INTRODUCTION

TO

MENTOR SOFTWARE

Pub. #517-30201
July, 1983

ADDS
Applied Digital Data Systems Inc.
A Subsidiary of NCR Corporation

7/83

This version of the MENTOR Introduction Manual supersedes the last issue dated 8/81.

Future revisions will have a list of changed pages below, with a margin bar on affected text pages to indicate these changes.

Users are urged to use the Comment Sheet at the back of the book to communicate their comments to the ADDS Technical Publications Department, Systems Division.

ADDS
Applied Digital Data Systems Inc.
A Subsidiary of NCR Corporation

TABLE OF CONTENTS

SECTION PAGE

LIST OF TABLES

LIST OF FIGURES

PREFACE

This manual describes many of the features available through the software and operating system of the family of MENTOR Computers.

Section 1 contains an overview of MENTOR software and the operating system. Sections 2 through 9 describe system software and applications in greater detail.

Other software reference manuals in the MENTOR Computer System series include:

MENTOR CalcuMentor Reference Manual

MENTOR DATA/BASIC Reference Manual

MENTOR DocuMentor Reference Manual

MENTOR EDITOR/RUNOFF Reference Manual

MENTOR ImpleMentor Manual

MENTOR INFO/ACCESS Reference Manual

MENTOR Office AugMentor Reference Manual

MENTOR Operator's Manual

MENTOR PROC Reference Manual

MENTOR System Reference Manual

SECTION 1:

INTRODUCTION TO DATABASE MANAGEMENT ON MENTOR

HANDLING INFORMATION TODAY

One of the biggest problems facing business today is how to handle the massive amounts of data needed to run an organization effectively. It seems that more information is available than ever before -- and more of it is necessary to make smart business decisions.

Many companies are turning to database management to help handle information. More and more computer systems use database management software, allowing users to store and retrieve data in several ways.

The concept of a database management system (or DBMS) will be expanded upon in this section. A database can be defined as a collection of interrelated data, stored in a fashion that permits many users to retrieve, sort and list it.

A computerized database management system can be defined as the software used to store, retrieve and reference information and to process data. The DBMS can also help maintain organizational standards by coordinating data access, security, modification and development. The ultimate goal of a DBMS is to save time and money by making the development of computer applications easier, faster, less expensive and more flexible.

DATABASE MANAGEMENT WITH MENTOR

The ADDS MENTOR Computer System is a powerful business tool. The MENTOR System is optimized to handle business information, so it offers you more power than other computerized DBMSs.

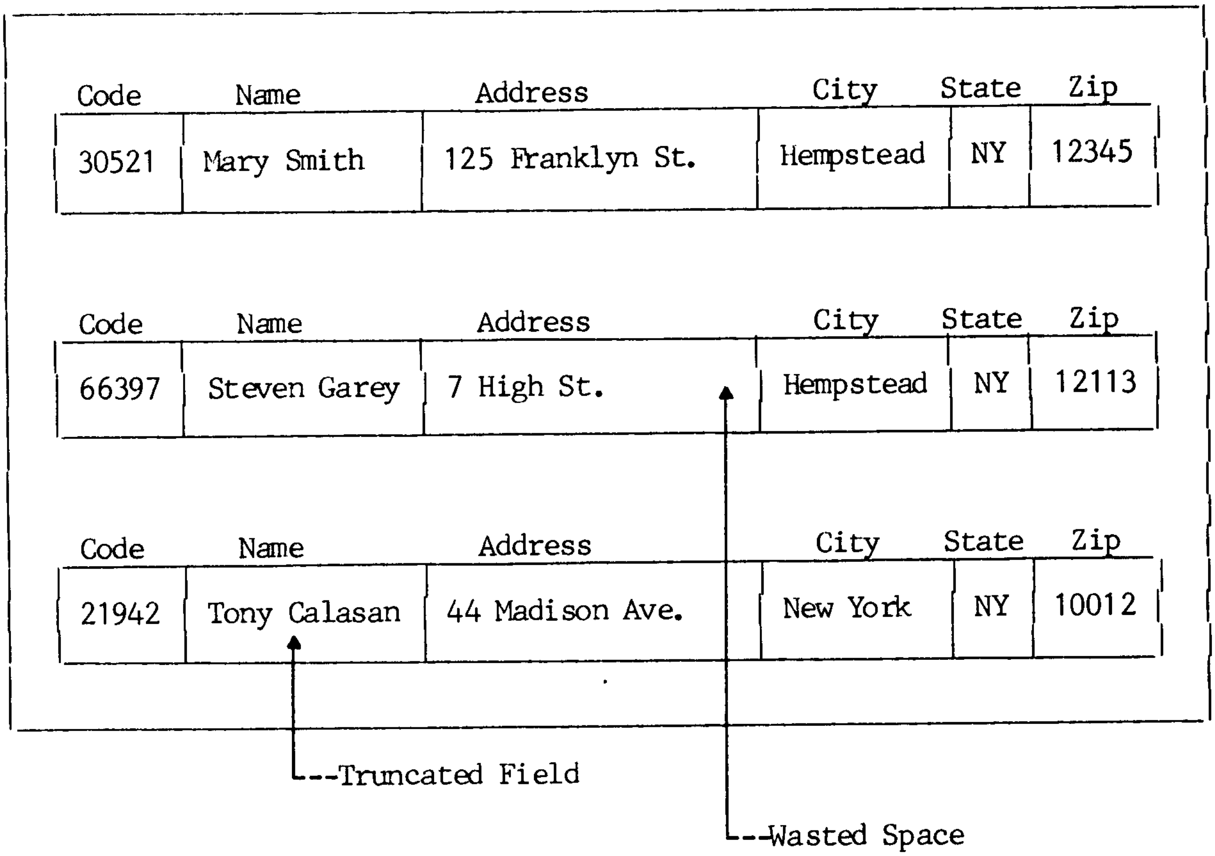
Database management systems on other computers are often only application programs, not part of the operating system. Use of such DBMSs often requires specific data formatting, and data retrieval can be exceedingly complex and slow. On MENTOR, however, the DBMS _is_ the operating system. Therefore, MENTOR offers several advantages in data storage and retrieval over less sophisticated management systems.

MENTOR FILE STRUCTURE

To understand how the MENTOR Operating System is optimized for database management, you must know something about the way information is stored. On MENTOR (as on almost every other computer system), data is stored in "files." A file can be defined as a mechanism for maintaining a set of like records (or items) together logically.

Classically, file data is linked physically, with specific addresses marking where one piece of data starts and another piece ends. In order for the operating system to determine where a specific kind of data can be found within a file, the data is stored in a prearranged pattern. This need for a pattern often makes it necessary to store the data in "fixed fields."

However, just as one size of clothing does not fit all people, one size of data field does not fit all data. Some data will be too short to efficiently use the space set aside in the data field. Other data will be too long to fit entirely in the space that has been set aside. The results: waste caused by fields that are too long and information loss caused by having to truncate long pieces of data.

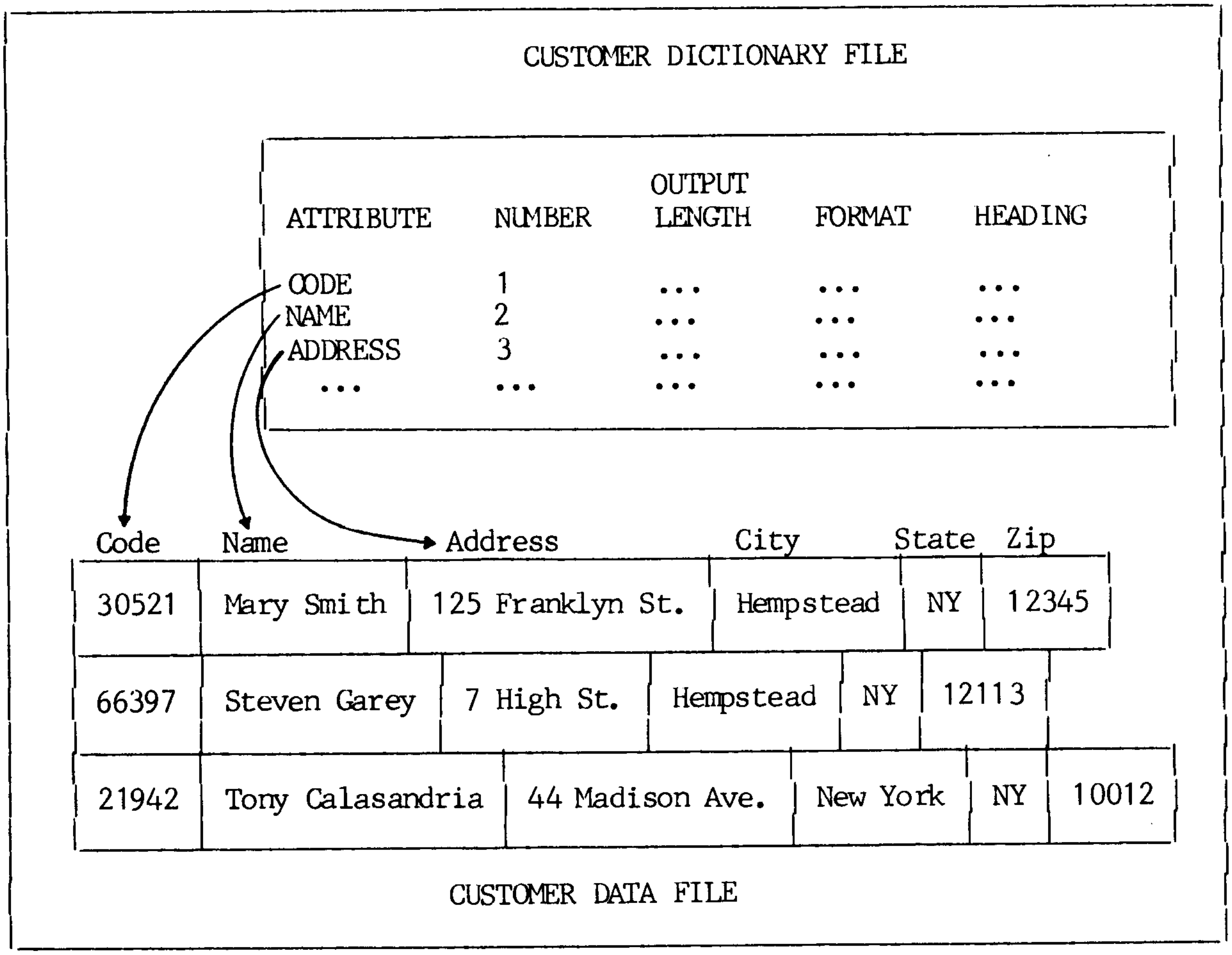


Code	Name	Address	City	State	Zip
30521	Mary Smith	125 Franklyn St.	Hempstead	NY	12345

Code	Name	Address	City	State	Zip
66397	Steven Garey	7 High St.	Hempstead	NY	12113

Code	Name	Address	City	State	Zip
21942	Tony Calasan	44 Madison Ave.	New York	NY	10012

Figure 1-1. Typical Fixed File Structure



CUSTOMER DICTIONARY FILE

ATTRIBUTE	NUMBER	OUTPUT LENGTH	FORMAT	HEADING
CODE	1	...	...	...
NAME	2	...	...	...
ADDRESS	3	...	...	...
...	...	...	...	...

