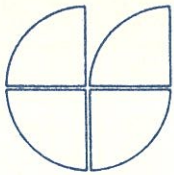


## DATA BASE ADMINISTRATION

### Introduction

In an environment where a data base includes data which is shared by many user programs, it becomes necessary for a data base definition to be developed centrally. In such a shared environment, a data base is, in a sense, a compromise between the needs of the various user programs, and the proper trade-offs can only be made centrally by a "Data Base Administrator".

On the basis of information as to the data required by individual programs, statistics on usage of data, and response requirements, the "Data Base Administrator" must make decisions; for example, on whether to repeat data redundantly and on what relationships to build into the data base. Based on these decisions it must declare the areas, records, data items, and sets required, and if necessary, restructure the data base.



## Functional Overview

The Data Base Management system consists of three principle functions, the Requestor, the Data Base Administrator (DBA), and the Data Base Manager (DBM). The Requestor is the individual desiring to use the data in the system. The Data Base Administrator defines the rules which control the access to the data and determines the manner in which the data will be stored. The Data Base Manager is a combination of hardware and software functions that give access to the data as instructed by the Requestor and Data Base Administrator.

The Requestor communicates with the DBA through formal and informal channels. The Requestor establishes a need and authority to access data. The DBA provides system authorization and an access strategy for the Requestor which is consistent with the overall use of the data base.

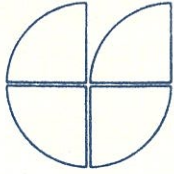
The DBA communicates with the DBM via a Data Base Description Language (DBDL), a Data Base Management Language (DBML), and a set of system utilities. The DBDL provides a formal and precise definition of the data, how it is stored, and how it is associated. The DBML provides a series of commands to the DBM to access the data. The DBA commands the DBM at both levels; namely the descriptors of the data and the data itself. The utilities provide the DBA with facilities for protecting, transforming, and manipulating data.

The Requestor communicates with the DBM via the DBML. Every action taken on the data base by the Requestor must have been authorized explicitly or in broad terms by the DBA.

The Requestor has primary responsibility for specifying the retrieval of data from the data base. He does this as part of his task of responding to management requests for information about the enterprise. Typically, this is accomplished through the writing of an application program. Those specific statements within the program that reference logical data will be drawn from the DBML which is designed to operate on the logical or application-related data. These statements are in turn interpreted by the DBM. This technique allows the physical nature of the data to be changed by the DBA without affecting the application programs, thus achieving the objective of data independence.

The Requestor refers to the system designer or application programmer who is acting as an agent for a user who has a need for data. This data may or may not be present in the data base. If the data is available, the Requestor presents his needs to the DBA. In addition, he would provide any special parameters needed to characterize his requirements. The DBA then has the responsibility of making any modifications to the data base to insure that this request can be materialized. The DBA would be responsible in seeing that this relationship is maintained as long as the need for such is present. If the data is not available, then the DBA would review with the Requestor the data to be added, the rules by which this data is to be maintained, and the

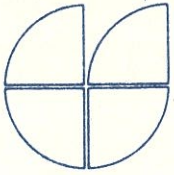




logical requests that are to be made on this data. The Requestor may also state any security rules that are to be applied. The DBA would then define to the system through the DBDL the necessary physical and logical descriptors associated with the Requestor's needs.

The Data Base Management Language (DBML) is the facility which provides the Requestor with the ability to manipulate logical data according to his requirements. Using the DBML, the Requestor asks only for the data that he needs. The access methods employed and the physical data are completely transparent to the Requestor.

The DBML contains certain commands which can be executed without further intervention by the Requestor. These commands are: OPENM(V), READM(V), ADD-M(V), DEL-M(V), and CLOSM(V).



### General Description - Data Base Administrator

The function of the DBA may be diffused in some portion of the user group or may exist as a central staff. However, the separation of this function is key to the success of a Data Base Management System. The complexity of modern data structures precludes casual attention to the critical factors affecting the storage and retrieval of data, and the proper administration requires broad knowledge of the use of data throughout the installation. Such knowledge seldom exists when an individual is deeply involved in a particular application area. In addition, the technical skills of system measurement and optimization require advanced training not feasible on a broad basis.

