REALITY™ by Microdata

What is REALITY?

REALITY™ is a true, generalized data base management computer system. It is a complete system that provides multiple users with the capability to instantly update and/or retrieve information stored in the on-line data files. Users communicate with the system thru local or remote terminals to access files that may be private, common or security-controlled. Each terminal user's vocabulary can be individually tailored to specific application jargon.

REALITY includes the powerful, yet simple to use ENGLISH™ inquiry language, DATA/BASIC., PROC high-level language, file mainenance tools, text editor, complete programming development facilities and a host of other user amenities. And it all runs in an on-line, multi-user environment with all system resources and data files being efficiently managed by a microprogrammed Virtual Memory Operating System.

REALITY is built on field proven Microdata computers and peripherals, utilizing microprograms to provide users with unrivaled performance and reliability in the medium-sized computer market.

Why is REALITY Different?

REALITY is uniquely different when measured from any angle — system capability, multi-user performance, file management languages, ease of programming, data structure, and architectural features. The high performance and fast response of REALITY are possible only thru the use of the high speed microprocessor which greatly reduces system overhead and program execution time.

The unique microprogrammed firmware contains the:
- Virtual Memory Manager
- Multi-user Operating System
- Special Data Management Instructions
- Input/Output Processors

The unique System Software includes:
- Languages — ENGLISH, DATA/BASIC, PROC, TCL
- Selectable/automatic report formatting
- Dynamic file-memory management
- Selectable levels of file/data security

The unique file structure provides:
- Variable length files/records/fields
- Multi-values (and subvalues) in a field
- Efficient storage utilization
- Fast accessibility to data item
- Selectable degrees of data security
- File size limited only by size of disc
- Record size up to 32K bytes

Features
- True data base management
- Microprogrammed Virtual Memory Operating System
- Up to 32 users
- On-line file update/retrieval
- Speaks ENGLISH
- Variable file/record/field lengths
- Dynamic file/memory management
- Automatic report formatting
- Speaks DATA/BASIC
- Total data/system security
- Fast terminal response
- Line printer spooling
- Special data management instructions
- High-speed generalized sort
- Small computer price
- Big computer performance

What is ENGLISH?

ENGLISH™ is a generalized information management and data retrieval language. A typical inquiry consists of a free-form sentence containing appropriate verbs, nouns, connectives, and data selection criteria. Each user's vocabulary can be individually tailored to his application jargon.

Verbs are action-oriented and include: LIST, SORT, COUNT, and SELECT. ***Nouns*** consist of file names, data attribute names, and record/field names. ***Connectives*** are used to combine grammatical phrases, alter the report format, and modify the action of the verb. ***Data selection criteria*** can specify a particular item or record, an entire file, or a conditional retrieval criteria combining relational and logical operators with data values.

ENGLISH is a dictionary-driven language to the extent that the vocabulary used in composing an ENGLISH sentence is contained in several dictionaries. Verbs, file names, and connectives are located in each user's master dictionary (M/DICT). Every user file consists of a data file and a dictionary file which contains a structural definition of the data. ENGLISH references the dictionary for data attribute descriptions. These descriptions specify data fields, functional calculations, interfile retrieval operations, and display format and positioning.

Merely using key words in an ENGLISH sentence may invoke many file accesses and functional operations before the requested display is automatically formatted on the terminal or printer. But most importantly, ENGLISH is easy to understand and use. It is not a programmer's language — it is a data management language for people who need fast access to information.

File Structure

The REALITY files are organized in a hierarchial structure with files at one level pointing to multiple files at a lower level. Four distinct file levels exist: System Dictionary, User Master Dictionary, User-File Dictionary, and User-File Data.

The SYSTEM DICTIONARY file contains all legal user log-on names, passwords, and security codes. Each entry points to a corresponding user master dictionary.

Each USER MASTER DICTIONARY (M/DICT) file contains all user vocabulary (verbs, nouns, connectives, and throwaways), all accessible file names, application PROCs (procedural programs), and attributes describing the structure of the information in a dictionary. The file name pointers can reference any file or dictionary in the system.