Code	Name	Address	City	State	Zip
30521	Mary Smith	125 Franklyn St.	Hempstead	NY	12345
66397	Steven Garey	7 High St.	Hempstead	NY	12113
21942	Tony Calasandria	44 Madison Ave.	New York	NY	10012

CUSTOMER DATA FILE

Figure 1-2. MENTOR File Structure

In MENTOR, data is accessed via the "dictionary" file that is associated with each data file. This dictionary can be compared to the table of contents in a book. Just as a table of contents defines what is found in each section of a book, a dictionary defines the form your data holds in the data file. Information management is simplified because the dictionary helps provide the user with a two-dimensional "image" of the database.

Data files store your data in a dynamic format. Each record (called an "item" in MENTOR) changes in size as data is added to or deleted from it. Fields (called "attributes" in MENTOR) can also be of variable length. Attributes can be further subdivided with special delimiters, creating any number of variable-length "values," which may be further subdivided into any number of variable-length "subvalues."

Thus, the MENTOR Computer System contains on-line:

- any number of files, which contain:
  - any number of items (records), which contain:
    - multiple attributes (fields), which may contain:
      - multiple values, which may contain:
        - multiple subvalues.

MENTOR places very few restrictions on storage capacity: each item may contain up to 32K bytes and the limit to file size is the capacity of the disk storage on your MENTOR. Such flexibility in storing data is possible because MENTOR uses a Virtual Memory System. Virtual memory systems are usually found only on larger computer systems, but MENTOR includes one to help make disk memory as accessible as main memory.

In a virtual memory system, data is stored in segments called "frames." Each of these frames is numbered with a unique "frame identification number" (or FID). Desired data can be accessed via the FID. If the frame needed is not currently in main memory, it is retrieved from the disk. To make room for the new frame, a frame in main memory that was not accessed recently is rewritten to the disk. See Figure 1-3.

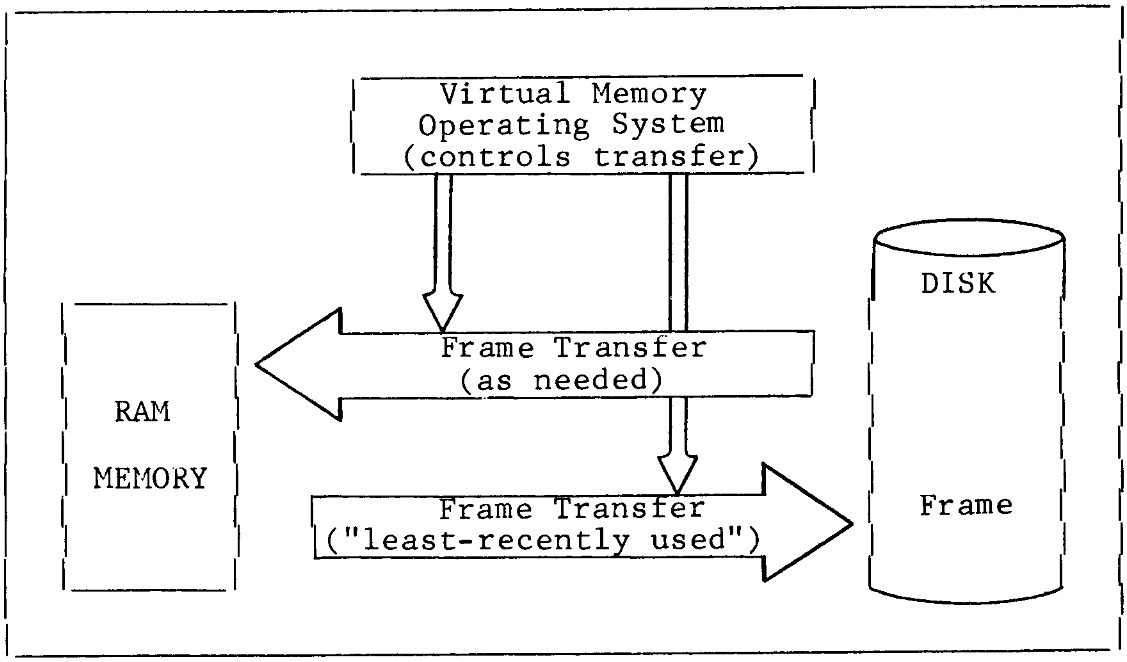


Figure 1-3. Virtual Memory Storage

ADVANTAGES OF THE MENTOR OPERATING SYSTEM

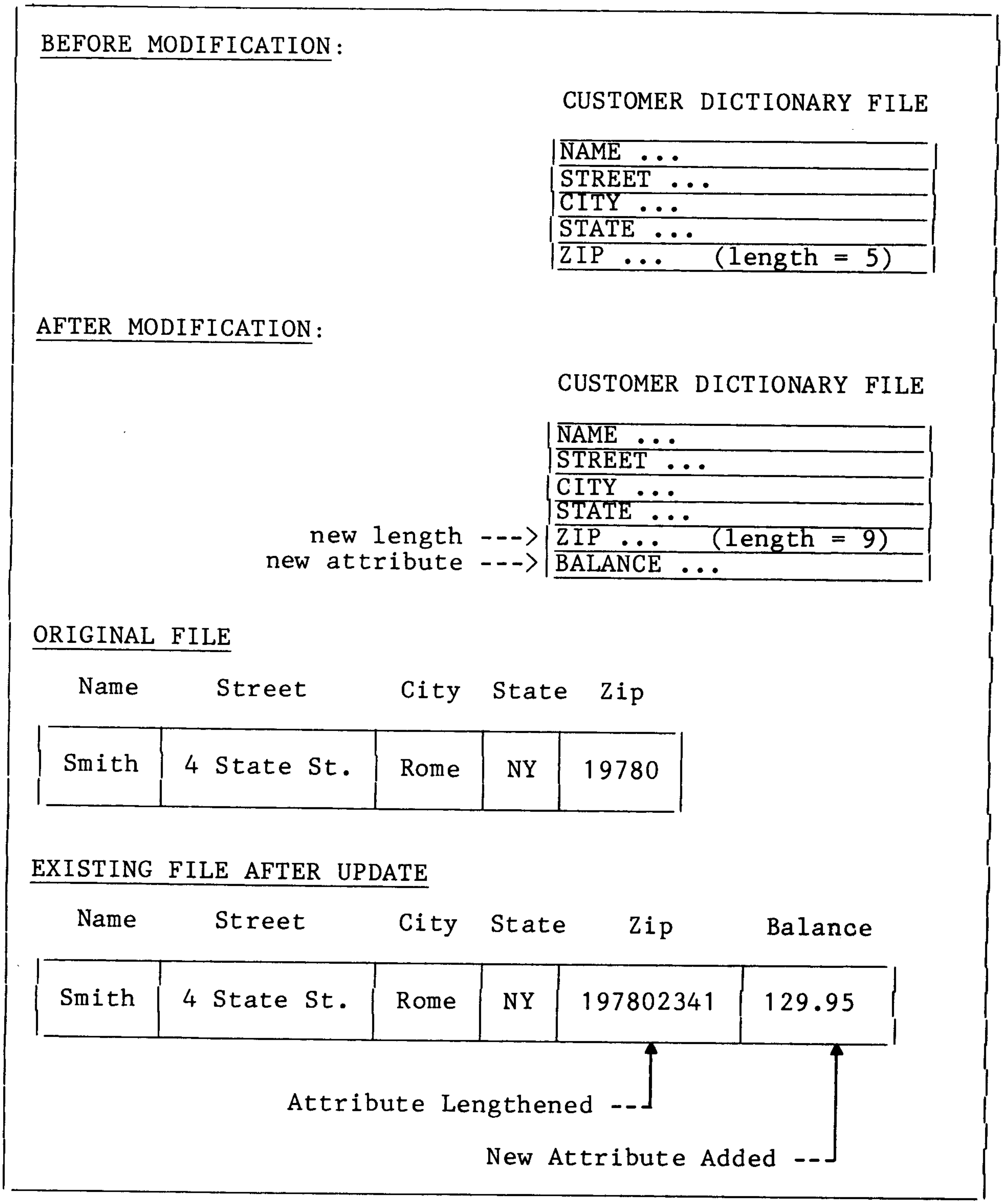
MENTOR's dynamic file structure not only eliminates the problem of data fields that do not fit the data; it eliminates the problem of adding new fields to or changing fields in records. Fixed-field systems require the writing of a program that will extract the data from the database, process it and reconstruct the database. In addition, every program that accesses any part of that database will have to be altered.

In MENTOR, adding to or changing fields in a record is as simple as adding to or changing the dictionary. Since data is not linked physically, only programs that call for the revised data have to be altered.

For example, consider a modification in one type of data: change a ZIP code field in a Customer File from five numbers to the new nine-number code. In computers with fixed data fields, such a change would require a program to extract the data from the database, change the field and generate a new database with a larger ZIP code field. In addition, any program accessing the Customer File would have to be altered to reflect the change in the ZIP codes.

In MENTOR, the same change requires modifying the dictionary entry that describes the ZIP code in the Customer File dictionary (see Figure 1-4). Of course, any program that requires a ZIP code would also have to be altered. However, any other programs that access the Customer File will not be affected.

The dictionary concept offers another benefit: a dictionary can be used to access more than one data file. For instance, you could have a Payroll File for the Smith Company, another Payroll File for the Jones Company and a third Payroll file for Widget Works Inc., and access all of them with one dictionary. This saves space and promotes uniformity because one dictionary does the work of three.



BEFORE MODIFICATION:

CUSTOMER DICTIONARY FILE

NAME ...
STREET ...
CITY ...
STATE ...
ZIP ... (length = 5)

AFTER MODIFICATION:

CUSTOMER DICTIONARY FILE

NAME ...
STREET ...
CITY ...
STATE ...
new length ---> ZIP ... (length = 9)
new attribute ---> BALANCE ...

ORIGINAL FILE

Name	Street	City	State	Zip
Smith	4 State St.	Rome	NY	19780

EXISTING FILE AFTER UPDATE

Name	Street	City	State	Zip	Balance
Smith	4 State St.	Rome	NY	197802341	129.95

Attribute Lengthened --

New Attribute Added --

Figure 1-4. Data Modification in MENTOR

MENTOR FILE HIERARCHY

There are four distinct file levels in MENTOR: the System Dictionary (SYSTEM), User Master Dictionaries (MDs), File Level Dictionaries (DICTs) and Data Files. Files at each level point to multiple files at the next lower level. This hierarchical structure is illustrated in Figure 1-5.

