MAY/JUNE 1983
Volume 11 Number 3

I.P. Sharp newsletter

MAGICSTORE

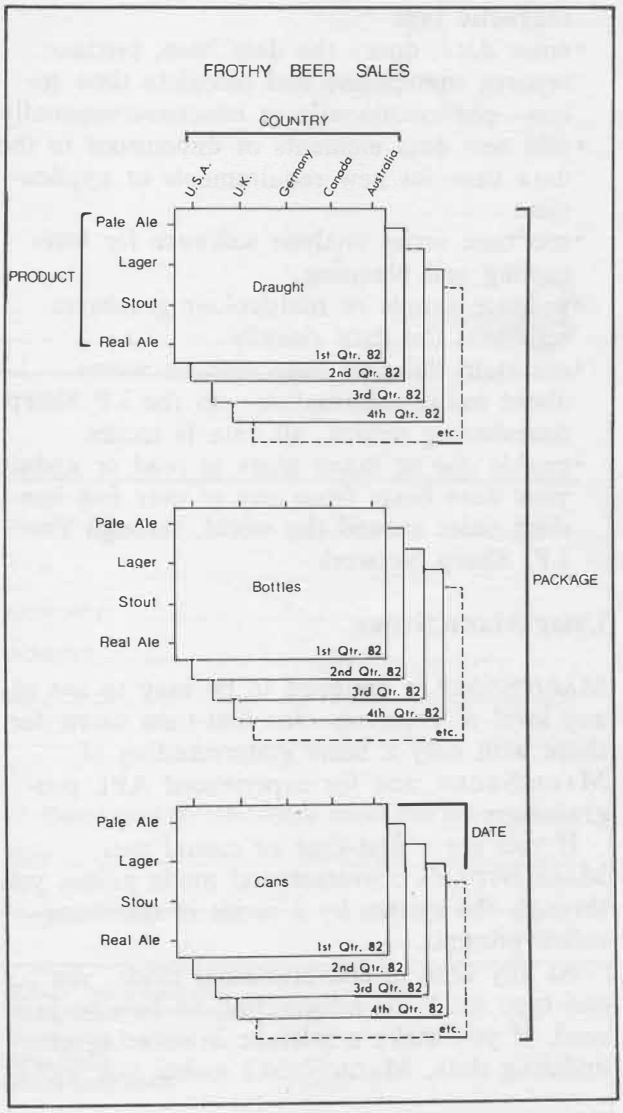
The data that we work with daily is rarely "flat"—that is, two-dimensional. It tends to have many dimensions, and to look like this data (which represents a beer company's worldwide sales).

With multidimensional data like this, analysis and reporting can cut across any dimension arbitrarily. For example, the management of the Frothy Beer Company is likely to want:

- reports showing sales in Germany over the last five years
- a graph of worldwide sales of draught beer over the last ten years, exposing the underlying trend
- a graph showing the proportion of total sales represented by pale ale, lager, stout, and real ale, respectively

If your information needs are similar to this, you will benefit from a software product recently released by I.P. Sharp: MAGICSTORE.

MAGICSTORE is available in public workspace 39 *MAGICSTORE*, and through the *LINK* function in all public MAGIC workspaces. MAGICSTORE is the successor to I.P. Sharp's successful WIZARD II product, which will still be available.



in this issue

... MAGICSTORE

MAGICSTORE is designed for the creation, maintenance, analysis, and reporting of multidimensional time series data bases—such as the Frothy Beer Company's sales information. Because MAGICSTORE is an interactive, easy-to-use product—from creating your own data base, through data entry, to analysing the information—you'll be able to respond effectively to management requirements and tight deadlines.

Using MAGICSTORE for your data base management needs, you can:

- create a data base of time series (for example, monthly sales and expenses by branch)
- enter or update data through a terminal; or extract data from existing files, or from magnetic tape
- enter data, query the data base, produce reports, manipulate and calculate time series—conversationally or nonconversationally
- add new data elements or dimensions to the data base for new requirements or applications
- use time series analysis software for forecasting and planning
- produce simple or multicolour graphs to represent the data visually
- maintain the data base with no worry about losing information—on the I.P Sharp timesharing service, all data is secure
- enable one or many users to read or update your data bases from any of over five hundred cities around the world, through the I.P. Sharp Network

Using MAGICSTORE

MAGICSTORE is designed to be easy to use at any level of expertise—for first-time users, for those with only a basic understanding of MAGICSTORE, and for experienced APL programmers.

If you are a first-time or casual user, MAGICSTORE's conversational mode guides you through the system by a series of questions—called prompts.

At any time in conversational mode, you can type *HELP* for information on how to proceed. If you make a mistake in entering or updating data, MAGICSTORE assists you by telling you what you did wrong, and allowing you to make changes and corrections as you go.

Once you've become familiar with MAGICSTORE, or if you're already an experienced APL programmer, you can use it in nonconversational mode. In this mode, you can extract and store data under program control. Using nonconversationl commands, you can easily apply MAGICSTORE to modelling, forecasting, consolidations, batch updating, or sophisticated data validation.

Combining data and software

MAGICSTORE is designed to be easily combined with other time series–oriented software.

You can assemble your own data base software, building on MAGICSTORE. And you can use MAGICSTORE with other I.P. Sharp products: MAGIC, for compression, combination with other data bases, and calculation; EASY, for forecasting and modelling; and SUPERPLOT, for business graphics.

You may need to refer to two or more data bases to produce special reports—for example, an analysis for market planning and forecasting. With MAGICSTORE, you can combine the information from your private data bases with data from another MAGICSTORE data base, or with data from I.P. Sharp's public data bases.

Applications

MAGICSTORE can be used for almost any application involving time series data—from simple storage and retrieval of a few facts, to more complex systems, such as financial consolidations or economic forecasts. (For example, I.P. Sharp makes extensive use of MAGICSTORE in maintaining a hundred of its own public data bases.)

SHARP APL customers around the globe are using MAGICSTORE in a variety of applications. Some of these include:

- personnel and employment record data bases
- price modelling systems
- financial consolidations of market and sales returns
- profitability and cashflow reporting
- income statements and balance sheets

. . . MAGICSTORE

- currency exchange rates data bases
- economic forecasts
- export products data bases
- company operating plan models
- sales forecasting and planning

For more information . . .

Just contact one of I.P. Sharp's offices around the world, and ask for the *MAGICSTORE Users' Guide*—you'll find all the information you need to use MAGICSTORE. Your I.P. Sharp representative can also arrange for demonstrations and user training.

If your interest is in purchasing MAGICSTORE for use on your in-house SHARP APL timesharing system, ask your I.P. Sharp representative for licence fees and a contract.

For information on how to convert your WIZARD II data bases to MAGICSTORE files, contact your I.P. Sharp representative or the MAILBOX group *MQ*. ■

Public Library Update

New	Old	Comments
1 *INFO*	1 *PUBLICLIBS*	Replaces 1 *PUBLICLIBS*, and contains up-to-date information on the contents of libraries 1 to 699.
3 *QUICKDRAW*	800 *PLOT*	New version, manual available. See article in this newsletter.
3 *SAGA*		New version, see *NEWS* in 3 *SAGA*, items 5–18, and article in this newsletter.
39 *MAGIC*	39 *MAGIC*	Now interfaces with IBM PC using MICROCOMM diskette. See MICROMAGIC manual, and *I.P. Sharp Newsletter*, March/April 1983.
39 *MAGICSTORE*	588 *WIZARD*	Name change, new version 3. See article in this newsletter.
48 *FASTNET*	548 *FASTNET*	Improved version; see *NEWS* in 48 *FASTNET*, and *I.P. Sharp Newsletter*, March/April 1983.
57 *CAPERS*		New version 3
499 *CAPERS*57	57 *CAPERS*	Version 2
499 *SAGA*∆*V*1	3 *SAGA*	Old version moved to graveyard; see *NEWS* in 3 *SAGA*, item 18 for conversion aids. ■

Clement Kent, Toronto

QUICKDRAW

If you are familiar with I.P. Sharp's family of graphics products, you may think QUICKDRAW is a new addition. However, QUICKDRAW has been available since 1979, but until recently it has been known as PLOT7221. Because QUICKDRAW is designed to work specifically with Hewlett-Packard 7221 series plotters, it takes advantage of all firmware-resident features of these plotters, and the result is a very efficient graphics product.