A USER-FILE DICTIONARY file contains attributes (and attribute synonyms) describing the structure of the data in the user-file. These attributes define the data field names, describe how it is to be accessed and displayed, and any functions or interrelations with other files or data records. Attributes are also used to explicitly access a data field by name or to implicitly reference a specified group of fields. Each dictionary also contains a pointer to the corresponding file data.

The USER-FILE DATA file contains the actual data stored in a variable field length format. In addition to the normal record/field data structure, a field can contain multiple values and a value can consist of multiple sub-values.

An ENGLISH Primer

An English statement contains several grammatical structures which can be represented as follows:

Verb	File (Noun)	Attributes (Nouns)	Selection Criteria	Miscellaneous Connectives
LIST SORT COUNT etc.	name DICT name	implicit explicit	all 'item-id' WITH . . .	LPTR DBL-SPC SUPP SORT-BY etc.
Required		Optional		

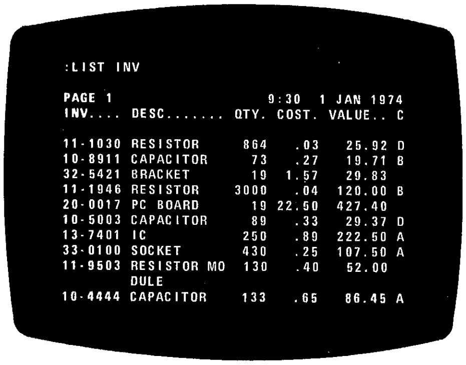
The **verb** must be the first word, while the other words in the statement can generally be in any order. The **file** specification permits accessing either the data or the dictionary of a file.

The **attribute** list may be explicitly stated using attribute names found in the file dictionary. If none are specified in the statement, the implicit attribute synonym list (1, 2, 3, . . .) in the file dictionary will be used to specify the displayed fields.

The **selection criteria** determine which items in the file will be operated upon. If nothing is specified, then all items will be used. One or more direct references may be made by specifying the item-id in single quotes. A conditional retrieval may be specified by a WITH clause. All items in the file will be interrogated, but only those meeting the specified criteria will be accepted. The WITH clause may be a simple or complex combination of attribute names, relational operators (=, >, LT, AFTER, etc.), logical operators (AND, OR), and data values ("100," "12/21/72," "RESISTOR," etc.).

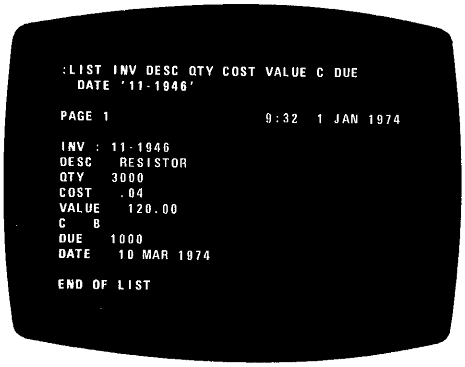
The **miscellaneous connectives** may be used to modify the effect of the verb or alter the display format.

The following examples demonstrate some of the capabilities of ENGLISH:



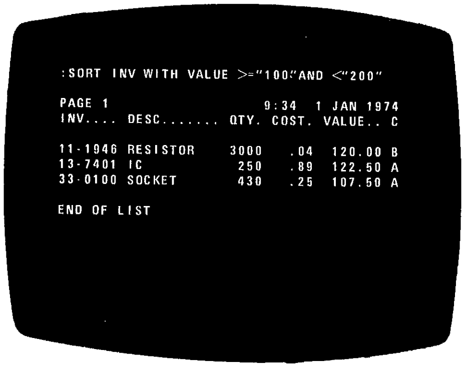
```
:LIST INV

PAGE 1                       9:30  1 JAN 1974
INV.... DESC....... QTY. COST. VALUE.. C

11-1030 RESISTOR     864   .03   25.92 D
10-8911 CAPACITOR     73   .27   19.71 B
32-5421 BRACKET       19  1.57   29.83
11-1946 RESISTOR    3000   .04  120.00 B
20-0017 PC BOARD      19 22.50  427.40
10-5003 CAPACITOR     89   .33   29.37 D
13-7401 IC           250   .89  222.50 A
33-0100 SOCKET       430   .25  107.50 A
11-9503 RESISTOR MO  130   .40   52.00
        DULE
10-4444 CAPACITOR    133   .65   86.45 A
```