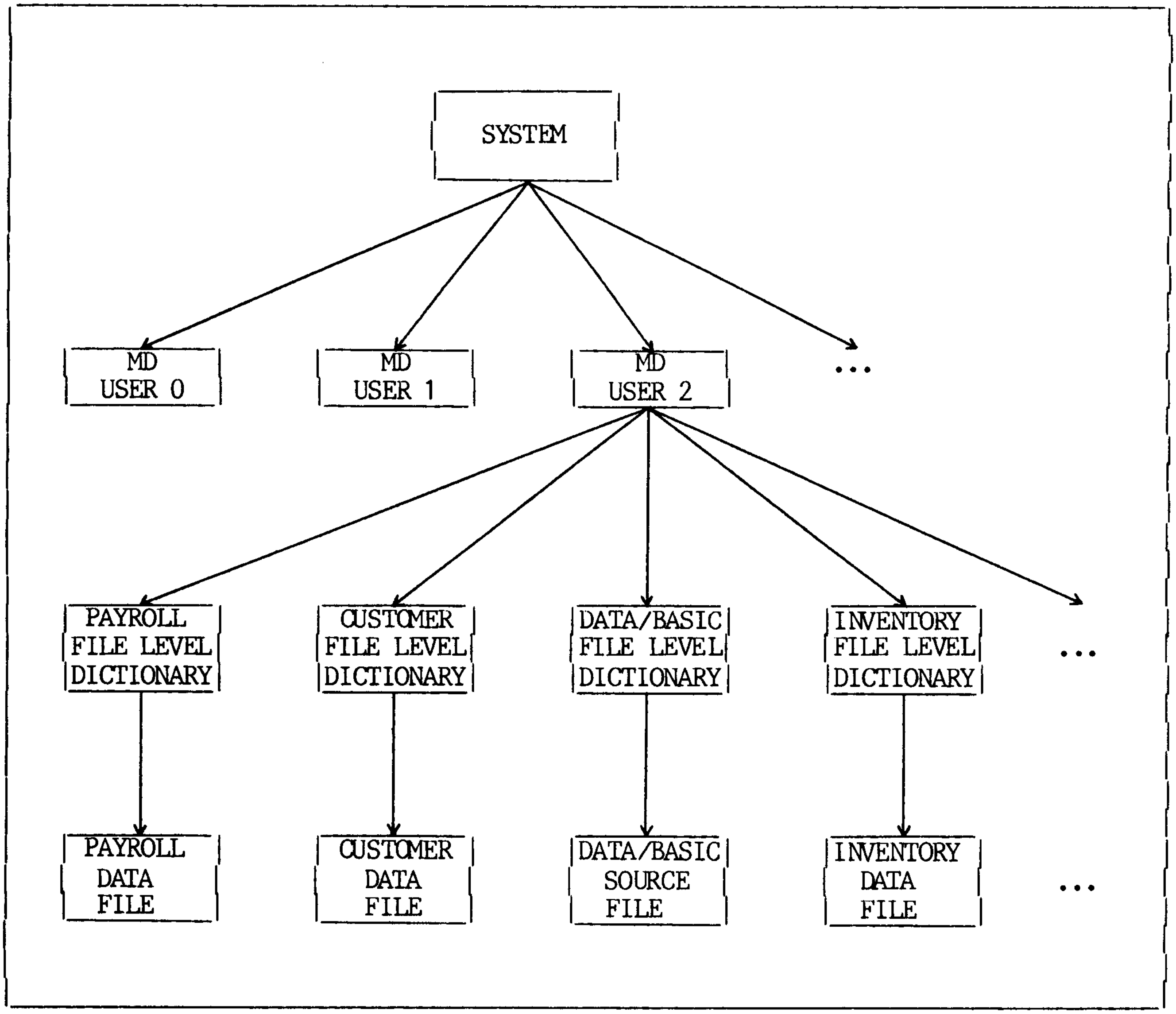

SYSTEM
MD USER 0
MD USER 1
MD USER 2
...
PAYROLL FILE LEVEL DICTIONARY
CUSTOMER FILE LEVEL DICTIONARY
DATA/BASIC FILE LEVEL DICTIONARY
INVENTORY FILE LEVEL DICTIONARY
...
PAYROLL DATA FILE
CUSTOMER DATA FILE
DATA/BASIC SOURCE FILE
INVENTORY DATA FILE
...

Figure 1-5. MENTOR File Hierarchy

System Dictionary

The System Dictionary (SYSTEM) is the highest-level dictionary in MENTOR. It controls the system and contains all user (or account) names and associated passwords, security codes and system privileges. This dictionary also contains pointers to each User Master Dictionary (MD) and to special SYSTEM files.

User Master Dictionaries

A unique User Master Dictionary (MD) exists for each user (or account). Each MD contains that user's vocabulary, PROCs (stored procedures) and pointers to all accessible files. Users can access any file in any account, providing they have the appropriate access privileges.

A standard vocabulary is established whenever an MD is created. One of MENTOR's powerful features is the ability to customize a vocabulary for each user. Synonyms and abbreviated words can increase work speed and cut down on errors. For example, a Customer File could be referred to as CUSTOMER or CUST or even just C in various MENTOR commands.

File Level Dictionaries

Each File Level Dictionary (DICT) describes the structure of the data in each Data File associated with it. DICTs may be shared by more than one Data File.

Some File Level Dictionaries do not have an associated Data File. These "single-level" files contain actual data. They may be referenced either as DICTs or as Data Files.

Data Files

Data Files contain the actual data.

PHYSICAL LAYOUT OF AN ITEM

Greater realization of the power of the MENTOR Computer System comes with an understanding of how data is set up within the file itself. As mentioned previously, data is stored in "items" (similar to records). Within each item, data is stored in "attributes," which can be broken up further into "values" and "subvalues." An illustration of the basic layout of a typical item appears in Figure 1-6.

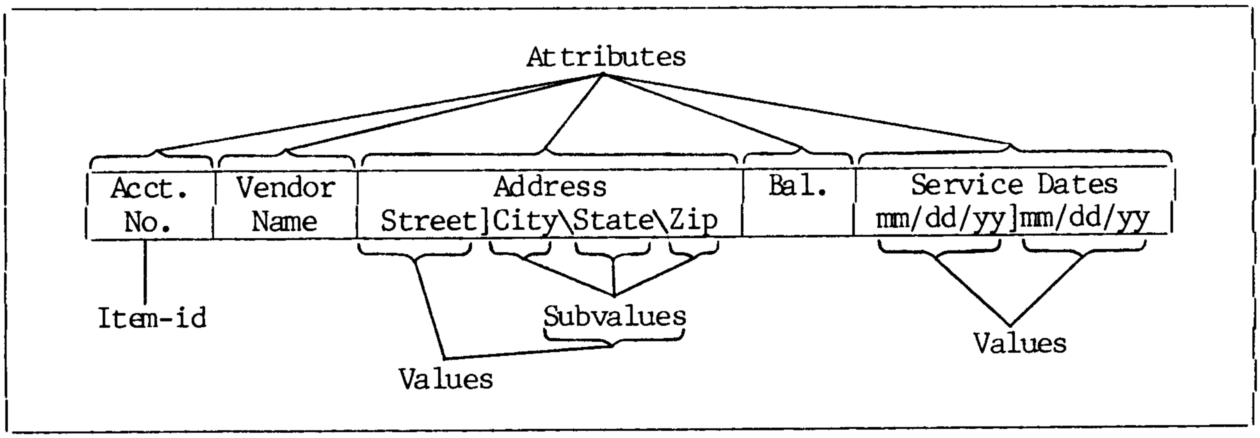


Figure 1-6. Sample Item

Each item is accessed via a unique "item-id" (corresponding to a "key" on other computer systems). This identifier (defined as Attribute #0) is used by the virtual memory system to access data directly. A "count" field associated with the item-id provides a means for relative positioning between items and special characters (^,] and \) separate the data in each item.

The physical structure of an item, with all its values and subvalues and delimiters, is worthy of study. Figure 1-7 illustrates the physical layout of the item shown in Figure 1-6.

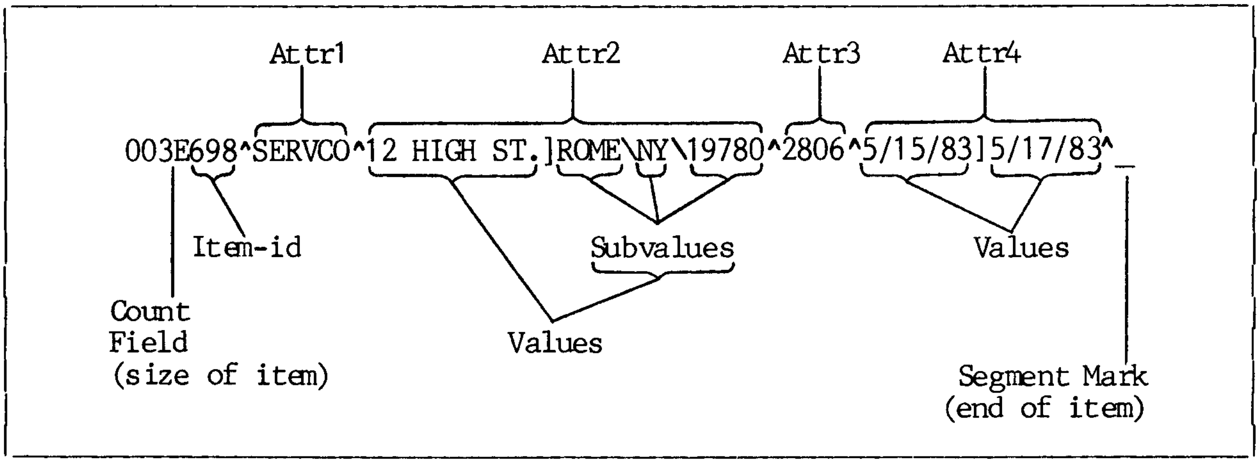

Attr1
Attr2
Attr3
Attr4
003E698^SERVCO^12 HIGH ST.]ROME\NY\19780^2806^5/15/83]5/17/83^_
Item-id
Subvalues
Values
Count Field (size of item)
Values
Segment Mark (end of item)

Figure 1-7. Physical Item Structure

MENTOR SOFTWARE

MENTOR offers some of the most advanced database management software available on any business minicomputer. A brief overview of the software available to users is presented here.

INFO/ACCESS

INFO/ACCESS is a non-procedural data retrieval language with commands which resemble normal speech. It has a relatively free-form syntax that is easy for the non-programmer to use. Yet INFO/ACCESS is sophisticated enough to allow trained users to generate complex reports.

INFO/ACCESS makes data retrieval easy. You can count, list, select and sort data according to criteria you specify. You can generate and save specially-selected lists. You can even generate lists containing a sample of your data or obtain statistics on your data. Information can also be sorted in ascending or descending order or grouped according to any attribute.

INFO/ACCESS also lets you generate reports with headings and footings, totals, etc. INFO/ACCESS is so easy to use that your only limit in its use will be your ingenuity in specifying how information should be retrieved.

ImpleMentor

ImpleMentor is a menu-driven application development tool that generates entire systems. In addition to creating DATA/BASIC application programs, ImpleMentor can document the systems and create needed files. Simple responses to customized prompts generate the programs, so even non-programmers can easily create or update a database. You can use DATA/BASIC to modify the programs and INFO/ACCESS to help format data. ImpleMentor-created application programs can be fully compatible with already-created databases.