If you have high-volume, production graphics applications, and a good basic knowledge of SHARP APL and function editing, QUICKDRAW provides a cost-effective solution to your needs.

With QUICKDRAW, you can produce a wide variety of business graphs including bar charts, pie charts, scattergrams, high-low-close plots, and several types of histograms. Your plot lines can be distinguished by colour, dashing, thickness, shading pattern, or "pictures" at data points.

One distinctive feature of QUICKDRAW is the amount of control you have over graphic layout. You can produce a plot in any shape and size, and position it anywhere on the page. You also determine the exact placement of certain elements of the graph, such as the legend, axes, title, and various labels.

Workspace 3 *QUICKDRAW* contains a main plot function (*ΔPLOT*), over 90 state setting functions, and several utility functions. You can use the state setting functions to specify plot characteristics such as: position and range of the axes; placement and length of tiks along the axes; diameter and origin of pie charts; size, font, position, and slant of text; speed of the plotting pens (useful when you are producing transparencies); width of bars; and position and dash pattern of grid lines.

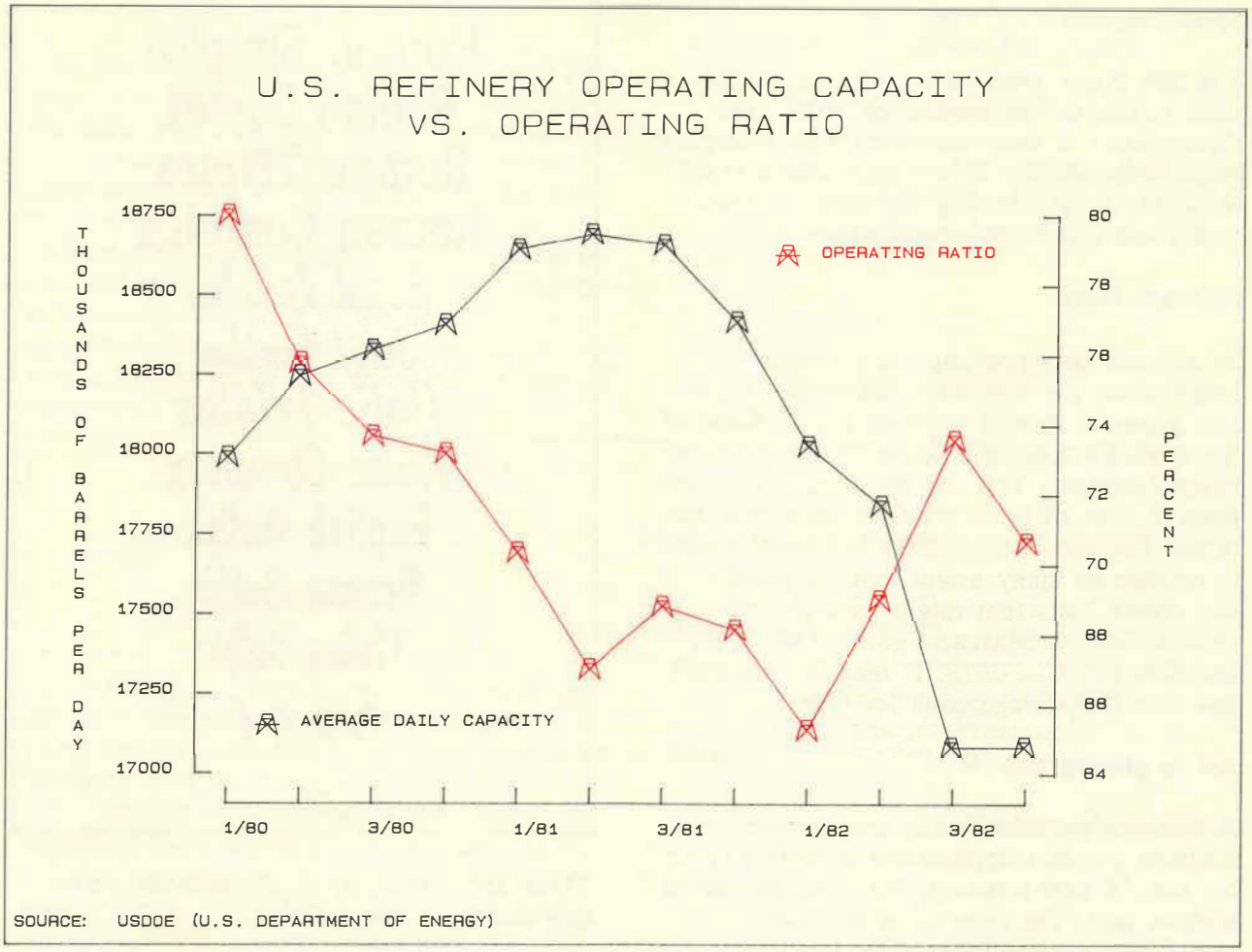
To simplify the process of producing a plot, there is a *DEFAULT* function that sets every state to its default setting. Before producing a plot, you just execute the state setting functions for states whose default setting is not appropriate. For example, the *DEFAULT* function sets a state of *NODASHING*, but you want your plot lines to be dashed. Thus you would execute the *DASHING* function and define your dash patterns.

The utility functions provide a number of advanced features not possible through the use of state setting functions. For example, functions are available for: operating the paper advance features of certain plotter models; performing tik, label, and date formatting for time series retrieved from any of the MAGIC data bases; and sorting the plotter output by pen number (for colour separations).

The *QUICKDRAW Users' Guide* is now available and can be obtained from your I.P. Sharp representative. All of QUICKDRAW's capabilities are fully described and well illustrated in this manual.

The following plot shows QUICKDRAW's capability to produce "pictures" at data points. ■

Walter Fil
Wendy Sexsmith, Toronto



It Pays to Convert to the New SAGA

A number of changes have been made to SHARP APL Graphics Aids (SAGA). SAGA, a device-independent graphics product, is the foundation for most I.P. Sharp graphics applications.

The new version of SAGA offers the following highlights:

- more efficient use of computer resources
- easier to use
- complete device independence by default
- software fonts to produce better looking text
- new functions to aid in text placement and to obtain device information
- support of a wider range of devices

For full documentation on line, type *NEWS* in workspace 3 *SAGA*.

The new version of SAGA is in workspace 3 *SAGA*; the old version has been moved to 499 *SAGAΔV1*. The old version will be available for a reasonable period of time. Thus all saved SAGA workspaces will continue to work as before.

Complete device independence

You can now write an application regardless of the device it will eventually be run on. A new state setting function *DEVIND*, for device independence, allows application programmers to stop the automatic resetting of normalized device coordinates (NDC) when a new device is selected. Thus you no longer need to know the size of the device that will produce the graphics output.

Specifying text

The new SAGA specifies text size in window units instead of centimetres or NDC units. Hence SAGA is more consistent with emerging graphics standards. This change allows more versatility by eliminating the need to recalculate text size to fit new windows.

Software fonts

In addition to supporting fonts already built into devices (i.e. hardware fonts), SAGA can now generate several software fonts. (Some of the available fonts are shown in the accompanying diagram.) You can also create your own fonts if none of those provided suits your purposes. The file 3 *SAGAFONTS* has been created to provide as many useful fonts as possible. If you create fonts that might have general use, submit them to MAILBOX group, *SAGAQ*, for inclusion in *SAGAFONTS*. If they are accepted, you save the storage costs for your font.