When the system prompts the display with a colon (:) the user may enter an ENGLISH statement. LIST INV will list all items in the file INV. The column headings are determined from the implicit attribute synonym list in the file dictionary. Data that is longer than the allocated print field will be folded onto succeeding lines. The item-id appears in the column headed by the file name INV.



```
:LIST INV DESC QTY COST VALUE C DUE
  DATE '11-1946'

PAGE 1                       9:32  1 JAN 1974

INV : 11-1946
DESC   RESISTOR
QTY   3000
COST   .04
VALUE   120.00
C   B
DUE   1000
DATE   10 MAR 1974

END OF LIST
```

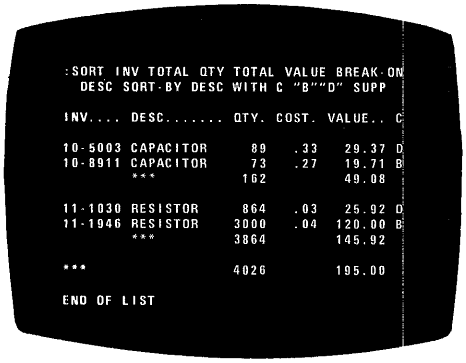
This LIST statement is explicitly naming the attributes to be displayed (DESC . . . DATE) from the file INV. A particular item selection ('11-1946') has also been specified. If the attributes requested for display cannot be fit across the page, ENGLISH will automatically revert to non-columnar display as seen here.



```
:SORT INV WITH VALUE >="100"AND <"200"

PAGE 1                       9:34  1 JAN 1974
INV.... DESC....... QTY. COST. VALUE.. C

11-1946 RESISTOR    3000   .04  120.00 B
13-7401 IC           250   .89  122.50 A
33-0100 SOCKET       430   .25  107.50 A

END OF LIST
```

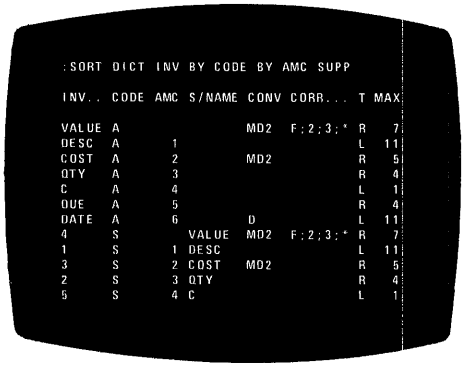
This SORT statement is using a conditional selection criteria. (WITH . . . "200"). All items in the file INV will be interrogated and only those meeting the selection criteria will be displayed in sorted order. When no sort keys are specified, the item-id will be used as the sort key.

This SORT statement is using the BREAK-ON/TOTAL feature to summarize (TOTAL) specified attributes. The totals will be displayed every time the value of the BREAK-ON attribute (DESC in this case) changes. A sort key (SORT-BY DESC) has been specified. The selection criteria can be read as 'WITH *the value of* C ***equal to*** "B" ***or*** "D" '. The connective SUPP is used to suppress the page title line (PAGE . . . 1974) that would normally be displayed. The last line of the report is a grand total.



```
:SORT INV TOTAL QTY TOTAL VALUE BREAK-ON
  DESC SORT-BY DESC WITH C "B""D" SUPP

INV.... DESC....... QTY. COST. VALUE.. C

10-5003 CAPACITOR     89   .33   29.37 D
10-8911 CAPACITOR     73   .27   19.71 B
        ***          162         49.08

11-1030 RESISTOR     864   .03   25.92 D
11-1946 RESISTOR    3000   .04  120.00 B
        ***         3864        145.92

***                 4026        195.00

END OF LIST
```

SORT DICT INV displays a scrted listing of the dictionary for file INV. Two sort keys are specified (BY CODE ***and*** BY AMC). The dictionary contains attributes (CODE = A) and attribute synonyms (CODE = S). The AMC specifies the field position of that attribute in the data record. The synonyms "1, 2, 3, 4, 5" form the implicit attribute synonym list. CONV is the conversion specification (D for date, MD2 for masked decimal with 2 fractional digits). CORR is the correlatives specification (F;2;3;* is the function of multiplying field 2 times field 3). The type (T) specifies display of data either left (L) or right (R) justified in MAX columns.



