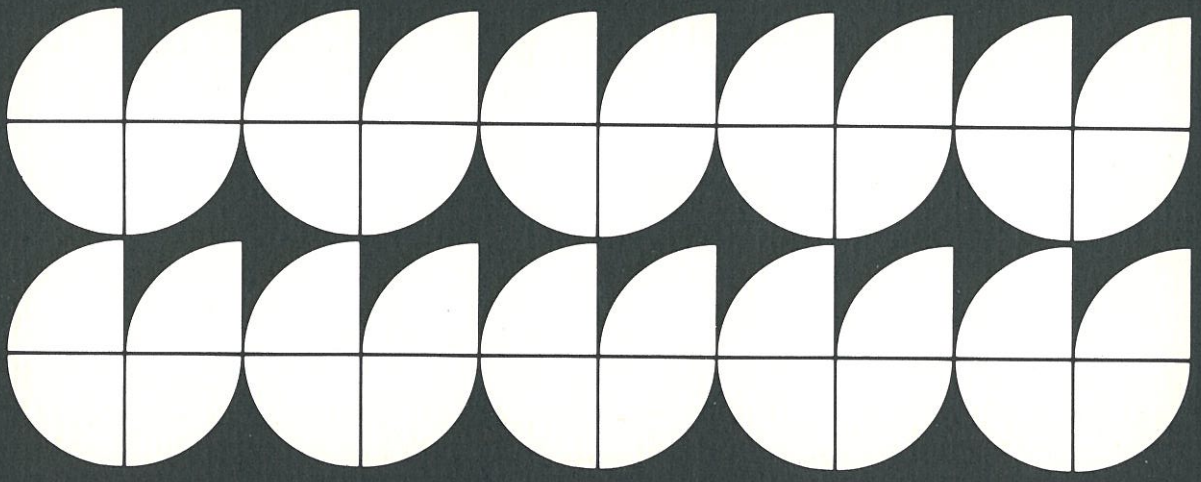
the "TOTAL" data base management system



THE COMPANY

Cincom Systems, Incorporated is a company dedicated to providing the highest possible quality computer support services and proprietary programs to data processing users.

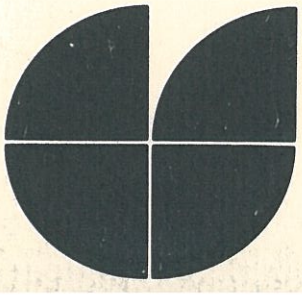
Founded in 1968, Cincom is presently divided into two service divisions: a Custom Systems Division and a Proprietary Systems Division.

The areas of responsibility within the Proprietary Systems Division are the conception, design, implementation, marketing, support, and maintenance of all proprietary systems products. Cincom specializes in those systems software products which are considered as tools with which the user may more easily and surely build his own specific applications. The primary emphasis and background of Cincom is in the area of Real Time On-Line, and Integrated Data Base Systems. Products currently available are:

a) TOTAL — the most advanced and widely accepted integrated data base management system in the industry

b) ENVIRON/I — the terminal monitoring and task management communications control system

This combination of products provides data base/data communications system availability for the IBM 360/370 DOS user, as well as a very powerful combination for the OS user.



Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Evolution of the Data Base Concept

The term Data Base is not a new one. Data Bases have existed in our companies and businesses from the very beginning the company was formed. Originally the data base or information base was typically in file cabinets and index files. A data base did truly exist from the beginning.

With the introduction of computers and accounting machinery into our companies, our data bases were then transformed into a form understandable and legible to this equipment, *files of information.*

In unit record, first and second generation computers, and in many third generation computer installations, programmers and systems people attempt to create and maintain control over their own data records and data files (data sets). These data are usually developed along functional or departmental boundaries and accessed on an application-by-application basis.

As result of this type of approach, early attempts at data base had two (2) distinct attributes: (a) no *integration* of information (b) much *redundant* information.

In this traditional approach, output from one application becomes input to another. Sorting, merging, and other associated techniques are used to structure and relate our data records and data sets with each other into some meaningful order. This approach is application oriented and suited only to *batch processing* environment. The concept of a *transaction processing* system was not economically feasible.

Systems designed with this approach to data base were not readily available to change, whether it be trivial or major. The *cost of change* would prohibit, in most cases, any changes to these systems.