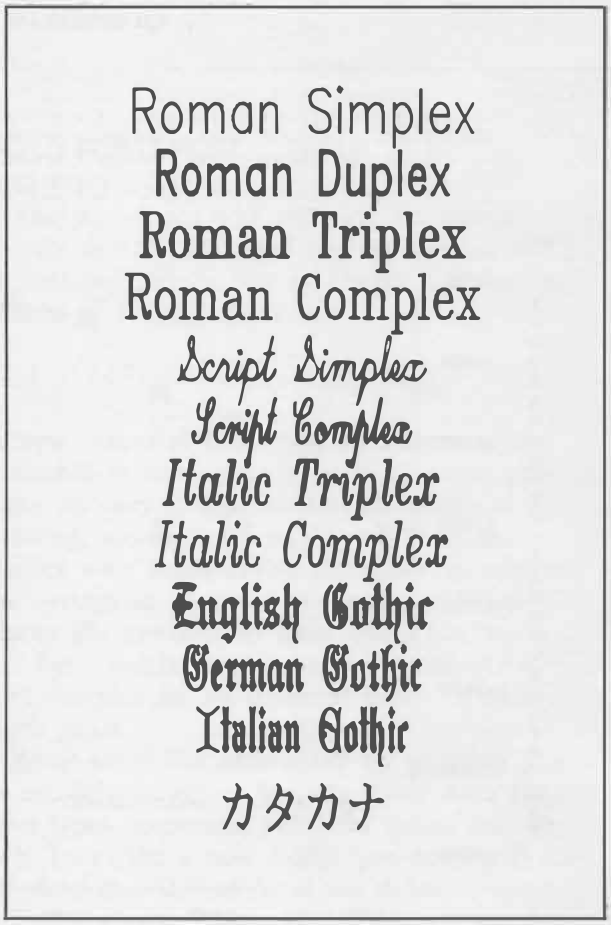
Aid to placing text

A common problem facing programmers in business graphics applications is finding space for text. A new function, *TEXTINQ*, returns in window units the dimensions of a box that would hold specified text. This function works for both hardware and software fonts. Given this information, you can decide where to place your text in the window.

New devices

Device support has been added for several new devices. These include: Hewlett-Packard flatbed plotters (HP9872B47A, HP9872C47A, HP7470A, and HP7470AU), the Ramtek 6211 ASCII colour video display terminal (RAM6211), the Tektronix 4027A ASCII colour video display terminal (TEK4027A), and the Xerox Diablo 1550 hardcopy terminal X1550). ■

Mike Holloway, Toronto



These are just a few of the software fonts now available through SAGA.

The Computerization of a Veterinary College

The University of Guelph, 100 kilometres west of Toronto, has been a SHARP APL in-house customer since the early 1970s. The University's SHARP APL service supports a total APL population of approximately 2 500 throughout the faculty, student body, and administration.

The University offers a wide assortment of academic programs, though it is probably best known for its expertise in agriculture and veterinary medicine. In fact, the Ontario Veterinary College (OVC) of the University of Guelph is one of only three such colleges in Canada.

OVC is both an academic institution and a veterinary medical facility. As an academic institution, the College has over 100 faculty members and approximately 500 undergraduate and graduate students. As a veterinary medical facility, the College has a 500-"bed" teaching hospital (handling admissions and discharges of about 40 animals per day), as well as an extensive research program.

In February 1980, OVC management decided that an integrated, computer-based system be developed to aid in the research, teaching, and administrative activities of the College. The proposed system should assist researchers in developing, monitoring, and evaluating data; assist teachers in the execution of their teaching programs; assist the hospital staff in providing patient care and support services; and provide record-keeping, billing, and communications.

A task force was set up to evaluate in depth the needs of the College and the tools available to satisfy those needs. As a result of its evaluation, SHARP APL was chosen as the environment in which to implement the Veterinary Medical Information Management System (VMIMS).

According to Barry Stahlbaum, staff analyst in the OVC Computer Group:

> "We evaluated various computing environments on the basis of their capability to deliver VMIMS from several points of view (user, operational, and technical). These considerations were used to measure the acceptability of alternate environments that were available on campus. In the final tally of the overall ranking of the various choices, the SHARP APL environment was selected.
>
> Many features available on SHARP APL (particularly enclosed arrays) have significantly reduced the overhead involved in the design and implementation of VMIMS. The use of I.P. Sharp products (such as MABRA, EDIT and WSDOC) satisfied our requirements for reporting, word processing, and documentation with no extra programming."

Because constant, continual communication with the end-user departments was deemed crucial to the success of this project, each functional area (e.g. Radiology, Anaesthesia) participates in the definition, planning, analysis, and implementation of its portion of VMIMS. This total end-user involvement has two major benefits for the project. First, it makes the technical people totally aware of and sympathetic to the needs and concerns of the users throughout the duration of the project. Second, the transition from paper to electronic record-keeping is less traumatic, since the users are involved from the earliest stages of development.

Muffy is admitted as the first client of VMIMS as Dr. Howard Clark, vice-president, University of Guelph, and his wife look on.

. . . Veterinary College

From start to finish, VMIMS development and implementation is expected to take four to five years. At full production, it will involve 70 video and 30 printer terminals and a user load of about 50 active users throughout the hospital. The data base itself will contain information on an estimated 12 000 admissions annually, with a retention period of ten years for most of the data for historical analysis and forecasting. This information will be available to students and faculty, in addition to the hospital staff.

On 2 February 1983, OVC celebrated the implementation of VMIMS in the Admitting and Medical Records sections of the hospital. At the opening ceremonies, Alan Meek D.V.M., coordinator of the project, commented on the future of VMIMS:

> "VMIMS will have considerable impact on all programs offered by the College. We will use computer technology to extend our mental capabilities, just as in the past we introduced the stethoscope and the X-ray to extend our ears and eyes, respectively."

OVC's computer system is the only one of its kind in Canada and, when completed, will be one of the most extensive in scope and application in North America.

If you wish more detailed information on the Veterinary Medical Information Management System, ask for the paper, "The Design and Implementation of an Integrated Computer System in a Veterinary College" by B.W. Stahlbaum and A.H. Meek. This paper was presented at the 1982 APL Users Meeting, and appears in the *Proceedings Volume I*, available from your I.P. Sharp office. ■

Dr. A. Meek, Dept. of Veterinary Microbiology and Immunology, University of Guelph
Brian Olson, I.P. Sharp Associates, London, Ontario

New 1981 Canadian Census Data Base

Canadian census data for 1981 is now available through workspace 39 *MAGIC*. CENSUS81 is of interest to demographers, planners and economists in the private sector, and at all levels of government. Information on cultural characteristics, demography, and family and household structure is available at several geographic levels:

Geographic Level	Level Code
Canada	*CA*
Province	*PV*
Census Division	*CD*
Census Subdivision	*SD*

New information will be added to the data base as it is released by Statistics Canada, the Canadian government's statistics agency.

CENSUS81 data comes in table form and can be accessed, one table at a time, for one or more geographic areas at a particular geographic level. For example, `'CUL1' CENSUS81 'PV/11,35'` produces a table of cultural characteristics—mother tongue by sex (*CUL1*) for the provinces of Newfoundland (11) and Ontario (35). All tables are automatically provided with a title and labels for use with MAGIC display. To obtain a list of the tables available at a particular level use `'TABLES' CENSUS81 'level'`. In the following example, `'TABLES' CENSUS81 'CA'` gives a list of table codes and descriptions for Canada.

Note: CENSUS81 data is not time series and hence the MAGIC state *NOTIMESERIES* must be set. If you try to access data with *TIMESERIES* set, a warning will be printed and the state reset to *NOTIMESERIES*.

For further information on CENSUS81 contact your I.P. Sharp representative or the MAILBOX group *ECOQ*. ■

Chris Taylor, Toronto

MAY/JUNE 1983
Volume 11 Number 3

I.P. Sharp
newsletter

technical
supplement 44

Decision Tables in APL

A **decision table** shows the conditions, relationships, and actions which constitute a complex logical process. You can use decision tables as you design and document systems. You can also build computer programs with them.