The DBDL is a language for the DBA function. The language allows the DBA to:

1. Describe the physical and logical attributes of data.
2. Define the relationships that exist between units of data.
3. Define the rules for the DBM to use in materializing logical data from physical data.
4. Define the rules by which the DBM is to provide security/integrity of the data base.
5. Define the rules for system performance measurement.

In describing data, the DBA will give each unit of data a name.

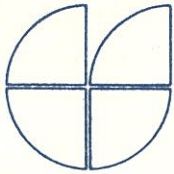
Using statements available to him in the DBDL, the DBA describes record types and the logical and physical data associated with them. The record fields, record types, and record relationships define the association between logical data structures and their representation in physical data structures.

Descriptive statements are required for physical data structures and logical data structures. Those required for physical data structures describe data sets, stored records, physical relationships, and data elements. Those required for logical data structures describe files, logical records, logical relationships, and data items.

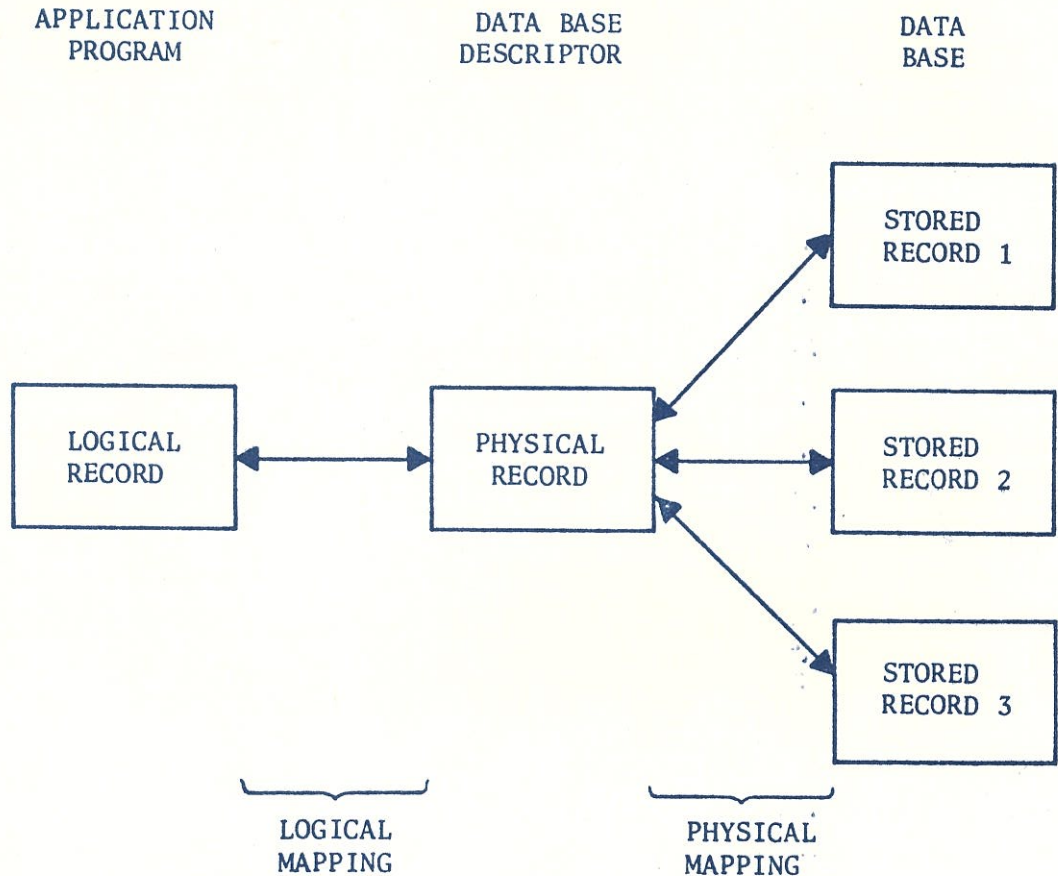
The DBA submits these DBDL statements to the DBM which derives the proper descriptors. These descriptors are then maintained in the Data Base Directory and controlled by the DBM in much the same way that data is maintained in the data base except that only the DBA has the authority to access and maintain this directory. In fact, the directory itself is a data base with its own prespecified descriptors that require a high degree of authority (i.e., the DBA) for access.