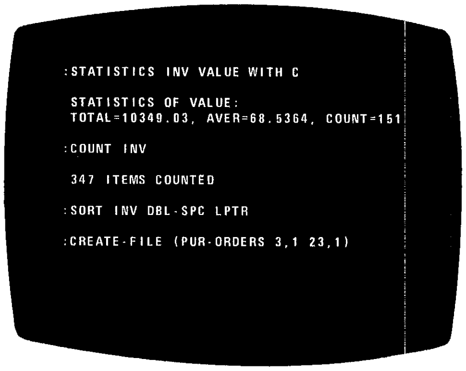
```
:SORT DICT INV BY CODE BY AMC SUPP

INV.. CODE AMC S/NAME CONV CORR... T MAX

VALUE A               MD2  F;2;3;* R   7
DESC  A      1                     L  11
COST  A      2        MD2          R   5
QTY   A      3                     R   4
C     A      4                     L   1
DUE   A      5                     R   4
DATE  A      6        D            L  11
4     S        VALUE  MD2  F;2;3;* R   7
1     S      1 DESC                L  11
3     S      2 COST   MD2          R   5
2     S      3 QTY                 R   4
5     S      4 C                   L   1
```

STATISTICS is used to sum a specified attribute (VALUE). The display shows the accumulated total, the count of the number of items that met the selection criteria (WITH ***any non-null value for*** C), and the average value.

COUNT is used to determine the number of items in a file which meet the selection crite'ia.

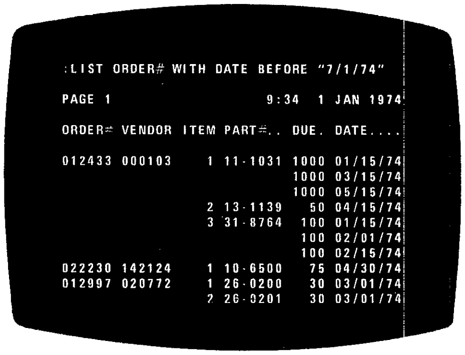
On the SORT INV statement, double spacing (DBL-SPC) will be used between items and the sorted listing will be directed to the line printer (LPTR).

CREATE-FILE is one of the many system verbs. It directs the memory manager to allocate a dictionary and a data file under the name PUR-ORDERS. The parameters (3,1 23,1) are used to optimize the organization of the file structure.



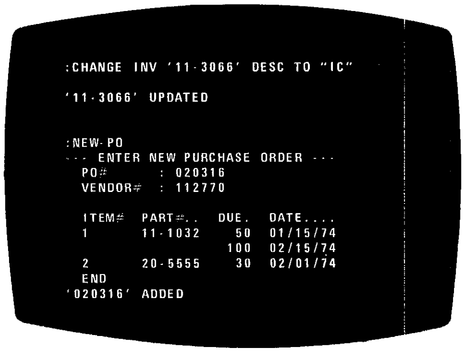
```
:STATISTICS INV VALUE WITH C

 STATISTICS OF VALUE:
 TOTAL=10349.03, AVER=68.5364, COUNT=151

:COUNT INV

 347 ITEMS COUNTED

:SORT INV DBL-SPC LPTR

:CREATE-FILE (PUR-ORDERS 3,1 23,1)
```

