- Discard Indexed Direct, using the symbolic keys of the record just read. The index entries are deleted from the index files (all indexes must be opened!).
- Write Indexed Direct, using the keys of the record just read and discarded. The LRN will be updated, and new index entries will be inserted in the indexfiles.

8.9 BUFFER MANAGEMENT

8.9.1 Block Buffers

SDM contains internal block buffers, used for the data files and for the index blocks. The block buffers are physically fixed in memory and have a fixed length. This means that they must be long enough to contain the longest block that has to be accessed, and that it may be inefficient use of memory space to have files with very different block lengths.

The minimum number of block buffers is 2 per disk driver.
The maximum number of block buffers that SDM can handle is 16. Which buffer will be used for a request is determined by a Least Recently Used (LRU) algorithm. A block buffer remains attached to a file during an I/O instruction and is released when the I/O is completed, except when the Delay option is used.

When an I/O operation is started and there is no free block buffer, SDM will wait until a block buffer is released.

The number of block buffers and their length is specified during Monitor generation.

8.9.2 Delay Option

SDM supports the Delay option. When this option is specified, a block buffer is attached to the file when the file is opened, and released when the file is closed.

Updates to the file are not written to the disk immediately but to the block buffer. This buffer is written to the disk when another block must be accessed, when the file is closed or when there are no free block buffers available.

The Delay option improves performance when files with a blocking factor greater than 1 are processed sequentially.

One extra block buffer per file opened simultaneously for which the Delay option is required, must be reserved during Monitor generation.

8.10 RETURN INFORMATION

Under SDM, the following return information may be generated:

8.10.1 Status Word

The following bits can be set in the Status word. These error messages are further discussed in chapter 10, Return Information.

bit 0 Request Error
bit 2 End of Medium
bit 3 End of File
bit 4 No Data
bit 5 Key not Found
bit 6 Duplicate Key at Read Indexed Sequential Instruction
bit 7 Retries performed for the disk transfer
bit 8 Data Management rule violated, more information in Return Status
bit 9 Duplicate Key Error
bit 10 New Volume Loaded
bit 11 Protection Error, Rollback
bit 12 Incorrect Length
bit 13 Data Error
bit 14 Throughput Error
bit 15 Disk not Operable

8.10.2 Return Status

In SDM, the Return Status may be set to the following values:

1 Not enough memory
3 Overflow
4 Illegal File Parameter
7 Illegal file code
8 Illegal ECB parameter
9 File name unknown