The DBA will need to define the relationships that exist in logical and physical data structures. For the physical structure, these statements describe the relationships that are maintained to facilitate the DBM's manipulation of physical data.



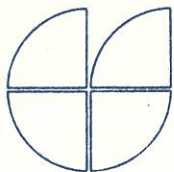


For logical relationships, the statements describe the relationships between logical records. They are defined to aid the application program in the manipulation of logical data.



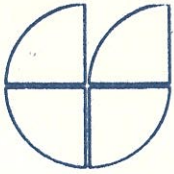
The physical mappings refer to access strategies to be used in manipulating physical data. Different strategies may exist for data retrieval versus data modification. The logical mappings will define logical structure as a subset of the associated record types, relationships, and constructs. The level of indirection created by this use of the data base descriptor concept provides the environment of data independence.

The DBA will specify the rules by which the DBM will perform data protection. The concept of specifying security rules implies that the security system has more than one dimension. At least two dimensions must be recognized. These are the right-to-know and the need-to-know.



In order for the DBA to evaluate the efficiency of operation of the Data Base Management System, he must have available to him pertinent data about how the system is functioning. To accomplish this task, he will take advantage of facilities available in the DBM. He will decide which functions and resources of the system he wishes to have measured and the manner in which this data is to be obtained.





## Practical Description - Data Base Administrator

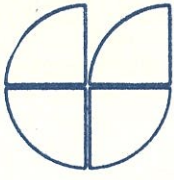
With the ever-increasing use of data base, the need for a Data Base Administrator is greater than ever before. Cincom Systems has compiled some ideas concerning the function of the DBA.

We feel that, used as a guideline, these ideas may help to develop some type of standardization in the area of DBA, as we have done in the area of Data Base Management. Whether we have a Data Base Administrator or a Data Base Administration Group would, of course, depend entirely on the size and needs of the particular installation. The function would remain the same.

The ideas presented here have come, in part, from TOTAL users through discussions at TOTAL Workshops. As these thoughts are expanded and added to, we will provide updated versions of this document.

The areas most important to the DBA function are presented in the following outline:

1. Maintain complete control over the definition of, and generation of, the Data Base Descriptors
  - a. Gather file and data needs of data base users
  - b. Control coding of data base definition
  - c. Distribute copies of definition to those who need it, specifically Project Leaders
  - d. Have complete control over changes made to Data Base Descriptors
2. Have responsibility, at least at the logical level, for the physical set-up and distribution of data files on the physical devices
  - a. Coordinate file needs with operations
  - b. Control best set-up for optimum performance
3. Coordinate user groups who are using a common data base
  - a. Eliminate redundant data
  - b. Establish security when needed
4. Assist users in determining best structuring of files, relationships, and data content of the data base
5. Provide standard utilities for use with data base
  - a. Dump/Restore routines
  - b. Standard file-load routines
  - c. File-list programs
  - d. Etc.