With direct storage devices, such as disk storage drives, file organization techniques are used to manage a data file or data set. These file organization techniques are usually vendor supplied and typically include sequential organization, a random organization and a variety of sequential organizations with indexed access (hereafter called indexed).

Many times the terms file management and data file or data management are used interchangeably. This interchangeable usage is generally considered valid.

However, to interchangeably use any of these terms with the term Data Base Management is an error. This single error in thinking is responsible for most of the failures in pursuit of the Management Information System (MIS).

Disk file organization and file management is concerned with the management and control of data on a file by file basis. It is clear that the MIS concept which would dictate that all relevant and related information from many other data sets also be available simply *cannot* be achieved with conventional data management techniques.

The requirement is that a true Data Base Management System be used which will manage virtually an unlimited number of data sets on an "integrated, non-redundant" basis and which will allow for entry and association of each of these data sets with any other data set in the data base. This is precisely the function of TOTAL — the Data Base Management System.

TOTAL SYSTEM TERMINOLOGY

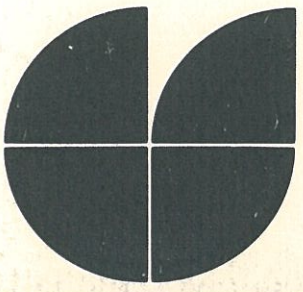
DATA FIELD – STANDARD UNIT OF INFORMATION.

DATA ELEMENT – A UNIQUE DATA FIELD OR A GROUP OF COMMONLY USED DATA FIELDS. THIS IS THE MODE OF COMMUNICATION WITH DATA UNDER 'TOTAL.'

DATA RECORD – A GROUP OF DATA ELEMENTS RELATING TO ONE OR MORE PARTICULAR IDENTIFICATIONS (KEYS).

DATA SET – A GROUP OF DATA RECORDS REPRESENTING DIRECTLY & SPECIFICALLY RELATED INFORMATION.

DATA BASE – A GROUP OF DATA SETS REQUIRED TO PROCESS A FAMILY OF APPLICATIONS.



Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Total System Criteria and Philosophies

The design philosophy and criteria of the TOTAL system provides certain very handsome benefits. Certain of these criteria are also felt to be essential requirements of a data base management system by such groups as CODASYL, GUIDE and Share.

Modular & Evolutionary in Design & Use — Significantly *reduces* the costs involved with change; major or trivial. Also relieves the "paralysis-by-analysis" syndrome too evident in systems design.

Data Independence — Application programs are sensitive only to data elements (fields), not records. This then allows no recompilations of old programs due to physical data changes (records, files, relationships, etc.).

Data Non-Redundancy — Elimination of the so-called "multiple versions of the truth". As a benefit of this, less DASD space is required to retain the data base. Significantly reduces the cost of maintenance and lessens the chance for inaccuracies.

Data Relatability — Allows the data base to be structured the way your company does business and at the same time provides non-redundant information to be a realistic thing.

Data Integrity and Security — Vital to the data base. The data base tends to become one of your company's "most valuable assets". Prohibits destroyed files and records by application programs and unauthorized entry to and from the data base.

Equipment, Language & Environment Independent — Supplies to you a major level of independence in such vital areas as operating system and computer, DASD devices, application programming language used, and your processing approach; batch or on-line. This *conversion as a way of life* problem is virtually eliminated from your data base viewpoint.

Optimum Performance & Efficiency — Traditionally, these two very vital factors for a successful system tend to oppose each other. We feel the TOTAL integrated data base approach allows you the benefit of *both* of these, which is vital.

The TOTAL system *does* encompass each of the afore mentioned design criteria. It also offers a number of other facilities and advantages which are discussed in this brochure.

Concept of Total

The TOTAL system embodies a "network structure" philosophy. There is *no* hierarchial overhead. Indexes, directories and overflow areas are eliminated. Relationships from one data set may be made on a *direct* basis to 2,500 other data sets within the data base. Up to 65,000 data sets may be managed on an integrated basis within one data base. It is possible to establish a virtually unlimited number of hierarchial levels or direct network structured relationships.

There is no limit to the number of data bases which may be developed, and any data set may be included in any number of different data bases. The TOTAL system features the highest possible performance in the areas of disk and core utilization and in retrieval speeds. Further, the system is self-optimizing, which prevents the usual performance degradation.