Decision tables may be familiar to you if you have used other business-oriented programming languages, particularly COBOL. Most APL users, however, seem to be unaware of their benefits. Decision tables enjoyed a period of popularity in the '60s [1,2]. They are now undergoing a renaissance through the influence of the advocates of "structured systems analysis" [3,4].

Decision tables have two features which naturally appeal to APL users. First, they describe logic in tables. They are thus like the rectangular data structures of conventional APL, and unlike the linear logic of most other programming languages. Second, they make extensive use of boolean manipulations. As you have gained experience with APL, you have probably become aware of the important part boolean logic plays in APL applications.

A standard decision table consists of four parts:

1) Condition stub
2) Condition entry
3) Action stub
4) Action entry

The **condition stub** is a set of statements. Each statement must evaluate to 1 or 0 (true or false).

The **condition entry** is a matrix. It contains all of the possible combinations of true and false results of each statement in the condition stub. One of the most useful features of decision tables is the ability to say that under specified conditions, the result of selected statements in the condition stub are irrelevant. These are known as *Don't Care* situations.

The **action stub** is a set of statements. One or more of these statements is executed, depending on the results of the condition stub statements.

The **action entry** is a matrix which defines which statements in the action stub are to be executed when the various combinations in the condition entry occur.

Decision tables are used to express complex logical problems. As the number of independent conditions which control the logic state increases, so does the utility of the decision table notation.

What sorts of problems can we solve with decision tables? They must be complicated enough to warrant the extra effort required. This means that there should be at least three independent conditions which determine the state of the process. There also must be a number of possible actions, usually a half dozen or more.

An example of such a problem is the process of "diamondizing" a function. The SHARP APL diamond makes it possible to place two or more statements on a single function line. Diamonds are most commonly used to connect statements when one of the following situations occur:

1) The programmer wants to see an entire function on a single display screen
2) The statements have a strong logical relationship
3) Workspace must be conserved
4) Every last milliunit of CPU time must be squeezed from the application

T1

. . . Decision Tables

The first two situations usually occur while a program is being developed, and the diamonds are inserted manually by the programmer. The latter two situations usually occur after the program is completed, and thus an automatic means of condensing the program is preferable.

The process of diamondizing a function is somewhat complicated. There are three independent conditions controlling the process, and thus eight (2*3) possible states. There are five distinct actions that may be taken. Branching implementations of this process often don't work.

What would the decision table look like? For each line in the source function, the following decision table would be evaluated.

Now a couple of these entries are superfluous. Since their actions columns are identical, rules 1 and 2 could be combined into *Yes, Yes, Don't Care.* In the same way, rules 3 and 4 could be combined into a single *Yes, No, Don't Care* column. This would give us a standard (three-valued) limited-entry decision table. As it stands now, it is a two-valued limited-entry table. But since we haven't explained how to handle *Don't Care* states yet, we will leave these duplications alone. The following program will diamondize canonical representations in the manner we have described.

```
    ∇ FUNCTION←MAX DIAMONDIZE IMAGE
      ;BUFFER;LINE;ACTLABELS;ACTENTRY
      ;ACTIONS;CODE;C1;C2;C3;WIDTH;LAMP
[1]   FUNCTION←IMAGE[,1;]
[2]   IMAGE← 1 0 ↓IMAGE
[3]   BUFFER←''
[4]   ACTLABELS←ACT1,ACT2,ACT3,ACT4,ACT5
[5]   ACTENTRY← 8 5 ρ 1 0 0 0 0, 0 1 0 1 0,
       1 1 0 0 1, 0 1 1 0 1, 0 1 0 1 0,
       0 1 0 1 0, 0 1 1 0 1, 0 1 1 0 1
[6] LINELOOP:→(0∊ρIMAGE)ρLINELOOPEND
[7]   LINE←IMAGE[1;]
[8]   IMAGE← 1 0 ↓IMAGE
[9]   LINE←(⌽∨\⌽LINE≠' ')/LINE
[10] CODE←=\LINE≠''''
[11] LAMP←CODE∧LINE='⍝'
[12] CODE←(CODE∧LAMP∨∧\~LAMP)/LINE
[13] C1←':'∊CODE
[14] C2←'⍝'∊CODE
[15] C3←MAX<(ρLINE)+ρBUFFER
[16] ACTIONS←ACTENTRY[1+2⊥C1,C2,C3;]/
     ACTLABELS
[17]ACTLOOP:→(0=ρACTIONS)ρACTLOOPEND
[18] →ACTIONS[1]
[19]ACT1:BUFFER←BUFFER,((0≠ρBUFFER)/
     ' ◇ '),LINE
[20] →CONTINUE
[21]ACT2:WIDTH←(¯1↑ρBUFFER)⌈¯1↑
     ρFUNCTION
[22] FUNCTION←(((1↑ρFUNCTION),WIDTH)↑
     FUNCTION),[1] WIDTH↑BUFFER
[23] →CONTINUE
[24]ACT3:WIDTH←(¯1↑ρLINE)⌈¯1↑ρFUNCTION
[25] FUNCTION←(((1↑ρFUNCTION),WIDTH)↑
     FUNCTION),[1] WIDTH↑LINE
[26] →CONTINUE
[27]ACT4:BUFFER←LINE
[28] →CONTINUE
[29]ACT5:BUFFER←''
[30]CONTINUE:ACTIONS←1↓ACTIONS
[31] →ACTLOOP
[32]ACTLOOPEND:→LINELOOP
[33]LINELOOPEND:→(0=ρBUFFER)ρEND
[34] WIDTH←(¯1↑ρBUFFER)⌈¯1↑ρFUNCTION
[35] FUNCTION←(((1↑ρFUNCTION),WIDTH)↑
     FUNCTION),[1] WIDTH↑BUFFER
[36]END:FUNCTION←(FUNCTION∨.≠' ')⌿
     FUNCTION ∇
```

Decision Table

Rule	1	2	3	4	5	6	7	8
C1: Label on this line?	*Y*	*Y*	*Y*	*Y*	*N*	*N*	*N*	*N*
C2: Comment on this line?	*Y*	*Y*	*N*	*N*	*Y*	*Y*	*N*	*N*
C3: Adding line will fill buffer?	*Y*	*N*	*Y*	*N*	*Y*	*N*	*Y*	*N*
A1: Append line to buffer						*X*		*X*
A2: Append buffer to function	*X*	*X*	*X*	*X*	*X*	*X*	*X*	
A3: Append line to function	*X*	*X*			*X*			
A4: Set buffer to line			*X*	*X*			*X*	
A5: Set buffer to empty	*X*	*X*			*X*	*X*		

. . . Decision Tables

Note that we use 2⊥ on line 16 to determine which state we are in. Then we use this value as an index to select a boolean vector from the set of condition entries. This boolean vector chooses the line labels which correspond to the necessary actions. By using this technique, we assume an implicit condition entry of ((number of conditions)ρ2)⊤0,⍳¯1+2* number of conditions.

There are several limitations to this method. First, it will only handle two-valued decision tables, but three-valued tables are far more useful. Second, the decision table handling is intertwined with the actual algorithm. What we would like is a general-purpose decision table utility function.

How .can you construct a general-purpose decision table processor in APL? The condition stub can be represented as a character matrix. It will have one APL statement in each row. Each statement must evaluate to a 0 or 1. The condition entry can also be represented as a character matrix. It will have one column for each relevant combination of conditions (rules). There will be one row for each statement in the condition stub. It may contain only three possible characters: *Y* for *Yes*, *N* for *No*, and - for *Don't Care*.

The action stub will also be a character matrix with one APL statement in each row. Both the condition stub and action stub could be a vector of enclosed character vectors. However, this does not give us any advantage. At the present time, if we want to disclose several enclosed elements of an array, they must have the same shape. The action entry will be a boolean matrix. It will have as many rows as the action stub and as many columns as the condition entry. A 1 means that the corresponding row in the action stub should be executed when the related column of the condition entry matches the concatenation of the results of the condition stub statements.