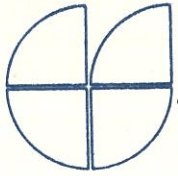


6. Maintain a Data Base Librarian

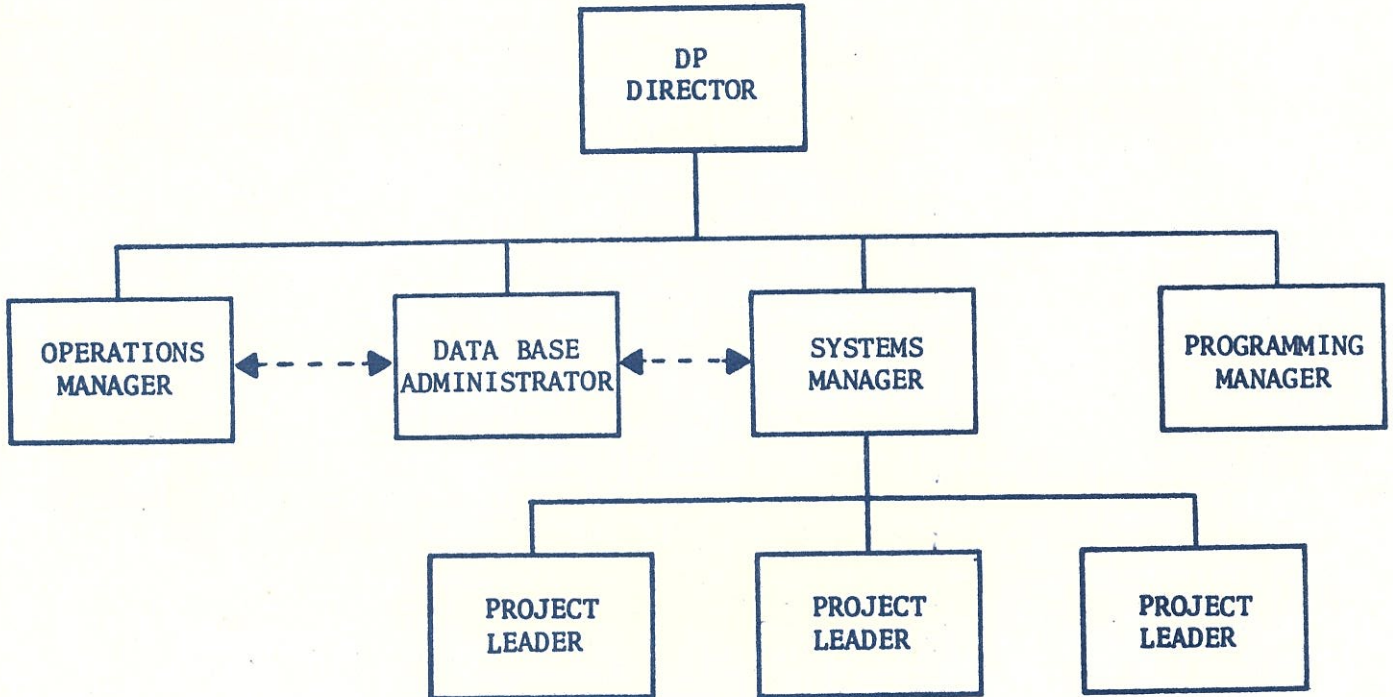
- a. Good security tool
- b. Aids in control of changes to data base

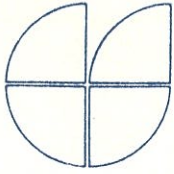
Attached are schematics which may be useful in establishing a Data Base Administrator position.