DATA/BASIC

DATA/BASIC combines the best features of the classic BASIC computer language with the ability to control the database management power of the MENTOR Computer System. DATA/BASIC features include dynamic arrays, alphanumeric variable names of any length, the capability for structured programming, optional statement labels, advanced string-handling capabilities, complex and multi-line IF statements, eleven digits of arithmetic precision, full INFO/ACCESS conversion and more.

Office AugMentor

The Office AugMentor, incorporating DocuMentor and CalcuMentor, helps you organize today's office electronically. AugMentor features an electronic desk diary, telephone number/address index, calculator and filing cabinet. Messages can be sent via electronic mail, documents can be created or modified with DocuMentor, financial planning can be handled through CalcuMentor and tele-conferences may be held. One option even lets you build INFO/ACCESS sentences through menus and prompts. AugMentor is easy to use: menus and prompts are extensive, and on many options you can have AugMentor search for data.

DocuMentor

DocuMentor makes text entry and document generation simple because it offers features common to advanced word processors. DocuMentor offers quick and easy text entry, formatting and editing. It also provides extensive document merging capabilities, including the ability to merge information selected with INFO/ACCESS. And there are many other features, permitting easy file maintenance and printer control.

CalcuMentor

CalcuMentor maximizes the power of the MENTOR database management system for sophisticated financial planning: model building, results analysis, "what-ifs," etc. A scrollable display window helps you view a spreadsheet containing virtually any number of columns and rows (and pages, permitting three-dimensional spreadsheets). Easy to use functions let you copy and recalculate data and generate reports with just a few keystrokes. Printer options include the generation of bar graphs on the Printronix printer.

Other MENTOR System Software

Modification of any on-line file item can be accomplished through the EDITOR. This system software offers conditional and unconditional string deletion, the ability to merge program text within items and files and the ability to prestore commands.

PROC is a procedural language, comparable to JCL, which can help you create sets of stored procedures, synonyms for file-names, verbs, etc. PROCs can also be used to run programs conditionally and to program interactively.

Database management software lets you generate, manage and manipulate files or portions of files. Utility software adds greatly to MENTOR's power, handling tasks from systems accounting to tape drive control.

RUNOFF is the text processor for the MENTOR Computer System, providing output formatting and text for DATA/BASIC programs, PROCs, etc. Features include the ability to set margins and tabs, justification, page numbering, etc.

SUMMARY

You can easily see that the MENTOR Computer System offers tremendous power in handling data. Part of this power comes from MENTOR's exceptional facility for change. It is easy to create, modify and maintain databases. It is easy to query the database either in a predetermined, report-generating manner or in an ad-hoc fashion.

The MENTOR Computer System was designed for maximum ease of use, even if the user has little or no programming experience. It maximizes your ability to get information from your databases and minimizes the need to spend time on unnecessarily complicated database management schemes.

SECTION 2:

INFO/ACCESS

OVERVIEW

INFO/ACCESS is the user-oriented data retrieval language used on the MENTOR Computer System. It lets you selectively retrieve data using commands that resemble normal speech. This information can be routed to other MENTOR software or to a terminal or printer for output.

INFO/ACCESS is simple, yet sophisticated. Its non-procedural English-type commands make data retrieval simple, even for non-programmers. A typical INFO/ACCESS sentence consists of an easy-to-remember action verb, a "noun" or file-name, data selection-criteria, output specifications, modifiers and options.

You can put INFO/ACCESS to work even if you do not know MENTOR's file structure. Generating "ad hoc" reports and business documents full of up-to-the-minute facts is easier than ever. "What-ifs" are so simple your only limit will be your ingenuity in specifying selection-criteria.

Some of INFO/ACCESS's features include:

- Relatively free-form command order

- Data selection using logical and arithmetic relationships

- Sorting capability on a variable number of ascending and/or descending parameters

- The ability to define the variables used to search, select, sort, total and output and to group on the basis of those variables

- The ability to generate and hold multiple specially-selected or -sorted lists for later use

- Generation of statistical information concerning files

- Automatic or user-specified report generation in either columnar or non-columnar form

- Support of 11-digit signed arithmetic

For more information about INFO/ACCESS, see the MENTOR INFO/ACCESS Reference Manual.

AN INFO/ACCESS PRIMER

In this topic you will see examples of typical office information requirements and how INFO/ACCESS can satisfy those requirements when applied to a typical database.

In Example 1, a list of all customer (account) numbers on file is needed. When the INFO/ACCESS sentence "LIST CUSTOMER" is typed at the TCL prompt (>), a list of item-ids (or keys) will be displayed on the screen, in the same order as they are stored in the Customer File.

```
>LIST CUSTOMER <cr>

PAGE 1                                  12:39:29  23 FEB 1983

CUST......

605
283
543
999
429

5 ITEMS LISTED.
```

Example 1. Simple INFO/ACCESS Sentence

INFO/ACCESS verbs can actually be applied to any MENTOR file. In Example 2, the information needed is a listing of the attribute names in the Customer File Dictionary.

```
>LIST ONLY DICT CUSTOMER <cr>

PAGE 1                                  12:46:18  23 FEB. 1983

CUST......

CODE
NAME
STREET
CITY
STATE
ZIP
BALANCE
DATE
PHONE

9 ITEMS LISTED.
```

Example 2. INFO/ACCESS Used on a Non-Data File

INFO/ACCESS can retrieve data selectively by key. Specific item-ids can be listed in what is called an item-list. The information needed here is the names and phone numbers of customers 429, 605 and 283 (see Example 3). "NAME" and "PHONE" are attributes which will be output for each customer selected.

```
>LIST CUST '429' '605' '283' NAME PHONE <cr>

PAGE 1                              12:43:57  23 FEB 1983

CUST......  NAME.......................... PHONE.......

429         ADDS                             516-231-5400
605         VIRGINIA VIRTUE                  345-6787
283         POLLY POPULAR                    908-765-9934

3 ITEMS LISTED.
```

Example 3. Sample Usage of Item-List

If you needed to find out which customers owe you a great deal of money for a long time, you could use the INFO/ACCESS sentence in Example 4. This sentence selects customers who have had a balance greater than $500.00 for more than six months. The message returned is "2 ITEMS LISTED," indicating that only two items in this file meet both conditions specified.

```
>SELECT CUSTOMER WITH BALANCE > "500" AND WITH DATE BEFORE
 "8/23/82" NAME BALANCE <cr>


2 ITEMS LISTED.
```

Example 4. Sample Usage of Selection-Criteria

Note that the SELECT verb in Example 4 did not list the specified data. This information has been reserved in a "select-list" which can now be saved for later use or used with other MENTOR software or applications.

To save the information selected in Example 4, enter the INFO/ ACCESS sentence in Example 5. The information you selected will be saved in a permanent list called CUSTOMER500.

```
>SAVE-LIST CUSTOMER500 <cr>

LIST SAVED - 1 FRAMES USED.
```

Example 5. Sample Usage of SAVE-LIST Verb

To find out just which customers have owed you more than $500 for more than six months, retrieve the saved list and display it with the following commands. Remember that you can output additional information simply by specifying the appropriate attribute.

```
>GET-LIST CUSTOMER500 <cr>

2 ITEMS SELECTED.

>LIST CUSTOMER NAME <cr>

PAGE 1                           13:12:19  23 FEB 1983

CUST......  NAME...........................

429         ADDS
605         VIRGINIA VIRTUE

2 ITEMS LISTED.
```

Example 6. Retrieving and Displaying Save-List Information

File sorting can be done in ascending or descending order. In Example 7, the information needed is a list of each customer's name and address, in ZIP code order. The Customer File will be sorted in increasing order of ZIP code.

In Example 8, the information needed is a "Customer Balance Report," sorted by name. A heading which includes the report's name and date, and a footing, which contains page numbers, will be printed with the report.

```
>SORT CUSTOMER BY ZIP NAME STREET CITY STATE ZIP <cr>

PAGE 1                                                             12:55:35  23 FEB 1983

CUST..... NAME......................... STREET.................. CITY.............. STATE ZIP..

283       POLLY POPULAR                 321 RIGHTEOUS PATH       ROLLING HILLS      RI    11234
429       ADDS                          100 MARCUS BLVD          HAUPPAUGE          NY    11788
999       TERRY TERRIFIC                56 GOODWILL AVE          GREAT DEEDS        GA    22445
543       FRANK FABULOUS                5 FRIENDLY FIELDS        FREE CITY          FL    33211
605       VIRGINIA VIRTUE               234 SATISFACTION STREET  SUNSHINE           SD    88345

5 ITEMS LISTED.
```

Example 7. Sample Usage of SORT Verb

```
>SORT CUSTOMER BY NAME BALANCE HEADING "CUSTOMER BALANCE REPORT FOR 'DC'" FOOTING "PAGE 'PN'" <cr>

                         CUSTOMER BALANCE REPORT FOR 23 FEB 1982

CUST...... NAME......................... BALANCE...........

429        ADDS                          $1,213.00
543        FRANK FABULOUS                    $4.30
283        POLLY POPULAR                    $45.30
999        TERRY TERRIFIC                  $456.23
605        VIRGINIA VIRTUE                 $678.95



 PAGE  1
```

Example 8. Sample Usage of HEADING and FOOTING Modifiers

INFO/ACCESS can also generate statistical information. In Example 9, the information needed is the total of what customers owe your company.

```
>SUM CUST BALANCE <cr>

TOTAL OF BALANCE = $2,397.78
```

Example 9. Sample Usage of SUM Verb

The STAT command totals a specific attribute, counts the number of items selected and determines the average value of the attribute. If, for example, you wished to determine the average balance owed by each of your customers, you could type in the INFO/ACCESS sentence in Example 10.

```
>STAT CUSTOMER BALANCE <cr>

STATISTICS OF BALANCE:

TOTAL = 2,397.78 AVERAGE = 479.5560 COUNT = 5
```

Example 10. Sample Usage of STAT Verb

INFO/ACCESS VERBS

Some of the major INFO/ACCESS verbs are discussed briefly below. A number of these verbs (e.g., SELECT, FIND) select file items conditionally; others operate on all file items. The items are then made available for use by other MENTOR software.

LIST and SORT

The LIST and SORT verbs generate formatted output. LIST simply lists the selected output, while SORT puts the output into a specified order.

SELECT and SSELECT

The SELECT and SSELECT verbs let users select a set of items. The SSELECT verb sorts these items.

FIND

The FIND verb selects a set of items containing a specified string.

SAMPLE and SSAMPLE

The SAMPLE and SSAMPLE verbs select a set of items matching certain criteria. The SSAMPLE verb sorts these items.

NSELECT

The NSELECT verb selects a set of items that are in a previously-selected list but that are not in another specified file.

COUNT

The COUNT verb totals the number of items meeting the conditions specified. That number is then relayed to the user.

SUM and STAT

The SUM and STAT verbs sum one specified attribute. The STAT verb also provides a count and an average. These derived statistics are then relayed to the user.