We will use four global variables to contain the four parts of the decision table:

CONDSTUB, *CONDENTRY*, *ACTSTUB*, *ACTENTRY*

These variables constitute the four parts of the decision table as described above.

In order to set up the decision table for use, you must do some initialization. Your program should start by executing the function listed below. It creates a function *STATE* which evaluates the state of the process, and prepares some global variables needed for executing the decision table. The *STATE* function returns a boolean vector. Its result is the concatenation of the results of executing each of the statements in the condition entry.

```
    ∇ SETUP∆DECTABX;JUNK;IMAGE
[1]  IMAGE←'STATE←',,',','(',CONDSTUB,
     ')'
[2]  JUNK←⎕FX((ρIMAGE)↑'STATE←STATE'),
     [⎕IO-0.5] IMAGE
[3]  TABLE←⍉CONDENTRY='Y'
[4]  MASK←⍉CONDENTRY≠'-'
[5]  RUNENTRY←⍉ACTENTRY=1 ∇
```

Your program can now efficiently evaluate the decision table by excuting the following function. It will work on any APL system. If you are using SHARP APL, you could also use diamonds to connect the statements without needing to fix a function.

```
    ∇ DO∆DECTABX;∆
[1]  ⍎⎕FX((¯1↑ρACTSTUB)↑'∆'),[⎕IO]
     (,(∧/TABLE=MASK∧(ρMASK)ρSTATE)⌿
     RUNENTRY)⌿ACTSTUB
```

This function uses the "Rule Mask" technique of Kirk [5] to evaluate a three-valued limited-entry decision table. It works as follows. If the condition entry is:

```
Y Y N N N N
Y N Y Y N N
- - Y N Y N
```

then the *MASK* variable will contain:

```
1 1 0
1 1 0
1 1 1
1 1 1
1 1 1
1 1 1
```

and the *TABLE* variable will contain the following:

```
1 1 0
1 0 0
0 1 1
0 1 0
0 0 1
0 0 0
```

If the *STATE* function returns 0 0 1, adding it with each row of the *MASK* variable will yield:

. . . Decision Tables

```
0 0 0
0 0 0
0 0 1
0 0 1
0 0 1
0 0 1
```

The fifth row of the *TABLE* variable matches the fifth row of this result, and so rule 5 is executed.

Typical decision table applications execute the decision table many times. Thus separating the setup from the actual execution should minimize CPU usage.

The APL implementation you use may handle decision tables more efficiently if execution is based on labels and branching, rather than the execute function. If this is so, you should use the following setup function. When you run it, it will create a *DO∆DECTABB* function which you can execute. (It also creates the same *STATE* function and global variables as *SETUP∆DECTABX* did.)

```
    ∇ SETUP∆DECTABB;JUNK;IMAGE
      ;LABELS;STRING;BODY;TOP;⎕IO
[1]   ⎕IO←1
[2]   IMAGE←'STATE←',,',',('⍕CONDSTUB,
      ')'
[3]   JUNK←⎕FX((⍴IMAGE)↑'STATE←STATE'),
      [0.5] IMAGE
[4]   TABLE←⍉CONDENTRY='Y'
[5]   MASK←⍉CONDENTRY≠'-'
[6]   RUNENTRY←⍉ACTENTRY=1
[7]   LABELS←⍳1↑⍴ACTENTRY
[8]   LABELS←⍕((⍴LABELS),1)⍴LABELS
[9]   LABELS←(((1↑⍴LABELS),3)⍴'ACT'),
      (+/∧\' '=LABELS)⌽LABELS
[10]  STRING←'LABELS←',1↓,',',LABELS
[11]  BODY←LABELS,':',ACTSTUB
[12]  BODY←((2×1↑⍴BODY)⍴ 1 0)⍉BODY
[13]  BODY[2×⍳(1↑⍴BODY)÷2;⍳5]←
      (((1↑⍴BODY)÷2),5)⍴'→LOOP'
[14]  TOP←'DO∆DECTABB;LABELS;NEXT'
      ∆JOIN STRING ∆JOIN
      'LABELS←(,(∧/TABLE=MASK∧
      (⍴MASK)⍴STATE)⌿RUNENTRY)/LABELS'
[15]  TOP←TOP ∆JOIN 'LOOP:→(0=⍴LABELS)/
      END' ∆JOIN 'NEXT←1↑LABELS' ∆JOIN
      'LABELS←1↓LABELS' ∆JOIN '→NEXT'
[16]  IMAGE←TOP ∆JOIN BODY ∆JOIN 'END:'
[17]  JUNK←⎕FX IMAGE ∇
```

If the action stub had five entries, the function it would create would look like the following:

```
    ∇ DO∆DECTABB;LABELS;NEXT
[1]   LABELS←ACT1,ACT2,ACT3,ACT4,ACT5
[2]   LABELS←(,(∧/TABLE=MASK∧(⍴MASK)⍴
      STATE)⌿RUNENTRY)/LABELS
[3]  LOOP:→(0=⍴LABELS)/END
[4]   NEXT←1↑LABELS
[5]   LABELS←1↓LABELS
[6]   →NEXT
[7]  ACT1:  action 1
[8]   →LOOP
[9]  ACT2:  action 2
[10] →LOOP
[11]ACT3:  action 3
[12] →LOOP
[13]ACT4:  action 4
[14] →LOOP
[15]ACT5:  action 5
[16] →LOOP
[17]END: ∇
```

Decision tables are a very useful method for representing complex logical processes. By using APL's powerful boolean operations, they can be efficiently evaluated. They are thus an effective alternative to traditional branching methods. If you would like to study these functions further, you can find them in workspace 777 *DECTAB* on the SHARP APL Service.

References

[1] London, Keith. *Decision Tables*, Auerbach Publishers, Princeton, N.J., 1972.

[2] Pollack, Solomon, Harry Hicks, and William Harrison. *Decision Tables: Theory and Practice*, Wiley-Interscience, New York, 1971.

[3] DeMarco, Tom. *Structured Analysis and System Specification*, Yourdon, New York, 1978, pp. 215-226.

[4] Gane, Chris and Trish Sarson. *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Englewood Cliffs, N.J., 1979, pp. 82, 88-95.

[5] Kirk, H.W. "Use of Decision Tables in Computer Programming", *Communications of the ACM*, vol. 8, no. 1 (January 1965), pp. 41-43.

Acknowledgements: My thanks to Doug Keenan for his assistance in preparing this article. ■

Robert Metzger, Rochester

... CENSUS81

CULTURAL CHARACTERISTICS - MOTHER LANGUAGE (9) BY SEX (3)			
	TOTAL	MALE	FEMALE
CANADA			
ENGLISH	14,918,460	7,396,495	7,521,965
FRENCH	6,249,100	3,070,905	3,178,190
GERMAN	522,855	256,085	266,770
INDIAN ∊ INUKTITUT	146,285	73,275	73,005
ITALIAN	528,780	275,425	253,355
NETHERLANDIC	156,645	79,670	76,975
POLISH	127,960	63,350	64,615
UKRAINIAN	292,265	143,695	148,570
OTHERS	1,400,835	709,385	691,455

Bank for International Settlements Data Base

The Bank for International Settlements (BIS) data base contains total assets and liabilities for 200 countries, and some regions. This data is collected from commercial banks (mainly in industrial countries) that report to the Bank for International Settlements. Data is available quarterly and semiannually from January 1979. The data is expressed in millions of United States dollars.

The Bank for International Settlements is one of the few sources available for debt data. It is the central bank for all central banks in the world and is based in Basle, Switzerland. This data is of interest to banks or organizations that are active in lending money to foreign countries.