The TOTAL system, is in effect, two systems in one:

a) a system which provides for the initial generation of a data base module and all subsequent modifications and expansions to this data base in an English-like language. This phase is called Data Base Definition.

b) a system which interacts with this data base, the operating system or supervisor and the application programmer. In this phase, TOTAL functions with the host language (COBOL, PL/1, etc.) for all types of communication with the data base. This phase is called Data Base Manipulation. At all times in both phases, you, the user, are in complete control with TOTAL.

Major advances in such integrated data base problems as *data integrity* and *security, concurrent updating, data independence* and *restart* and *recovery* are facilitated by TOTAL.

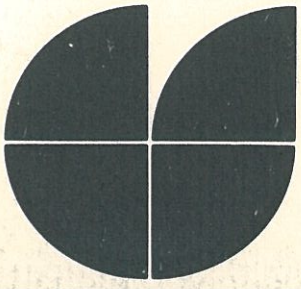
Data Base Definition (DBDL)

TOTAL DBDL is the language used to describe and declare the data base system, and/or subsystem data bases within the overall data base. This is done in terms of names and types of data sets, data records, and data elements; and the data set and data record relationships required at this time. After definition, the data base is compiled and cataloged.

The TOTAL DBDL is an extremely high level, English-like free form coding. A user can be fully trained in a matter of hours and he will then be capable of defining and developing data bases of great complexity.

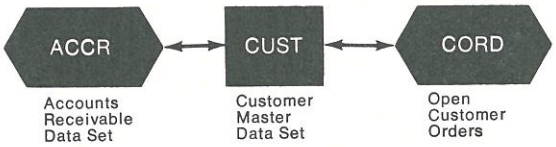
In order to modify either an operational subsystem data base or the entire data base, it is only necessary to define the changes, recompile the data base and recatalog. The computer time is but a matter of minutes.

After compilation and cataloging of this data base, the user is ready to begin application programming using his choice of host language processors such as COBOL, PL/1, Assembler, etc. and the TOTAL Data Manipulation Language (TOTAL DML).



Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Illustrative Total Network Structured Data Set Relationship For Simple Order Entry and Accounts Receivable System

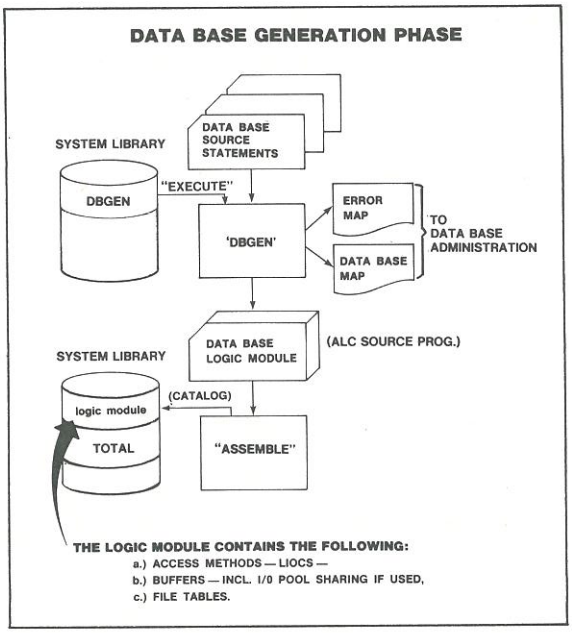


Sample Data Base Definition Using the TOTAL DBDL

The Customer Master File (CUST) and each data set to be included in this data base would be described in TOTAL DBDL in this manner:

DATA ELEMENT	PROCESSING NAME	DIGITS	DBDL CODED ENTRIES
Customer Number	CUSTCTRL	6	BEGIN-DATA-BASE-GENERATION:
Customer Name	CUSTNAME	30	DATA-BASE-NAME=xxxxxx
			BEGIN- MASTER- DATA-SET:
Customer Address	CUSTADDR	30	DATA-SET-NAME=CUST
Customer City & State	CUSTCTYS	20	MASTER-DATA:
Customer Zip Code	CUSTZIPC	5	CUSTROOT=8
Special Credit Terms	CUSTCRED	6.	CUSTCTRL=6 CUSTOMER NO.
—	—	-	CUSTLK01=8=CORD LINK TO OPEN ORDERS
—	—	-	CUSTLK02=8=ACCR LINK TO ACCTS. REC.
Any Other Elements Required		-	CUSTNAME=30 FULL NAME
—	—	-	CUSTADDR=30
—	—	-	CUSTCTYS=20
—	—	-	CUSTZIPC=5
—	—		CUSTCRED=6
Customer Credit Limit	CUSTCLIM	4	CUSTCLIM=4 PACKED AMOUNT (xxxx.xx)
Accts. Rec. Balance	CUSTARBL	4	CUSTARBL=4
Outstanding 60 Days	CUSTAR60	4	CUSTAR60=4
Outstanding 90 Days	CUSTAR90	4	CUSTAR90=4
			END-DATA:

Data elements, processing names and digits are under complete control of the user. There is no limit to the length of elements, the number of elements or the alpha numeric or special character mix.



In the sample data base generation run shown above, the time involved to perform that run would be, for example on a 360 Model 30, approximately forty (40) seconds of actual computer time.

At the conclusion of the 'DBGEN', the logic of your data base has been established and catalogued on your system. You will note also that a module named TOTAL is also resident on this library. These two modules now constitute the *complete* TOTAL data base management system necessary for you to start developing user applications.

the "TOTAL" data base management system

Concept of the TOTAL Data Manipulation Language (DML)

TOTAL DML is a language which the programmer uses to communicate between his program and the data base. TOTAL DML is not a complete language by itself. It relies on a host language to furnish a framework and to provide the procedural capabilities required to manipulate data in primary storage.

TOTAL DML functions in conjunction with the host language, such as COBOL, PL/1, Assembler, etc. at the CALL or MACRO level. The user's application program then is a mixture of host language commands and DML functions. TOTAL DML interacts with the data base. It is the manipulative language for the data base.

All calls to and from the data base to retrieve data, to add new data or data relationships, to delete data or data relationships, to modify data or data relationships are defined in TOTAL DML.

TOTAL DML features comprehensive safeguards and analytical capabilities to assure proper processing. Diagnostics and messages are provided which indicate the successful execution of a function, or the status in case of an unsuccessful execution. For example, TOTAL DML will indicate that a duplicate record already exists if the user attempts to add such a duplicate record to the data base.

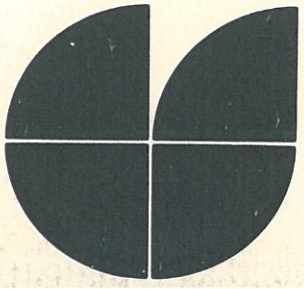
TOTAL functions at the element level. An element is defined as one or more of the "fields" that comprise a logical record. Upon the execution of a TOTAL DML command, one or more elements as specified by an element list are passed to or from the user host program in the stated sequence of that element list. The user is not required to do any further manipulation as to sequencing, positioning, inclusions or omission of elements. TOTAL features an extremely powerful and comprehensive repertoire of data base manipulation commands.

Since the TOTAL system is capable of manipulating data at the element level, subsequent expansions of the record for additional elements or relationships have no adverse effect on programs which use the originally defined record. Old programs do not require recompilation when new elements are added to records. After the host language program has received the data, the application programmer uses the host language for whatever logical arithmetic, or manipulative processing he wishes.

The host language, then, is the language of specific data "policy" manipulation. This manipulation is application oriented and very specific. TOTAL DML functions within the framework of the host language to provide specific information to and from the data base.

TOTAL DML is the "language of the data base". TOTAL DML functions and host language statements are intimately mixed in the user application program.

TOTAL, utilizing the facilities of the DBDL and the DML, provides a completely integrated data base which is available to any application programmer, any of whom may be using different host language processors. TOTAL provides data elements to these programs in such a fashion that new elements, new data sets and new data relationships may be added to the data base, in order to respond to changing requirements, without adversely affecting the current operational programs.