T-DUMP, ISTAT and HASH-TEST

The T-DUMP verb allows the user to selectively dump dictionary and/or data files to magnetic tape. The ISTAT verb provides a listing of file-hashing statistics. The HASH-TEST verb is used in determining the optimum size for a file.

SECTION 3:

ImpleMentor

OVERVIEW

ImpleMentor is a menu-driven application generator that enables you to create data entry and data output application programs (Screens and Forms) which are written in DATA/BASIC. With ImpleMentor, you can easily create or update MENTOR databases. Before ImpleMentor, the creation or update of a database required more cumbersome methods (e.g., using database management software and the EDITOR to create the database, developing a DATA/BASIC program to prompt for data entry, etc.).

ImpleMentor can be used by both non-programmers and programmers. Even if you have no programming experience, you can create "Fill-in-the-Blank"-type data entry and data output application programs by responding to ImpleMentor prompts. Programmers may include more complex options, such as extensions of the DATA/BASIC language, in a single step. ImpleMentor greatly increases programmer productivity and minimizes maintenance costs.

With a single modification to an ImpleMentor-created data entry application program, you can easily create a data output application program which produces invoices, checks, receipts, etc. In addition, database information can be printed in a columnar or non-columnar format using INFO/ACCESS commands.

ImpleMentor features include:

- No need for programming experience -- create and maintain applications in hours rather than days -- even if you have never written a program before

- Easy-to-create application programs -- programs are created based on your responses to ImpleMentor prompts -- and ImpleMentor-created programs run as quickly as handwritten programs

- Easy-to-use data entry programs (Screens) -- data is entered simply by responding to customized prompts and can be verified by pattern matching, length testing, range checking and comparison

- Formatted data entry application programs -- data is displayed in a formatted Screen

- Data retrieval by description -- ImpleMentor uses Cross Referencing techniques that retrieve data by its description rather than by codes and numbers

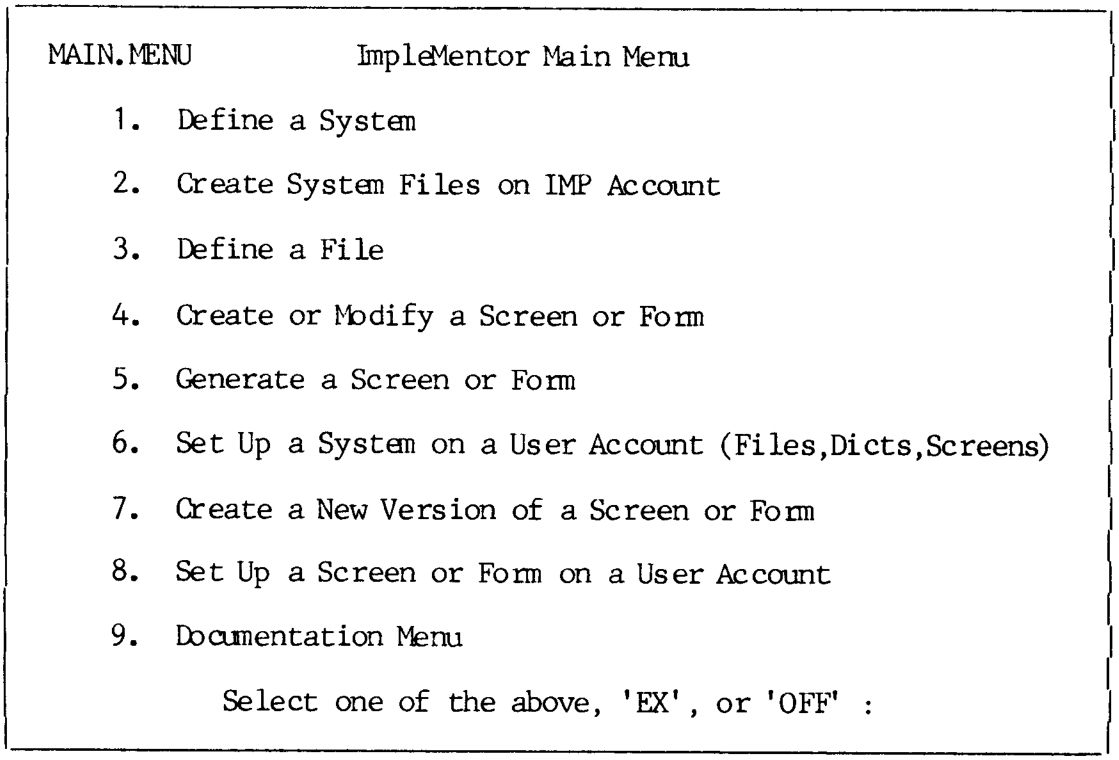
- Easy modification -- application programs can be modified in minutes

- Self-documenting application programs -- each application program is accompanied by an automatically-produced report that describes the details of the application

- The ability to use DATA/BASIC subroutines or INFO/ACCESS commands -- applications can be customized to suit your needs

USING ImpleMentor

ImpleMentor is menu-driven for ease of use. Using selections from the ImpleMentor Main Menu, you may simply create, modify, copy and document application systems. Example 1 illustrates your choices as they appear on the ImpleMentor Main Menu.



```
MAIN.MENU              ImpleMentor Main Menu

      1.  Define a System

      2.  Create System Files on IMP Account

      3.  Define a File

      4.  Create or Modify a Screen or Form

      5.  Generate a Screen or Form

      6.  Set Up a System on a User Account (Files,Dicts,Screens)

      7.  Create a New Version of a Screen or Form

      8.  Set Up a Screen or Form on a User Account

      9.  Documentation Menu

             Select one of the above, 'EX', or 'OFF' :
```

Figure 3-1. ImpleMentor Main Menu

Defining a System

This ImpleMentor Main Menu selection lets you define your system by specifying the system name and description and listing all the files (including sizing information) used by the system.

Creating System Files

ImpleMentor will actually create all the files specified when you defined the system. This Main Menu selection eliminates the tedious task of creating each file individually with the CREATE-FILE command.

Defining a File

Although this selection is optional, defining the layout of each file simplifies the creation of ImpleMentor Screens or Forms and permits setup of dictionary items for use in INFO/ACCESS reports.

Creating or Modifying a Screen or Form

With this Main Menu selection, you can create an application program definition or modify an existing definition by responding to ImpleMentor prompts.

Generating a Screen or Form

ImpleMentor will actually generate the DATA/BASIC code for your application program. This selection files, generates, compiles and catalogs the application program definition.

Setting Up a System on a User Account

After you develop a system, you must copy it to the account where the application will be used. This selection enables you to copy all the system files, dictionaries and necessary verbs to other accounts.

Creating a New Version of a Screen or Form

With this ImpleMentor Main Menu selection, you can copy a Screen or Form before modifying it. This eliminates the need to duplicate much of a definition when more than one application is being created.

Setting Up a Screen or Form on a User Account

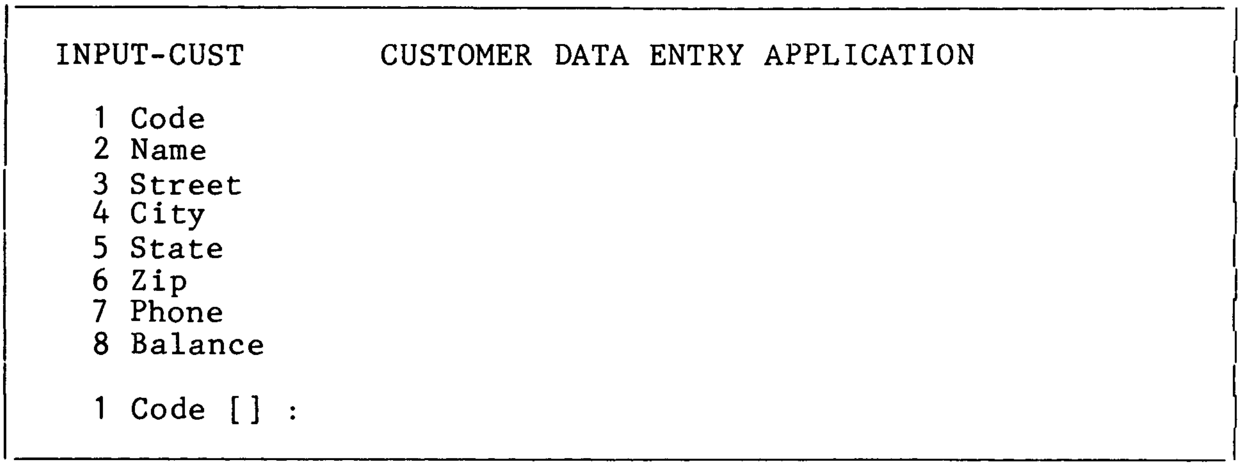
This selection copies individual Screens and/or Forms to other accounts.

Documentation

A Documentation Menu is displayed in response to this selection, permitting simple and thorough documentation of your system (including all your files, data entry and data output application programs).

USING AN ImpleMentor-CREATED APPLICATION PROGRAM

Typing in the command "**INPUT-CUST <cr>**" at TCL executes an ImpleMentor-created data entry application program called INPUT-CUST. This application program, shown in Example 1, will create or update the Customer File.



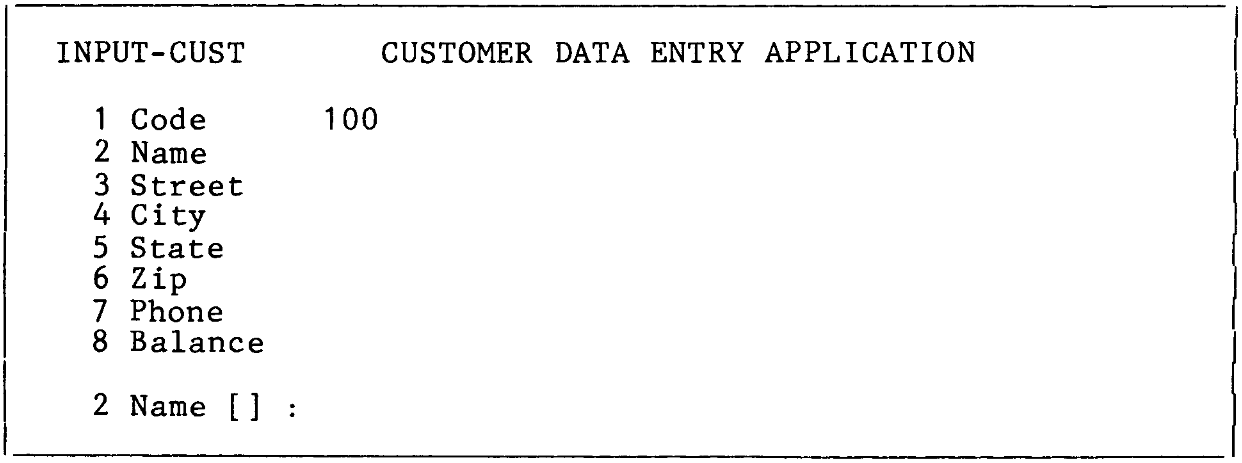
```
INPUT-CUST       CUSTOMER DATA ENTRY APPLICATION

  1 Code
  2 Name
  3 Street
  4 City
  5 State
  6 Zip
  7 Phone
  8 Balance

  1 Code [] :
```