The following facts are available for each data base:

Quarterly (*BISQ*)

LI Liabilities
AS Assets

Semiannual (*BISS*)

LI Liabilities
AS Assets
AS1 Assets up to 1 year
AS2 Assets 1-2 years
AS3 Assets over 2 years
ASO Assets unallocated
UCC Undisbursed credit commitments

The BIS data base is available through workspace 39 *MAGIC*. A directory of the facts and the country codes is produced by typing:

BISQ 'DIRECTORY' (for quarterly data)
or *BISS 'DIRECTORY'* (for semiannual data)

An online description of the data base is available by typing either:

BISQ 'DESCRIBE'
or *BISS 'DESCRIBE'*

There is no subscription fee to access this data base.

The following examples illustrate the use of the BIS data base using MAGIC. ■

Ann Rockley, Toronto

...BIS

EXTERNAL POSITIONS OF REPORTING COMMERCIAL BANKS
TOTAL ASSETS AND MATURITY DISTRIBUTION IN ∘/∘

	HUNGARY				
	TOTAL ASSETS IN $MILL.	UP TO 1 YEAR IN ∘/∘	1-2 YEARS IN ∘/∘	OVER 2 YEARS IN ∘/∘	UNALLOCAT IN ∘/∘
1HF/80	7,768	42.2	8.2	41.3	8.3
2HF/80	8,002	42.9	8.5	41.3	7.4
1HF/81	6,991	35.9	8.2	48.5	7.4
2HF/81	7,714	40.4	7.2	45.0	7.4
1HF/82	6,418	33.2	8.6	49.3	8.9

	ROMANIA				
	TOTAL ASSETS IN $MILL.	UP TO 1 YEAR IN ∘/∘	1-2 YEARS IN ∘/∘	OVER 2 YEARS IN ∘/∘	UNALLOCAT IN ∘/∘
1HF/80	5,550	43.3	5.8	28.2	22.7
2HF/80	5,776	42.7	6.3	28.2	22.8
1HF/81	5,428	43.1	8.5	25.0	23.4
2HF/81	5,067	35.3	10.5	27.5	26.7
1HF/82	4,469	40.3	8.3	28.8	22.5

	POLAND				
	TOTAL ASSETS IN $MILL.	UP TO 1 YEAR IN ∘/∘	1-2 YEARS IN ∘/∘	OVER 2 YEARS IN ∘/∘	UNALLOCAT IN ∘/∘
1HF/80	15,752	30.8	16.7	35.9	16.6
2HF/80	16,173	33.1	16.7	34.1	16.2
1HF/81	14,699	32.3	15.3	35.8	16.6
2HF/81	15,228	36.1	12.4	34.7	16.7
1HF/82	13,797	34.3	11.7	37.3	16.7

SOURCE: PUBLICATION OF BANK FOR INTERNATIONAL SETTLEMENTS

See *When* They Run

You can now ascertain when a data base has been most recently updated before accessing the data. The new function *TIMECHECK* in workspace 39 *MAGIC* shows the date of the most recent data at the last update, the data base name, the time at which the update was finished, and any comments. The comment is provided so that in the event of any exceptional circumstances, such as the necessity for a retransmission of update data, you can see the reason for the delay and an estimated time of completion. For example:

```
      TIMECHECK 'INS,SITC'

UPDATED TO  DATA BASE          DATE(TIME) OF UPDATE     COMMENTS

 DEC 1982   INS                 5 APR 1983 (18:59 UTC)
            SITC               11 FEB 1983 (19:40 UTC)
```

The INS and SITC data bases are checked for the most recent updates in this example. There are no comments because there were no problems with the updates.

TIMECHECK uses a standard short name for each data base. Often this is the same name you use to access the data in 39 *MAGIC*; however, some names differ. To get a complete list of *TIMECHECK* short names and their full description, type:

```
TIMECHECK 'LIST'
```

For some data bases, the *UPDATED TO* field is not meaningful since various series contained in a data base are updated at different points in the cycle. In these cases, it is sufficient to know the date and time of the last update. For these data bases, such as SITC in the example above, the *UPDATED TO* field is blank.

You can also get a complete report on the status of all data bases within a certain group by entering the group name. For example, to produce a list of all the aviation data bases and the dates to which they are current, type:

```
TIMECHECK 'AVIATION'
```

Alternatively, you can produce a report on all data bases that have a common string at the start of their names. For example:

```
TIMECHECK 'CURRENCY'
```

produces a list of all the currency data bases (*CURRENCYC*, *CURRENCYH*, etc.)

For an online description of *TIMECHECK*, type:

```
)LOAD 39 MAGIC

TIMECHECK 'DESCRIBE'
```

For further information on *TIMECHECK* contact your I.P. Sharp representative, or the MAILBOX group *DATA*. ■

Audrey Sharp, Toronto

New Actuarial Products

Two new actuarial products are now available on the SHARP APL Service. A completely new version of ACTPAK is available through workspace 623 *ACT*, and a newly released actuarial profit testing system is in workspace 123 *ASAP*.

ACTPAK has been available on the SHARP APL Service since the early 1970s. Now we have taken advantage of extensions to SHARP APL and improved coding techniques to completely redesign the package. All ACTPAK functions have been rewritten to use a standard syntax, making the new version much simpler to use. As well, ACTPAK is now more flexible and efficient.

One major change is that no distinction is made between calculations in select or varying interest modes, and those in ultimate and fixed interest mode. Previously, state settings set appropriate APL functions to calculate the commutation tables. Now a single APL function, *CMFNS*, handles all such calculations, and the mode of calculation is transparent to the user. To reduce costs and the likelihood of *WS FULL*, commutation tables are now calculated in an N-task.

Another significant change is in the method of passing parameters to the actuarial functions. In the new version, all functions are monadic, and parameters are given in a standard order with no limit on their rank. Previously, both monadic and dyadic functions were used in ACTPAK because some functions required only one parameter, while others needed up to five. In some cases, you had to combine parameters before passing them to the actuarial functions; this method also limited the rank of the parameters to 0 or 1.

The following example shows the calculation of monthly temporary annuities-due guaranteed five years, with terms of 10 to 15 years inclusive, for ages 55, 60 and 65, using female annuitant mortality, recently published in the United States. The interest rate used is 10% for the first 5 years, 9.5% for the next 5, and 9% thereafter.

```
      )LOAD 623 ACT

      'BGA83F' CMFNS .10 5 .095 5 .09

      GANDXNM 55 60 65 FOR (10+⍳6) GUARANTEED 5 PAYABLE 12
7.931936 8.154627 8.355947 8.537565 8.701009 8.84769
7.813135 8.01861  8.201807 8.36452  8.508442 8.635169
7.596456 7.77239  7.925736 8.058637 8.173119 8.27109
```

The workspace 623 *ACT* will be moved shortly to 123 *ACT*, and the old version will be retained indefinitely in library 499 for the benefit of current users.

123 *ASAP*

A new actuarial profit testing system is now available in workspace 123 *ASAP*. *ASAP* provides an easy-to-use, interactive means of defining, storing, and modifying asset share studies with considerable flexibility in tailoring the system to your requirements. *ASAP* replaces the generalized asset share package that has been available to users of the SHARP APL Service on an informal basis for about three years.

ASAP simplifies data entry by allowing you to set default values for any study data (age, plan types, expenses, actuarial bases, and so on). To define a new study, you need only enter data that differs from the default values.

For example, if you are creating a number of studies that use the same interest and expense assumptions throughout, you may set these assumptions prior to creating the new studies. Then you can request that subsequent prompts for these assumptions be suppressed. Similarly you may suppress prompts for data that you never use. For example, an actuary interested only in renewable term business may wish to remove any reference to cash values and policy dividends.

You can enter data easily with a number of utilities provided in *ASAP*. Mortality and other actuarial data can be retrieved directly from the actuarial data base, while functions to produce standard reserving and cash value factors are included. Both asset-share and emerging-cost calculations are performed, and you may specify which results are to be displayed on completion of the calculations. Overall profitability is determined by discounting

actuarial insurance

... New Actuarial Products

emerging profits back to the date of issue of the policy, and the internal rate of return is calculated where appropriate.