Attributes ITEM and PART # on this listing may contain multiple values per field which are displayed in a columnar format. In addition, the attributes DUE and DATE show a further level of indenturing by having multiple subvalues for a given value. ENGLISH will automatically display values and subvalues in an indentured format while maintaining their corresponding relationships.



```
:LIST ORDER# WITH DATE BEFORE "7/1/74"

PAGE 1                   9:34  1 JAN 1974

ORDER# VENDOR ITEM PART#.. DUE. DATE....

012433 000103    1 11-1031 1000 01/15/74
                           1000 03/15/74
                           1000 05/15/74
                 2 13-1139   50 04/15/74
                 3 31-8764  100 01/15/74
                            100 02/01/74
                            100 02/15/74
022230 142124    1 10-6500   75 04/30/74
012997 020772    1 26-0200   30 03/01/74
                 2 26-0201   30 03/01/74
```

The PROC high-level procedural programming language can be used to write ENGLISH-like verbs or to create an interactive data entry routine. The CHANGE proc accepts all required information in a single statement. The NEW-PO proc prompts the user for required information and sets up column headings which can then be filled in by the user.



```
:CHANGE INV '11-3066' DESC TO "IC"

'11-3066' UPDATED


:NEW-PO
--- ENTER NEW PURCHASE ORDER ---
  PO#       : 020316
  VENDOR#   : 112770

  ITEM#  PART#..  DUE.  DATE....
  1      11-1032    50  01/15/74
                   100  02/15/74
  2      20-5555    30  02/01/74
  END
'020316' ADDED
```

REALITY System Configurations

Basic System

- Central Processor Unit
- 16K bytes core
- 4 terminal channels
- 5 million byte disc
- 9 track, 800 BPI, 25 ips mag tape
- 165 cps, 132 column printer with desk
- Attractive upright cabinet
- One CRT display terminal with desk
- All software and firmware

REALITY Options

- Up to 32 terminals
- Up to 40 million bytes of disc
- Up to 64K core memory
- 300 line/minute printer
- Telephone line modems

Other Microdata Products

Proprietary OEM minicomputers

Microdata is the world's leader in microprogrammable minicomputers. Today thousands of Microdata minicomputers are successfully operating in a wide variety of applications. These computers offer real performance advantages and the protection of your proprietary software.

OEM-designed peripherals

Microdata can give you the competitive edge with our line of OEM disc drives and magnetic tape transports. They're IBM compatible and their price is low. And because we manufacture them ourselves (and use them in our own systems) they have the quality, performance and delivery you need.

Complete systems ready for market

If you want someone to put it all together for you, then Microdata's total system manufacturing and test capabilities are just what you need. We've got a great track record in building reliable, headache-proof, cost effective systems for such diverse applications as data communications, traffic control, telephone monitoring, phototypesetting, business systems, and computer science classrooms.

Microdata Sales Support

International Representatives

Canada D.E. McMullen, Toronto 416/362-1157
England CMC Ltd., London 0442.61266
France Intertechnique, Plaisir 460.33.00
Germany CMC Frankfurt/Main 06 11/69 13 66
South Africa Spectro Data Division (PTY) Ltd., Randburg 48-1473
Venezuela Henrique Pfeffer, C.A., Caracas 34.58.51

REALITY Dealers

California Data Products, San Diego, CA 714/560-0777
Century 21 Data Systems, Farmington, MI 313/478-4400
Computer Management Services, Inc., Portland, OR 503/227-1457
Datatel, Alexandria, VA 703/549-4300
Electronic Systems, Bellevue, WA 206/641-4990
Electronic Systems of Colorado, Englewood, CO 303/773-1510
Insurnational Inc., Dallas, TX 214/358-2468
Keystone Data Systems, Cherry Hill, NJ 609/779-1901
Minicomputer Sales & Leasing, West Orange, NJ 201/736-9116
Ohio Data Products, Cleveland, OH 216/621-1888
PPI Programs Inc., New York, NY 212/697-4466
Southeast Data Services, Inc., Juneau, AK 907/586-2524
Southern California Data Products, Irvine, CA 714/833-9132 or 213/437-6098
Systems Management, Inc., Des Plaines, IL 312/298-3840
The Computer Works, Oakland, CA 415/547-6565
Tidelands Data Products, Inc., Houston, TX 713/440-6111

Microdata Corporation
17481 Red Hill Avenue Irvine, California 92714
Telephone 714/540-6730 - TWX 910-595-1764