Example 1. Execution of a Sample Data Entry Application

The INPUT-CUST application uses a Formatted Screen with Bottom Line Entry. (Notice the prompt at the bottom of the Screen.) After data has been entered at the first prompt (Code), it is moved to the Code field as shown in Example 2, and the second prompt (Name) is displayed at the bottom of the screen.



```
INPUT-CUST       CUSTOMER DATA ENTRY APPLICATION

  1 Code     100
  2 Name
  3 Street
  4 City
  5 State
  6 Zip
  7 Phone
  8 Balance

  2 Name [] :
```

Example 2. Entering Data Using a Sample Application

Once an item has been stored in a file, it may be retrieved simply by entering the Code at the code prompt; or, if the file has been designed with cross-referencing capabilities, by entering any portion of the data stored in some other field (e.g., all or part of the Name: Applied, Digital Systems, Applied Data).

```
INPUT-CUST          CUSTOMER DATA ENTRY APPLICATION

   1 Code        100
   2 Name        APPLIED DIGITAL DATA SYSTEMS
   3 Street        100 MARCUS BLVD
   4 City          HAUPPAUGE
   5 State         NY
   6 Zip           11788
   7 Phone         516-231-5400
   8 Balance        $12.34

   2 Name [] :
```

Example 3. Retrieval of an Item From the Customer File

The data entry application program (INPUT-CUST) shown in the previous examples, can be copied and modified to become an output application program (Form). This program prepares shipping labels. Any fields that are not necessary for the shipping label have been deleted and the fields have been reformatted as shown in Example 4 below.

```
1 Ship To:
    2       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    3       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    4       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    5       XX
    6       XXXXX
```

Example 4. A Data Output Application Program Definition (Form)

Execution of the data output program or Form is accomplished by typing the name of the Form at the TCL prompt (>). Each item in the Customer File, as specified in the application program definition, will be printed on a separate page. For example, to execute a Form named SHIP-LABELS, type in the following command:

```
>SHIP-LABELS <cr>
```

Example 5 illustrates a shipping label that was printed with the SHIP-LABELS data output application program.

```
    Ship To:
             APPLIED DIGITAL DATA SYSTEMS
             100 MARCUS BLVD
             HAUPPAUGE
             NY
             11788
```

Example 5. A Sample Shipping Label

For more information about ImpleMentor, see the MENTOR ImpleMentor Manual.

SECTION 4:

DATA/BASIC

BASIC (Beginner's All-Purpose Symbolic Instruction Code) is a simple, yet versatile programming language suitable for solving a wide range of problems. BASIC is especially easy for the beginning programmer to master.

DATA/BASIC is an extended version of BASIC, specifically designed for use on MENTOR's database management system. DATA/BASIC's many features include:

- Optional variable-length statement labels (statement numbers)
- Alphanumeric variable names of any length
- Multiple statements on one line
- String handling with unlimited, variable-length strings
- Pattern matching
- Complex and multi-line IF statements
- Dynamic one- and two-dimensional arrays
- Formatting and cursor control
- Floating point arithmetic with up to eleven-digit precision
- INFO/ACCESS data retrieval capabilities
- File access and update capabilities
- External subroutines
- Magnetic tape input and output

For more information about DATA/BASIC, see the MENTOR DATA/BASIC Reference Manual.

STATEMENT	DESCRIPTION
ABORT	Terminates program and returns to TCL.
BREAK	Controls access to DATA/BASIC DEBUGGER.
CALL	Branches to external subroutine.
CASE	Provides conditional selection of a sequence of DATA/BASIC statements.
CHAIN	Passes control to another DATA/BASIC program.
CLEAR	Re-initializes all variables.
CLEARFILE	Clears data section of specified file.
COMMON	Controls variable storage space allocation; may be used to share variables among CHAINed programs.
CRT	Displays data at terminal.
DATA	Stores data in stack for subsequent output requests.
DEBUG	Transfers control to DATA/BASIC DEBUGGER.
DELETE	Deletes specified file item.
DIM	Reserves space for arrays.
ECHO	Turns system echo on and off.
END	Designates the physical end of the program.
ENTER	Transfers control from one cataloged program to another.
EQUATE	Allows variable to be defined as equivalent of another.
FOOTING	Specifies page footing.
FOR	Specifies the beginning of a program loop.
GOSUB	Transfers control to a subroutine.
GOTO	Transfers control to another statement.
HEADING	Specifies page heading.
IF	Provides for conditional execution of specified statements.
INPUT	Requests data from the user.
INPUT @	Requests data, verifies it and moves cursor to a specific position.
INPUTERR	Displays error.
INPUTNULL	Defines a character which can cause a null value to be assigned to a variable on input.
INPUTTRAP	Provides conditional execution of a statement or subroutine.
LOCK	Sets an execution lock.
LOOP	Provides for structured program loops.
MAT	Assigns a value to each element of an array.
MATCHES	Compares string value to a pre-defined pattern.
MATREAD	Reads a file item into an array.
MATREADU	Reads a file item into an array; sets update lock.
MATWRITE	Writes a file item with the contents of an array.
MATWRITEU	Writes a file item with the contents of an array; does not unlock update group.
NEXT	Specifies the ending of a program loop.
NULL	Specifies a non-operation.
ON...GOTO	Computed GOTO.
ON...GOSUB	Computed GOSUB.
OPEN	Selects a file for subsequent I/O.

Table 4.1. Summary of DATA/BASIC Statements

STATEMENT	DESCRIPTION
PAGE	Paginates and prints heading.
PRECISION	Selects number of decimal places for calculations (0-4).
PRINT	Prints specified data.
PRINT @	Prints data at specified position on terminal or after formatting screen.
PRINTER	Controls selection of printer or terminal for program output.
PROCREAD	Reads PROC's primary input buffer.
PROCWRITE	Writes expression to PROC's primary input buffer.
PROMPT	Selects a prompt character for the terminal.
READ	Reads a file item.
READNEXT	Reads next item-id.
READT	Reads next magnetic tape record.
READU	Reads a file item; sets update lock.
READV	Reads an attribute.
READVU	Reads an attribute; sets update lock.
RELEASE	Unlocks file group(s).
REPEAT	Used in conjunction with the LOOP statement.
REM	Specifies a remark (comment) statement.
RETURN	Returns control from a subroutine.
REWIND	Rewinds magnetic tape.
RQM	Terminates program's current time quantum.
SELECT	Builds set of item-ids or attributes for READNEXT.
SLEEP	Suspends processing.
STOP	Designates the logical end of the program.
SUBROUTINE	Specifies a program branch.
UNLOCK	Resets an execution lock.
WEOF	Writes an EOF mark on magnetic tape.
WRITE	Updates a file item.
WRITET	Updates a magnetic tape record.
WRITEU	Updates a file item; does not unlock update group.
WRITEV	Updates an attribute value.
WRITEVU	Updates an attribute value; does not unlock update group.

Table 4.1. Summary of DATA/BASIC Statements (continued)

FUNCTION	DESCRIPTION
@	Controls terminal cursor.
ABS	Returns an absolute value.
ALPHA	Tests for alphabetic value.
ASCII	Converts string from EBCDIC to ASCII.
CHAR	Converts numeric value to ASCII character.
COL1	Returns column position preceding FIELD-selected substring.
COL2	Returns column position following FIELD-selected substring.
COS	Returns cosine of an angle.
COUNT	Counts occurrences of substring within a string.
DATE	Returns current system date.
DCOUNT	Counts values separated by a delimiter.
DELETE	Deletes attribute, value or subvalue from dynamic array.
EBCDIC	Converts string from ASCII to EBCDIC.
EXP	Returns exponential value.
EXTRACT	Returns attribute, value or subvalue from dynamic array.
FIELD	Returns a delimited substring.
ICONV	Provides for INFO/ACCESS input conversion.
INDEX	Returns column position of substring.
INSERT	Inserts attribute, value or subvalue into dynamic array.
INT	Returns the integer value.
LEN	Returns length of string.
LN	Returns natural logarithm.
LOCATE	Returns the index of a substring in a dynamic array.
NOT	Returns logical inverse.
NUM	Tests for numeric value.
OCONV	Provides for INFO/ACCESS output conversion.
PWR	Raises expression to a power.
REM	Returns remainder.
REPLACE	Replaces attribute, value or subvalue in dynamic array.
RND	Generates a random number.
SEQ	Converts first string character to numeric value.
SIN	Returns sine of an angle.
SPACE	Generates string containing blanks.
SQRT	Computes square root.
STR	Generates specified string.
SYSTEM	Interrogates tape errors and obtains printer status information.
TAN	Returns tangent of an angle.
TIME	Returns system time of day.
TIMEDATE	Returns system time and date.
TRIM	Removes extraneous blanks from a string.

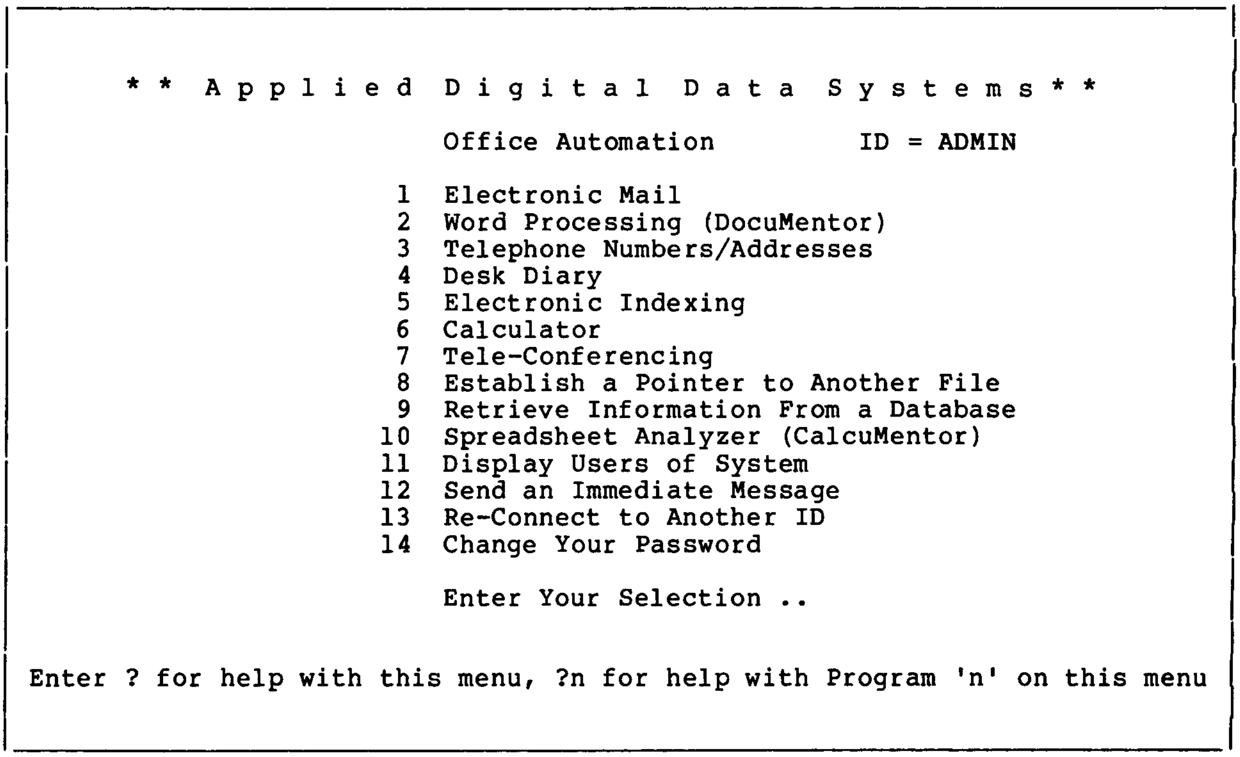
Table 4.2. Summary of DATA/BASIC Intrinsic Functions