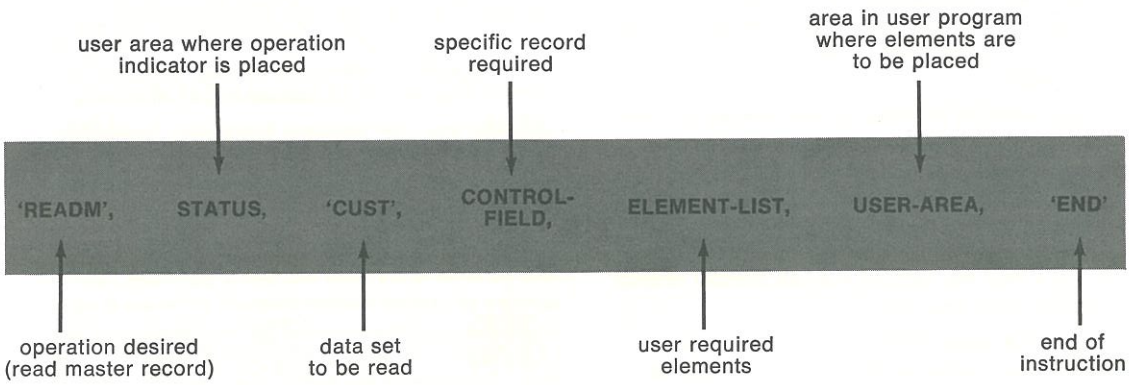


Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Sample Total DML Statement As Used in Cobol

The following parameter lists must be given to a 'CALL' statement according to the rules of the calling language. The parameters enclosed in quotes (') must be presented to the Call statement as literal values, other parameters must be defined as storage areas to be used by the user program and TOTAL.

CALL 'DATBAS' USING



At the completion of this statement, the specific record elements requested for Customer No. 123* are placed in the user defined areas precisely as desired by the application programmer. It is not necessary for him to perform any extractive and move functions.

The STATUS area will contain **** which indicates a successfully completed operation. The control is back to COBOL and the user is free to perform whatever operations he wishes.

**Assumes that Customer No. 123 was set up as the desired key in CONTROL-FIELD*

ref: diagram on page 10

Types of Data Sets

In order to effectively handle the many varieties and categories of types of data, TOTAL provides two types of files or "data sets" — Master and Variable Entry. A brief discussion of the characteristics of these data sets follows:

Master Data Sets — These data sets are organized and managed according to any user-selected primary control field. These control fields are unrestricted as to length or content.

Within each record in the data set the number of "data elements" and their individual sizes are limited only by the size of the logical record itself. Master data sets are relatively stable and predictable as to record content and number of records. They may be thought of as existing because the "company is in business". Each record within the data set may be directly related to up to 2,500 other data sets.

In normal maintenance TOTAL Master Data Sets are self-optimizing and *never* require reorganization. As a record is deleted the space is immediately available for reuse by the system.

TOTAL Master Data Sets are reloaded only if the user wishes to reformat the basic record or if the physical space extents are exceeded. In both instances, only the affected data set is mounted on the disk storage drives.

All other data sets in the data base are unaffected and do not require reorganization, processing or updating of any type. This is true irrespective of the number of data sets that may be directly related to the Master Data Set being reorganized.

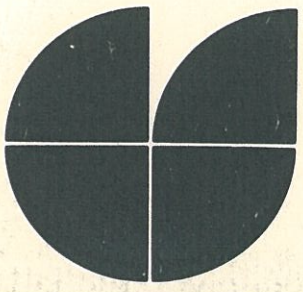
Data retrieval performance is extremely high and does not degrade as the number of relationships to other data sets increase.

Variable Entry Data Sets — Variable Entry Data Sets are data sets which may be entered by a variable number of control fields. Further, they contain a variable (and highly volatile) number of data records. The data records may have variable formats and variable numbers of record types within each format. The data sets and data records may have unique relationships based on user-specified attributes.

It can be said that Variable Entry Data Sets exist because the "company is doing business". They reflect the business functions as they occur and are interactive and interrelated between Master Data Sets.

There may be up to 2,500 different record types within one variable entry data set. Each type may have any number of unique relationships to other data sets. There are no restrictions to the number, size, content, or relationships or records or data elements other than those imposed by hardware and operating systems considerations.

Just as in Master Data Sets, Variable Entry Data Sets do not require reorganization due to performance degradation. The system is self-optimizing and always functions at peak performance. If the read arm is already on the proper cylinder, the seek will not be performed. If the record is already in core storage, the disk read will not be performed. There are no reserved cylinders for indexes or directories. All disk space is available to the user for storing his prime data.