For further information on the new ACTPAK or *ASAP*, contact the actuarial support group in Toronto, or the MAILBOX group *ACTQ*. ■

Chris Burke, Toronto

conferences

APL Australia 83

More than a hundred people gathered in Sydney in February for the first Australian APL conference, organized by the Australian APL Users Group (APLUG). The keynote speakers were Ken Iverson of I.P. Sharp Associates, Toronto, and Ray Polivka of IBM, Poughkeepsie, New York.

Polivka opened the conference with some personal perspectives on APL, gained from his many years of experience with the language.

Lib Gibson of I.P. Sharp, Toronto, spoke on "Designing User-friendly APL Systems". She highlighted the concepts of open and closed systems, explaining how either approach to system design can result in systems that are *friendly* to the end user.

In his paper "Access to Functions", Stephen Taylor, of I.P. Sharp, Sydney, proposed that function definition be extended to allow functions to have objects permanently bound to them. Gary Gould, also of I.P. Sharp, Sydney, talked about "A Relational Data Base Management System in APL", a system he developed with other I.P. Sharp staff in Sydney.

Neville Holmes, president of APLUG, humorously traced the development of APL in Australia at the conference dinner. In 1970, APL history was made in Australia with the installation of an interpreter on the IBM System /360 Model 67 at the IBM Systems Development Institute in Canberra. Since then, the use of APL has grown to the point where many major organizations within the country are using the language extensively.

The conference closed with Iverson's talk on "Teaching APL", in which he reiterated the theme of the current lack of adequate APL training. He stressed the importance of reference materials in the teaching of any discipline, including APL programming, and fostering independent studies through experimentation. Following this philosophy, Iverson developed the course *A Working Introduction to APL*. ■

Frank Arthur, Sydney

bulletin board

Helsinki

TMT–Team Oy, our representatives in Helsinki, have moved their offices to:

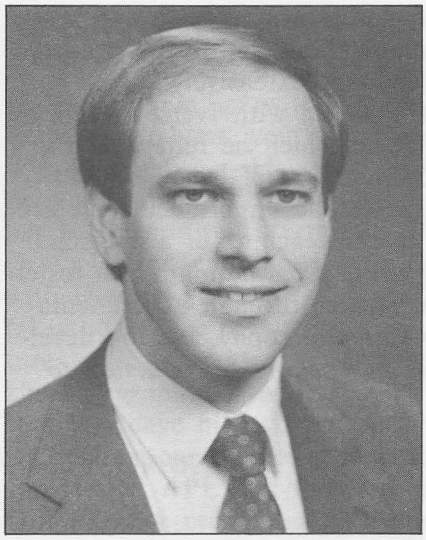
Westendintie 1
SF-02160
Espoo 16, Finland

Phone: (0) 4521155

London

Our office in London, Ontario has a new telephone number:

519 673-4060



Philadelphia

Laurin McArthur is the new branch manager in Philadelphia. Before moving to Philadelphia, Laurin worked in our New York office, and was involved primarily with in-house conversions.

Laurin is a graduate of the University of South Carolina with a B.A. in European history and an M.A. in secondary education.

Laurin's move to I.P. Sharp was a natural progression after having worked with the South Carolina Department of Education, Management Information Section, as a programmer. There, he became an avid APL user for research projects and statistical reports and publications. ■

Dallas

Josh Levine, formerly of the Applications Software Department in corporate headquarters (*I.P. Sharp Newsletter*, March/April 1982), has been appointed branch manager in Dallas. ■

publications

New from APL PRESS

"APL Quote Quad": The Early Years
ISBN 0-917326-13-X, $35.00 U.S., 468 pp.

This book contains the entire first three volumes (April 1969–9 June 1972) of the APL newsletter, *APL Quote Quad*. These first 14 issues—long out of print—contain articles and correspondence of lasting interest to the APL community. Among the many papers are Larry Breed's "The APL PLUS File Subsystem", Jim Brown's "Using the Ackermann Function to Rate Programming Languages", and Mike Jenkins' "Domino—An APL Primitive Function for Matrix Inversion—Its Implementation and Applications". Phil Abrams, Art Anger, Bob Korsan, Alan Perlis, Al Rose, Clark Wiedmann, and many others number among the contributors.

These early newsletters predate the formation of ACM's SIGPLAN Technical Committee on APL (STAPL), which later assumed the responsibility for publishing *APL Quote Quad* as a quarterly.

APL Blossom Time, 45 rpm, $5.00 U.S.

J.C.L. Guest's witty musical tribute to the pioneers of APL features Jim Brown and company rendering the tune in a professional studio recording on side 1. Side 2 offers the historic premier performance by the "APL Corral" and a cast of hundreds at APL81 in San Francisco. The jacket includes the complete lyrics and the likeness of J.C.L. Guest (a.k.a. Mike Montalbano) from the APL PRESS Rogues' Gallery of APL Characters.

Coming up soon

Probability in APL and *Statistics in APL*

Two new booklets by Linda Alvord, Mathematics Department Chairperson, Scotch Plains–Fanwood High School, Scotch Plains, New Jersey, are currently in production. These supplementary texts for high school courses present APL approaches to probability theory and statistics by an educator who has used them in her classes. They include APL exam-

... APL PRESS

ples, functions, exercises, problems, and solutions.

In her books, Alvord explains APL primitive functions and how to use them to solve problems on an APL computer.

For information on APL PRESS publications, contact:

MollieO Patrick, Business Manager
APL PRESS
220 California Avenue
Palo Alto, California 94306

Phone: 415 327–1700 ■

Arlene Azzarello, Palo Alto

mailing request

- ☐ Please amend my mailing address as indicated.
- ☐ Please add the following name(s) to your Newsletter mailing list.
- ☐ Please send me a Publications Order Form.
- ☐ Please add my name to the Aviation Newsletter mailing list.
- ☐ Please add my name to the Energy Newsletter mailing list.
- ☐ Please add my name to the Financial and Economic Newsletter mailing list.
- ☐ Please add my name to the Promis Newsletter mailing list.

Name: _____

Title: _____

Company: _____

Address: _____

_____

_____

☐ Please send me information about your courses in:

City: _____ Tel.: _____

The *I.P. Sharp Newsletter* is published by I.P. Sharp Associates, Suite 1900, 2 First Canadian Place, Toronto, Canada M5X 1E3.
Your comments and contributions are welcome.

Editor: Irene Shimoda
Circulation: Mary Kopfensteiner
Printed in Canada
May 1983
ISSN 0226 854X

international offices

Aberdeen
I.P. Sharp Associates Limited
5 Bon Accord Crescent
Aberdeen AB1 2DH
Scotland
(0224) 25298

Amsterdam
InterSystems BV
Kabelweg 47
1014 BA Amsterdam
The Netherlands
(020) 86 80 11
Telex: 18795 ITS NL

Atlanta
I.P. Sharp Associates, Inc.
1210 S. Omni International
Atlanta, Georgia 30335
(404) 586-9600

Boston
I.P. Sharp Associates, Inc.
1 Liberty Square
Boston, Massachusetts 02109
(617) 542-2313

Brisbane
I.P. Sharp Associates Pty. Ltd.
6th Floor, Dalgety House
79 Eagle Street
Brisbane, Queensland 4000
Australia
(07) 229-8330

Brussels
I.P. Sharp Europe SA
Boulevard de la Cambre 36, bte 5
1050 Bruxelles
Belgium
(02) 649 99 77

Calgary
I.P. Sharp Associates Limited
Suite 550, Bow Valley Square 4
250-6th Avenue S.W.
Calgary, Alberta T2P 3H7
(403) 265-7730

Canberra
I.P. Sharp Associates Pty. Ltd.
16 National Circuit
Barton, A.C.T. 2600
Australia
(062) 73-3700

Chicago
I.P. Sharp Associates, Inc.
Suite 3860
55 West Monroe Street
Chicago, Illinois 60603
(312) 782-3177