SECTION 5:

Office AugMentor

OVERVIEW

The Office AugMentor helps you organize office activity, supplementing or replacing many procedures currently done by hand. The result: increased office productivity and decreased paper flow. Even new users will find AugMentor productive, because the system is easy to use -- it is entirely menu-driven, each option features extensive prompting and helpful explanations of each menu option are available when needed.



```
     * * Applied Digital Data Systems * *

                    Office Automation        ID = ADMIN

                 1  Electronic Mail
                 2  Word Processing (DocuMentor)
                 3  Telephone Numbers/Addresses
                 4  Desk Diary
                 5  Electronic Indexing
                 6  Calculator
                 7  Tele-Conferencing
                 8  Establish a Pointer to Another File
                 9  Retrieve Information From a Database
                10  Spreadsheet Analyzer (CalcuMentor)
                11  Display Users of System
                12  Send an Immediate Message
                13  Re-Connect to Another ID
                14  Change Your Password

                    Enter Your Selection ..


Enter ? for help with this menu, ?n for help with Program 'n' on this menu
```

Figure 5-1. Office AugMentor Main Menu

AugMentor FEATURES

AugMentor helps you spend less time keeping track of data: menu options let you refresh your memory by listing available categories of data and built-in menu management features let experienced users go right from one Main Menu to another. AugMentor also includes the following features:

Electronic Mail

You can send and receive mail electronically. Send newly-entered messages or previously-created documents, or send an immediate message to any logged-on user. Direct your mail to one person or group or to everyone on the system. AugMentor keeps track of all messages. If mail is waiting for you, you will be notified automatically -- even if the mail arrived while you were away from your desk.

Word Processing (DocuMentor)

AugMentor provides several time-saving features that simplify use of the DocuMentor Word Processing System. User-created document templates help standardize the layouts of common office documents like memos, status reports, etc. Menu options simplify the editing and printing of RUNOFF files. And AugMentor can help you find a specific document even if all you remember is a line or phrase of the text.

Electronic Telephone Number/Address Index

Replace your current telephone/address book and make names, addresses and telephone numbers easier to find by storing them electronically. Entries can be displayed on your terminal or printed out, and listed individually, as part of a telephone directory or as address labels. AugMentor's ability to search electronically means that if you can recall part of an entry, you will be able to find the name, phone number and address.

Desk Diary

AugMentor can replace your engagement calendar and make personal time management a more efficient proposition. Note appointments, meetings and luncheons in the desk diary and AugMentor will remind you of these events automatically. Check your daily calendar or consult a monthly calendar that highlights important days in the next several weeks. AugMentor can even schedule meetings automatically by determining a time each person is free.

Electronic Indexing

AugMentor's electronic indexing can help you locate information in your office. Group material in "file folders" and "filing cabinets" just as you would group paper files. Identify the location of specific data by checking the index yourself or have AugMentor search for it electronically.

Calculator

AugMentor can replace your desktop calculator, letting you perform calculations including square roots and exponents. Generate subtotals and grand totals and use any of a hundred storage registers to hold results or to perform register arithmetic.

Tele-Conferencing

AugMentor lets people meet without having to leave their desks. User-defined networks limit the persons who can participate in the interactive dialogue. Participants can even "whisper" to each other during the meeting. In addition, complete minutes are kept on each tele-conference until they are deleted by the leaders of the meetings.

Retrieving Information From a Database

AugMentor makes INFO/ACCESS even easier to use by simplifying the procedure used to assemble sentences. Menus display available verbs and sort and selection-criteria. Plain-English prompts let you add output specifications. You can use AugMentor-created sentences just as you would any other INFO/ACCESS sentence, in report generation or other data retrieval functions.

Spreadsheet Analyzer (CalcuMentor)

CalcuMentor provides a spreadsheet that may be used for financial planning on MENTOR. It also gives you the ability to do complex forecasting, budgeting, model-building and decision-making. Since MENTOR power is integrated with CalcuMentor, spreadsheets of almost unlimited size are possible. And data from other MENTOR files can be included in spreadsheets automatically.

Management Options

Other AugMentor options include the ability to list people currently using the system, to log to another ID in the AugMentor account or to create or change a password on your account. An Administrator account is also included to permit the setup of other users and to maintain the entire system.

SUMMARY

AugMentor saves time and effort -- instead of spending time and energy *maintaining* typical office data, people can now spend time and effort on *using* that data more efficiently.

For more information about Office AugMentor, see the MENTOR Office AugMentor Reference Manual.

SECTION 6:

DocuMentor

OVERVIEW

DocuMentor expands the MENTOR Computer System by adding word processing and a full-screen editor to MENTOR's extensive database management capabilities.

DocuMentor makes it easy to create, format and edit perfect reports, forms, letters, memos and more. Standard documents may be individualized by merging them with other MENTOR files. You can even use INFO/ACCESS to specify the information you want to use. Numerous prompts guide you to the next step and to help menus. DocuMentor is simple enough to use comfortably and versatile enough to use flexibly.

DocuMentor OPERATION

DocuMentor operates on the ADDS Viewpoint® and some Regent® terminals. Their typewriter-like keyboards simplify text entry. The screens dedicate up to 24 lines to text. The last line is reserved for prompts and status indicators.

Ruler Mode

Ruler Mode is one of DocuMentor's three distinct operating modes. In this mode, users create electronic "rulers" that control margins and tabs. Features include typewriter-style tabs, automatic-indent tabs, right-justification tabs, number-alignment tabs and centering tabs. A "hot" zone is also present to allow word-wrapping and hyphenation. In addition, up to nine user-created rulers may be saved for later use in an editing session.

Edit Mode

DocuMentor features several commands that control cursor movement and text insertion and deletion. Cursor movement may take place between characters, words, lines, sentences, paragraphs and pages. The insertion and deletion of characters, words, sentences and lines can be accomplished with a single touch. You may also move the cursor to the top or bottom of the screen or document or to the beginning or end of a line.

Other editing features include global search and replace, spelling checking, changing letter case, document assembly and character transposition. Both Ruler Mode and Insert Mode are accessible while editing.

Regent and Viewpoint are registered trademarks of Applied Digital Data Systems Inc.

Insert Mode

DocuMentor's third operating mode permits the automatic and semi-automatic insertion of data. In this mode, Backslash commands, Tilde commands and Special Text Insertion commands may be inserted into a document.

Backslash commands help format documents. These commands help you set line spacing, right-justify text, create headers and footers and tables of contents, break pages, chain documents and generate prompts and comment lines.

Tilde commands permit the introduction of changeable data into your text and control some printing features. The commands permit underlining, boldface printing and listing the document's name, the current page number and the current date and time.

Reserved Tilde commands, called Special Text Insertion commands, control DocuMentor's merging capability.

Merging

Merging is perhaps DocuMentor's most outstanding feature. The Special Text Insertion commands in DocuMentor permit merging from the keyboard or from a database.

For example, you may want to send one memo to your company's hourly staff and another memo to salaried personnel. You can do it easily with DocuMentor. Or you could tell your best customers about a new product by sending personalized sales letters to just those clients. DocuMentor and INFO/ACCESS can be combined to make it easy to select information from your database and to put the right information into the right documents.

DocuMentor SCREEN EDITOR

The DocuMentor Screen Editor extends full-screen editing capability to any on-line item. RUNOFF files, PROCs, DATA/BASIC programs and other items may be displayed on a full screen and modified with DocuMentor cursor positioning and editing functions.

SUMMARY

Documentor makes text creation and editing quick and easy. And DocuMentor's ability to merge with any part of your database gives you unprecedented flexibility in personalizing letters and forms and any of your other correspondence. Let DocuMentor show you what an integrated word processing package can do for your business.

For more information about DocuMentor, see the MENTOR DocuMentor Reference Manual.

SECTION 7:

CalcuMentor

CalcuMentor provides sophisticated financial modeling capabilities on your MENTOR Computer System. CalcuMentor speeds up and simplifies decision-making, model-building and "what-if" analysis. This spreadsheet can help you generate more realistic forecasts and budgets. It can also make cost and cashflow projections more accurate. Pro forma financial statements are easier than ever to create.

But CalcuMentor does more than provide a spreadsheet for your MENTOR System. CalcuMentor helps optimize the power of the MENTOR database management system for business problems. It brings a new dimension to the amount and variety of data that can be used in your spreadsheets.

It gives you great computing power: spreadsheets may have any number of columns and rows -- and the ability to create separate pages lets you use three-dimensional spreadsheets. Multiple spreadsheets can be consolidated according to a user-determined relationship. Data and formulas are easy to move from one portion of the spreadsheet to another. Finished spreadsheets can be printed on a printer attached to the system or to your terminal and bar graphs can be printed on a Printronix printer.

Other CalcuMentor features include:

- Help screens which provide operational details for all functions; the screens can be viewed at any time

- A scrollable display window that lets you create and view extra-wide spreadsheets

- The ability to center or left- or right-justify data in columns of variable width and print with or without column and row headings

- Decimal precision of up to four places

- Single-keystroke execution of formulas and calculations

- Recalculation of data by row or column; this can be done automatically or manually

- Sorting of spreadsheet data by multiple keys, in ascending or descending order

CalcuMentor is also available as part of Office AugMentor. For more information about Calcumentor, see the MENTOR CalcuMentor Reference Manual.

SECTION 8:

OTHER MENTOR SYSTEM SOFTWARE

EDITOR

The EDITOR permits the modification of any on-line file item. These items can include DATA/BASIC programs, PROCs, data items and dictionary items. The EDITOR uses the "current-line" concept -- most commands permit only one line to be altered or deleted at a time. However, certain commands permit specific operations on groups of lines.

The EDITOR includes the following features:

- Two variable-length temporary buffers
- Absolute and relative current-line positioning
- Line-number prompting on input
- The ability to merge lines in the same or other items
- Character-string location and replacement
- Conditional and unconditional line deletion
- The ability to prestore commands
- Input/output formatting, using RUNOFF commands

Sample EDITOR commands appear in Table 8.1. For more information about the EDITOR, see the MENTOR EDITOR/RUNOFF Reference Manual.