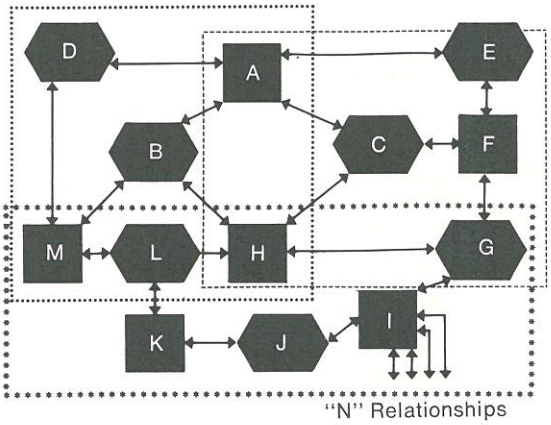
Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Total Network Structure

Below is illustrated a "TOTAL Network Structured" data base which is made up of several integrated data bases.

It is important to note that the evolutionary concept of TOTAL makes it easy to add new data to records, new data sets and data relationships without impacting the present systems. The illustrated data base could grow over time to include hundreds of data sets and hundreds of new relationships, with little if any impact to operational programs.

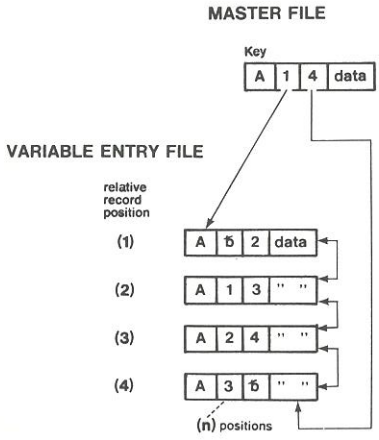
TOTAL — NETWORK STRUCTURE *(Data Base "A1")*



- The data sets within the . . . comprise a unique data base.
  The date sets within the - - - comprise a unique data base.
  The data sets within the * * * comprise a unique data base.

Note that several data sets appear in more than one data base and that data set "H" appears in all data bases so that any change to data set "H" by any program from any data base will immediately be "known" by all other programs.

NETWORK STRUCTURE (SIMPLE)
USING CHAINING/THREADING SCHEME



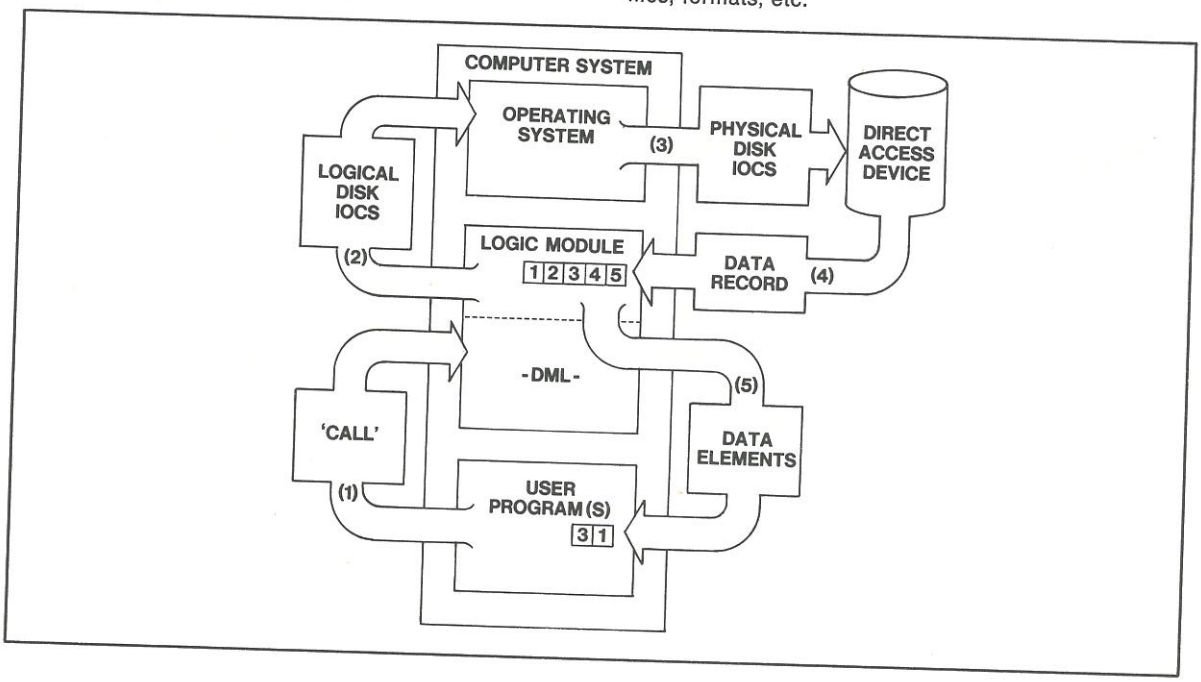
- ALL THREADS—BI-DIRECTIONAL
- RECORD-TO-RECORD THREADS; NO INDEX, TABLE, ETC.