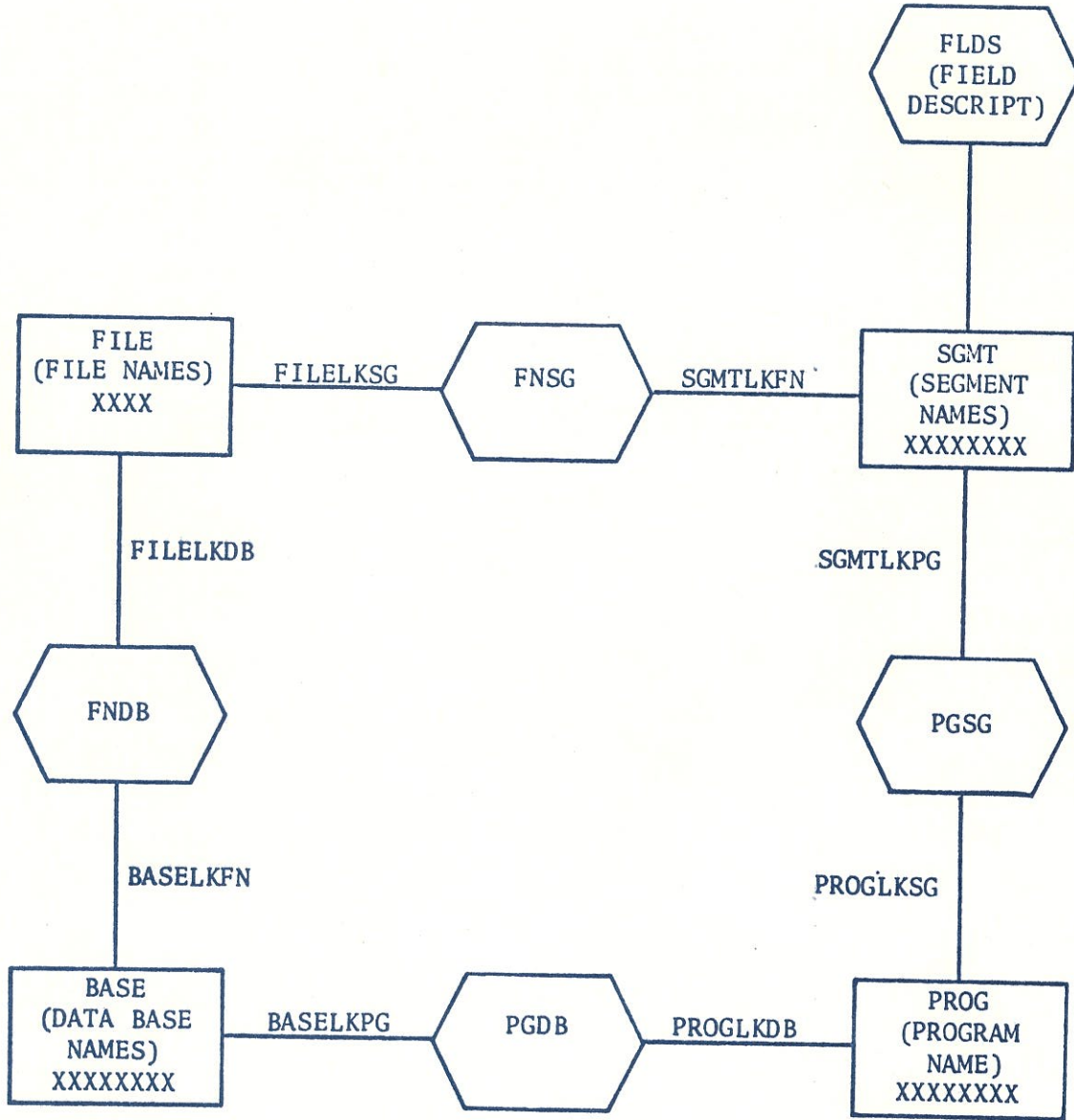


FUNCTIONAL STRUCTURE FOR DATA PROCESSING GROUP

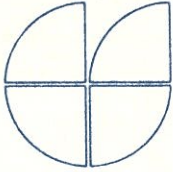




DATA BASE LIBRARIAN







### TOTAL Sequential Processes

The attached diagram shows the various sequential processes that can be executed in relation to each TOTAL file.

The Sequential Format contains a four byte work field (DBADDR) followed by each data element in the same arrangement as in the corresponding physical TOTAL file. Application conversion and sequential processing programs must provide for the DBADDR field but should ignore its contents. It is used only during the High Speed Load/Update but is included in all layouts to minimize the number of programs that need be written.

Application Conversion Programs - Your application's conversion programs must read your old data, do any necessary conversion, editing, reformatting, etc., and create the Sequential Format with the leading four byte field called, DBADDR. DBADDR may contain anything at this point.

Application Sequential Processing - Your application's sequential processing programs should be set up to read the output of the Sequential Dump procedure. This output is in the Sequential Format and is in sequence by the major CTRL field.

Application Sequential Update - If you desire to sequentially update a large percentage of the records in a TOTAL file, it is suggested that you process against the output of the Sequential Dump function while writing your updated records into the Sequential Format on tape. These should be put through the request location (RQLOC) process and, instead of going through the Load program, should pass through your program which reads the Sequential Format and updates the TOTAL file.

DATA  
IN OLD  
FORMAT

DMFS

TOTAL SEQUENTIAL PROCESSES