COMMAND	DESCRIPTION
A	Executes last Locate (L) command again.
AS	Toggles Assembler Format on/off.
B	Moves current-line pointer to bottom of item.
C	Lists column numbers.
DE	Deletes lines from current item.
EX	Exits the EDITOR, cancelling any changes made.
F	Merges updates into previously existing item.
FD	Deletes item from file.
FI	Files current item and permits editing of next item in list.
FS	Files current item and continues editing session.
G	Moves current-line pointer to a specified line.
I	Allows insertion of new lines or a new character string.
L	Lists a specified number of lines or locates a specified string.
M	Macro expansion.
ME	Merges lines from another item.
N	Increments the current-line pointer.
P	Creates or prestores an EDITOR command.
PD	Displays prestored EDITOR commands.
R	Replaces a number of lines.
RU	Replaces a number of character strings.
S	Suppresses printing of line numbers.
S?	Displays the size of the item being edited.
T	Moves current-line pointer to top of item.
TB	Sets tabs.
U	Moves current-line pointer up.
X	Cancels last I, ME, DE or R command.
XF	Cancels last Update (F) command.
Z	Sets print column limits.
<cr>	Lists lines one at a time in ascending sequence.
?	Denotes the position of the current-line pointer.
^	Toggles the special effect of the ^ character within a parameter string.

Table 8.1. Summary of EDITOR Commands

PROC

PROC is the procedural language used on MENTOR. It is comparable to JCL (Job Control Language) in its method of operation, but PROC is more powerful: it has conditional capabilities and can interact with users.

PROC is designed to help the experienced programmer create customized applications. It can help the user become more efficient by permitting a sequence of TCL commands to be executed automatically or by creating verb synonyms and alternate file-names. PROCs can also be used to test and verify data entered at the keyboard. PROCs are easily invoked: their one-word names are typed in as TCL statements.

Other PROC features include:

- The ability to branch to other PROCs and subroutines and to pass along arguments

- Pattern matching

- Relational character testing

- Free-field and fixed-field character movement

- Optional command labels

PROCs can speed up and simplify often-repeated, lengthy tasks. For example, instead of entering an INFO/ACCESS sentence repeatedly to gain a complete listing of information, you could create a PROC that duplicates the functions of that sentence. A typical INFO/ACCESS sentence is shown below (it is assumed that the file and attributes listed are present):

```
>LIST INVENTORY WITH DESCRIPTION "COIL" INV DESCRIPTION COST <cr>
```

This sentence will list each "coil" in the Inventory File and will also output three specified attributes for each item listed. To list other parts in the file, INFO/ACCESS sentences would have to be entered repeatedly. Alternately, a PROC could be created so you could use the same INFO/ACCESS sentence, having to change only the name of the part to be listed. Example 1 illustrates such a PROC.

```
001 PQ
002 HLIST INVENTORY WITH DESCRIPTION
003 O THE DESCRIPTION YOU WANT IS +
004 IP
005 A"
006 H INV DESCRIPTION COST
007 P
EOI 007
```

Example 1. A Sample PROC

The lines in the PROC (which could be called INV) have the following functions:

001 Marks this entry as a PROC
002 First part of INFO/ACCESS sentence: lists user-specified items in the Inventory File
003 Displays a prompt: "THE DESCRIPTION YOU WANT IS"; permits entry of the desired attribute name
004 Stores the first part of the sentence in a buffer
005 Surrounds the entered data in double quotation marks
006 Ends INFO/ACCESS sentence by including output of Inv, Description and Cost attributes
007 Processes completed INFO/ACCESS sentence
EOI 007

Since this PROC and the INFO/ACCESS sentence perform the same function, the PROC name could be typed in instead of the entire INFO/ACCESS sentence. This cuts down on the time spent entering commands and reduces the likelihood of keyboard errors that could occur if the INFO/ACCESS sentence were entered several times. The PROC may be invoked as often as you like; each time it will stop and permit you to enter the name of the attribute you wish to list. For more information about PROC, see the MENTOR PROC Reference Manual.

DATABASE MANAGEMENT SOFTWARE

The database management software lets you generate, manage and manipulate files or portions of files. See Example 2 for illustrations of the use of this system software.

Create-File generates new dictionary and/or data files by creating file entries in your Master Dictionary and by reserving disk space.

Clear-File clears the data and/or dictionary sections of a file and maintains the space that remains. Any "overflow" file space your file may have used is returned to an available file space pool.

Delete-File deletes the data and/or dictionary sections of a file. The storage space your file used is returned to an available file space pool.

Copy copies selected items or an entire file to a designated peripheral (terminal, line printer, etc.) or to another file in your Master Dictionary.

For more information about database management software, see the MENTOR System Reference Manual.

>CREATE-FILE CUSTOMER 1 11 <cr>	Reserves disk space for a Customer File with a dictionary modulo of 1 and a data modulo of 11.
>CLEAR-FILE DATA STOCK <cr>	Clears the data section of the Stock File.
>DELETE-FILE TEST <cr>	Deletes the dictionary and data sections of the Test File.
>COPY DICT SAMPLE * <cr> TO:(DICT TEST <cr>	Copies all the items in the dictionary of the Sample File to the dictionary of the Test File.
>COPY SAMPLE N1 N2 N3 <cr> TO:<cr>	Copies items N1, N2 and N3 of the Sample File to the user's terminal.

Example 2. Sample Usage of Database Management Software

UTILITY SOFTWARE

Utility software is called upon to handle many functions on the MENTOR Computer System. This software performs math functions, determines file statistics, performs system accounting tasks, aids communication between computer users, sets terminal and printer format characteristics and helps perform cold-starts and power-up diagnostics.

One of the most versatile portions of the utility software is the Spooler. It holds all jobs that are to be printed (jobs also may be stored on tape). Each user may send work to the printer without conflicting with other users. When the desired printer is available, the Spooler will send out any jobs that are waiting.

Spooler features include:

- Retention and control of 600 files in up to 125 output queues and the ability to request up to 125 copies of an item as a single entry

- Control over one parallel and up to 19 serial printers

- The ability to redirect output to a specific printer (e.g., to send word processing documents to a letter-quality printer)

- The ability to transfer print files to and from tape (independent of Spooler activity)

A few illustrations of utility software usage are shown in Example 3. For more information about utility software, see the MENTOR System Reference Manual.

>T-ATT <cr>	Attaches tape drive to terminal issuing command.
>ITEM TEST N1 <cr>	Displays frame identification number (FID) of item N1 of the Test File.
>SP-STATUS <cr>	Displays current Spooler status.
>ACCOUNT-SAVE <cr>	Saves the contents of an account on magnetic tape.

Example 3. Sample Usage of Utility Software

RUNOFF

RUNOFF combines with the EDITOR to provide basic text-processing capabilities for MENTOR. Formatting options include adjustable margins and tab stops, justification, highlighting of text and chapter and section numbering and formatting.

RUNOFF also offers these features:

- Overriding upper-/lower-case mode and commands for boldface text
- Headings and footings containing the current system time and/or date, page number, source-file name and literals
- Creation of tables of contents and index lists
- The ability to interact with other MENTOR software to chain and merge files and accept information from the keyboard and from INFO/ACCESS select-lists

Example 4 illustrates a sample RUNOFF source-file in its stored form and as it would be printed. For more information about RUNOFF, see the MENTOR EDITOR/RUNOFF Reference Manual.

RUNOFF Source-File DOC DEMO

```
001 .LEFT MARGIN 15 .LINE LENGTH 42
002 .JUSTIFY
003 This paragraph is right-justified.  Notice that the
004 right margin forms a straight line.
005 .SKIP 1 .NOJUSTIFY
006 The Nojustify command is used here.  Now the text does
007 not form a straight line against the right margin.
```

RUNOFF Command

```
>RUNOFF DOC DEMO <cr>
```

RUNOFF Output

```
This paragraph is right-justified.  Notice
that  the right  margin  forms  a straight
line.

The Nojustify command is used here.  Now
the text does not form a straight line
against the right margin.
```

Example 4. Sample RUNOFF Source-File and Output

SECTION 9:

TERMINAL CONTROL LANGUAGE

The Terminal Control Language (TCL) is the primary interface between the computer user and the MENTOR system software.

TCL prompts the user with a ">" symbol. Most software can be called directly from TCL with a single input statement. Some software returns to TCL automatically after completion of processing. Other software returns to TCL only after being exited by the user. This differentiation combines speedy processing with the flexibility of working on separate control levels.

A special feature of TCL is the TCL Stacker, which is integrated with the EDITOR. The Stacker makes your work easier by letting you store, review, modify and execute TCL statements. A few easy-to-remember keystrokes are all you need to manipulate an entire stack of TCL statements.

You can refresh your memory by displaying the last statement ... or the last 20. You can issue Stacker commands that will execute several TCL statements consecutively. You can even modify a previously-used statement to generate another statement.

The Stacker enables users to master the mechanics of TCL quickly. Wasted time, caused by typing errors and repeated entry of long command strings, is cut down greatly. And your ability to execute several TCL statements automatically cuts down on time spent at the terminal. The MENTOR TCL Stacker can greatly improve your productivity.

COMMAND	DESCRIPTION
.E4	Edits line 4.
.L3,1	Lists 3 stored TCL statements starting with line 1.
.X1,3,4	Executes lines 1, 3 and 4.

Table 9.1. Some Typical Stacker Commands

VERB	DESCRIPTION
ADDD (ADDX)	Adds decimal (hexadecimal) numbers.
BASIC	Compiles a DATA/BASIC program.
BUFFERS	Identifies all frames currently in RAM.
CATALOG	Catalogs a DATA/BASIC program.
DUMP	Displays data in virtual memory in ASCII or in hexadecimal.
ISTAT	Generates and lists a file-hashing histogram for a file.
ITEM	Provides usage statistics on file items.
MULD (MULX)	Multiplies decimal (hexadecimal) numbers.
POKE	Inserts a TCL command/string into another port's input buffer for execution.
POVF	Displays content of system overflow table.
RTD (DTR)	Converts radix to decimal (and vice versa).
RUN	Executes a DATA/BASIC program.
SUBD (SUBX)	Subtracts decimal (hexadecimal) numbers.
TIME	Prints system time and date.
WHAT	Displays current system parameters.
WHO	Prints the port number and account name for the terminal being used.

Table 9.2. Some Typical TCL Verbs

COMMENT SHEET

LET US KNOW . . . if you have any recommendations for improving this publication. After writing your comments, fold, tape and mail this pre-addressed form to us. All comments will be given careful consideration and become the property of Applied Digital Data Systems Inc.

PUBLICATION TITLE ________________________________ DATE __________

PUBLICATION NO. ________________

YOUR NAME ________________________________ TITLE __________

COMPANY ____

ADDRESS ____

____

Did you find errors in this manual?
If so, please specify by page number.

____

____

____

Did you find this manual well-organized, usable, and easy to understand? If not, please make suggestions for improvement.

____

____

____

Is the information in this manual complete? if not, please indicate what is missing and where it should be placed.

____

____

____

Please indicate the type of user/reader that you most nearly represent.

__ Applications programmer
__ Occasional programmer
__ Consultant
__ Management personnel
__ Operator
__ Sales
__ Prospective user

ADDS
Applied Digital Data Systems Inc.
A Subsidiary of NCR Corporation