the "TOTAL" data base management system

Data Independence

In order to achieve a level of modularity and an evolutionary approach toward growth and change, the *data independence* feature is of vital importance. As discussed earlier, it is that ability to remove *any* and *all* physical dependency by an application program on data. By communicating at the data element level, we do achieve this independence.

A pictorial description of this is demonstrated below. The modules in the center represent the TOTAL system. As denoted, a record containing five (5) data elements was read into TOTAL by the Operating System. This record could possibly be 150 characters in length, but the user program shown only requested and received elements 1 and 3 but in *reverse order!* This would constitute only thirty (30) characters of data. With this approach the user program is sensitive only to those elements that it requires, not in records, files, formats, etc.



Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

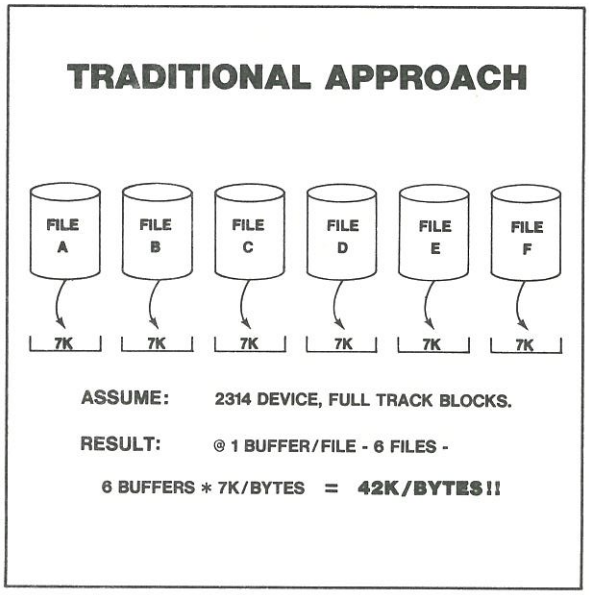
Total I/O Pool Sharing Feature

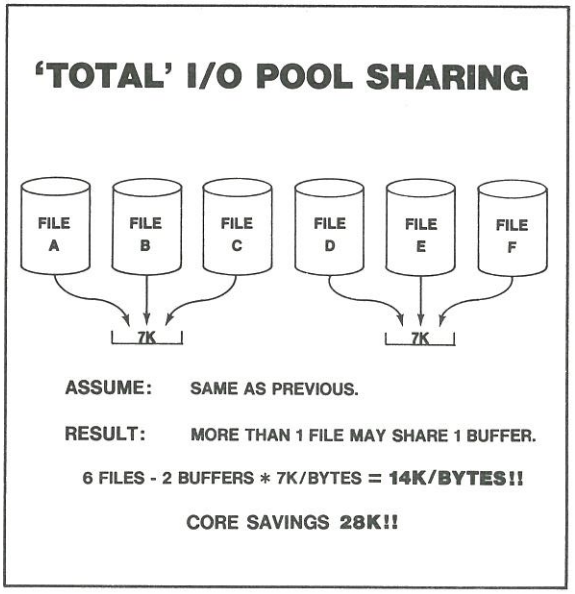
Optimum performance and efficiency is one of the TOTAL system's strongest advantages or benefits to the user. Traditionally these two vital attributes of any system tend to be mutually exclusive of each other. Often the move to data base means significant increases to computer requirements. This is not so with TOTAL.

The TOTAL system helps bridge this gap with the POOL SHARING feature. An example that is common in describing this topic is that of having *large buffers* for each file so that we achieve higher density recording of data and a better performance curve.

As depicted in the example at the left, this would mean large core requirements for such an approach. What is accomplished here is a high level of performance, but efficient use of core is the trade-off.

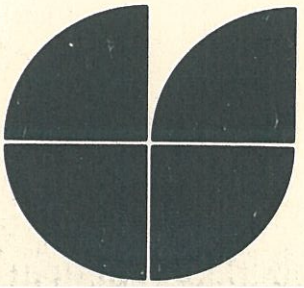
By invoking TOTAL's concept of POOL SHARING, the example at the right shows the same physical file formats but by utilizing *more than* one file for the same buffer, we see a significant gain in efficient use of core while maintaining the same disk utilization, and in most instances with little trade-off in throughput.





Total Fact Sheet

Version	Root Core Requirements	Features
1. IBM/360/DOS/OS MFT or MVT	8K	a) supports multi-programming
2. RCA Spectra/70 TDOS/DOS	8K	b) provides "read only" for read-write protection
3. Honeywell 1200 Series MOD 1 MSR OS-200	13K	c) may be used for *batch* or *on-line type* systems
		d) prevents concurrent updating, yet supports concurrent accessing of same files
4. IBM/360-370/OS MVT or MFT MULTI-TASKING/CENTRAL	13.5K	a) supports full multi-tasking *real-time* and batch systems
		b) core resident *once*
		c) dynamic logging of data base
		d) restart and recovery facilities
		e) *data reservation* vs. ENQ-DEQ at the user record level
		f) horizontal and vertical buffering
		g) data lock sensed and resolved
		h) full concurrent processing
		i) supports "own code" facilities

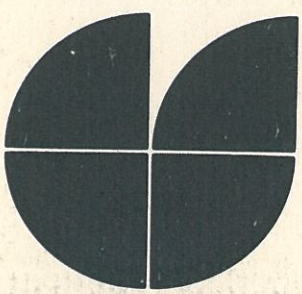


Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300

Nine reasons why Total is one of the most widely and successfully used data base management systems in the country

This TOTAL approach to data base management and data base manipulation — whereby all communication with the data base is separate from the procedural code of the host language — is of critical importance in the implementation of the MIS. This separation of data base manipulation and host language procedural coding provides several major and significant break-throughs and advantages to the user. These advantages are:

1. A common data base can be created and maintained, available to all applications but not adversely affected by changes to various individual applications.
2. All communication with the data base, (such as reads, writes, adds, deletes) functions in a standardized approach which provides the highest levels of performance, integrity and flexibility — but allows the application programmer complete control and freedom with respect to the procedural logic of his host language application.
3. All data management programming effort and testing is *eliminated* as a function of application programming. This effort can account for up to 80% of all programming effort in integrated data base environments.
4. Evolutionary data base and systems design capabilities are achieved. For example, new file relationships, and data elements can be added to data records which are used by operational programs without impacting these old programs. It is not necessary to recompile old programs even though new data elements are added to data records which are used by the old programs.
5. Significant hardware enhancements are gained by TOTAL features — two examples are 1) I/O Area Pooling significantly reduces core requirements while at the same time increasing effective disk storage capacity and performance — 2) elimination of all external directories and indexes significantly speeds retrievals and maintenance and provides more useable disk space for prime data.
6. Duplication of data files and redundancy of data is eliminated.
7. Unique flexibility and independence is gained in such vital areas as host language, types of secondary storage devices used, and operating system types and release levels. Conversion as a "way of life" is eliminated.
8. User retains control over structuring of data files, data relationships, physical placement of data files, etc. but is freed of this major programming effort.
9. The approach of Data Base Security and Integrity is so comprehensive that it is virtually impossible for programmers to blow files, break chains, lose records, or make the many other errors which can adversely impact operational systems. In addition, with comprehensive data base logging and restart facilities, if a power failure or some other physical problem occurs, full restart and recovery of your data base can often be achieved in a matter of minutes.



Cincom Systems Incorporated • 2300 Montana Ave. • Cincinnati, Ohio 45211 • Phone 513/662-2300