Copenhagen
I.P. Sharp ApS
Østergade 24B
1100 Copenhagen K
Denmark
(01) 11 24 34

Coventry
I.P. Sharp Associates Limited
7th Floor B Block
Coventry Point, Market Way
Coventry CV1 1EA England
(0203) 56562

Dallas
I.P. Sharp Associates, Inc.
Suite 1148, Campbell Center
8350 North Central Expressway
Dallas, Texas 75206
(214) 369-1131

Denver
I.P. Sharp Associates, Inc.
Suite 416
5680 South Syracuse Circle
Englewood, Colorado 80111
(303) 741-4404

Dublin
I.P. Sharp Associates Limited
Segrave House
Earlsfort Terrace
Dublin 2, Ireland
(01) 763605

Düsseldorf
I.P. Sharp GmbH
Kaiserswertherstrasse 115
4000 Düsseldorf 30
West Germany
(0211) 45 20 52

Edmonton
I.P. Sharp Associates Limited
2358 Principal Plaza
10303 Jasper Avenue
Edmonton, Alberta T5J 3N6
(403) 428-6744

Halifax
I.P. Sharp Associates Limited
Suite 706, Cogswell Tower
2000 Barrington Street
Halifax, Nova Scotia
B3J 3K1
(902) 423-6251

Helsinki
TMT-Team Oy (Agent)
Westendintie 1
SF-02160
Espoo 16, Finland
(0) 452 1155

Hong Kong
I.P. Sharp Associates (HK) Limited
Suite 606, Tower 1
Admiralty Centre, Hong Kong
5-294341

Houston
I.P. Sharp Associates, Inc.
Suite 375, One Corporate Square
2600 Southwest Freeway
Houston, Texas 77098
(713) 526-5275

London, Canada
I.P. Sharp Associates Limited
Suite 510, 220 Dundas Street
London, Ontario N6A 1H3
(519) 673-4060

London, England
(European Headquarters)
I.P. Sharp Associates Limited
132 Buckingham Palace Road
London SW1W 9SA, England
(01) 730-4567
Telex: 8954178 SHARP G

Los Angeles
I.P. Sharp Associates, Inc.
Suite 1230
1801 Century Park East
Los Angeles, Ca. 90067
(213) 277-3878

Madrid
I.P. Sharp Associates Limited
Serrano 23, Piso 8
Madrid 1, Spain
(91) 276 70 54

Melbourne
I.P. Sharp Associates Pty. Ltd.
520 Collins St., 7th Floor
Melbourne, Victoria
3000, Australia
(03) 614-1766

Mexico City
Teleinformatica de Mexico SA
(Agent) Mail to:
Arenal N 40, Chimalistac
Mexico 20 D.F., Mexico
(905) 550-8033

Miami
I.P. Sharp Associates, Inc.
Suite 240
15327 N.W. 60th Avenue
Miami Lakes, Florida 33014
(305) 556-0577

Milan
Informatical Society Italia Srl
(Agent)
Via Eustachi 11
20129 Milan, Italy
(02) 221 612

Montreal
I.P. Sharp Associates Limited
Suite 1610
555 Dorchester Boulevard W.
Montreal, Quebec H2Z 1B1
(514) 866-4981

New York
I.P. Sharp Associates, Inc.
Suite 210
230 Park Avenue
New York, N.Y. 10169
(212) 557-7900

Newport Beach
I.P. Sharp Associates, Inc.
Suite 1135
610 Newport Center Drive
Newport Beach, Ca. 92660
(714) 644-5112

Oslo
I.P. Sharp A/S
Postboks 486 Sentrum
Dronningens gate 34
Oslo 1, Norway
(02) 41 17 04

Ottawa
I.P. Sharp Associates Limited
Suite 600, 265 Carling Ave.
Ottawa, Ontario K1S 2E1
(613) 236-9942

Palo Alto
I.P. Sharp Associates, Inc.
Suite 201, 220 California Ave.
Palo Alto, Ca. 94306
(415) 327-1700

Paris
I.P. Sharp SARL
9 Rue du Cirque
75008 Paris
France
(1) 225 98 20

Philadelphia
I.P. Sharp Associates, Inc.
Suite 604, 437 Chestnut St.
Philadelphia, Pa. 19106
(215) 925-8010

Phoenix
I.P. Sharp Associates, Inc.
Suite 503
3033 N. Central Avenue
Phoenix, Arizona 85012
(602) 264-6819

Rochester
(United States Headquarters)
I.P. Sharp Associates, Inc.
1200 First Federal Plaza
Rochester, N.Y. 14614
(716) 546-7270

Rome
Informatical Society Italia Srl
(Agent)
Piazza Della Rotonda 2
00100 Rome, Italy
(06) 656-5925

San Francisco
I.P. Sharp Associates, Inc.
Suite C-415, 900 North Point St.
San Francisco, Ca. 94109
(415) 673-4930

Saskatoon
I.P. Sharp Associates Limited
Suite 303, Financial Bldg.
230-22nd St. E.
Saskatoon, Saskatchewan
S7K 0E9
(306) 664-4480

Seattle
I.P. Sharp Associates, Inc.
Suite 223, Executive Plaza East
12835 Bellevue Redmond Road
Bellevue, Washington 98005
(206) 453-1661

Seoul
Daewoo Corporation (Agent)
541, 5-Ga, Namdaemoon-Ro
Jung-Gu (CPO Box 2810)
8269 Seoul, Korea.
771-91/2
Telex: 23341-5/24295 DAEWOO K

Singapore (Far East H.Q.)
I.P. Sharp Associates(S) Pte. Ltd.
Suite 1601, CPF Building
79 Robinson Road
Singapore 0106
Republic of Singapore
2230211
Telex: 20597 IPSAS RS

Singapore
Singapore International
Software Services Pte. Ltd.
(Agent)
Suite 1601, CPF Building
79 Robinson Road
Singapore 0106
Republic of Singapore
2230211

Stockholm
I.P. Sharp AB
Kungsgatan 65
S111 22 Stockholm, Sweden
(08) 21 10 19

Sydney (Australian H.Q.)
I.P. Sharp Associates Pty. Ltd.
8th Floor, Carlton Centre
55 Elizabeth St.
Sydney, New South Wales 2000
Australia
(02) 232-6366

Tokyo
INTEC, Inc. (Agent)
37-18, 3-chome, Hatagaya
Shibuya-ku, Tokyo 151
Japan
(03) 320-2020
Telex: 2322008 INTECA J

Toronto
(International Headquarters)
I.P. Sharp Associates Limited
Suite 1900
2 First Canadian Place
Toronto, Ontario
M5X 1E3
(416) 364-5361
Telex: 0622259 I P SHARP TOR

Vancouver
I.P. Sharp Associates Limited
Suite 902, 700 West Pender St.
Vancouver, B.C. V6C 1G8
(604) 687-8991

Victoria
I.P. Sharp Associates Limited
Chancery Court
1218 Langley Street
Victoria, B.C. V8W 1W2
(604) 388-6365

Vienna
I.P. Sharp Ges.mbH
Renngasse 4
A-1010 Wien, Austria
(0222) 66 42 48

Warrington
I.P. Sharp Associates Limited
1-3 Dolmans Lane
Warrington, Cheshire
WA1 2ED England
(0925) 50413/4

Washington
I.P. Sharp Associates, Inc.
Suite 305, 2033 K Street N.W.
Washington, D.C. 20006
(202) 293-2915

Wayne
I.P. Sharp Associates, Inc.
Suite 303
155 Willowbrook Blvd.
Wayne, New Jersey 07470
(201) 785-8050

White Plains
I.P. Sharp Associates, Inc.
Suite 312 West
701 Westchester Avenue
White Plains, New York 10604
(914) 328-8520

Winnipeg
I.P. Sharp Associates Limited
Suite 208
213 Notre Dame Avenue
Winnipeg, Manitoba R3B 1N3
(204) 947-1241

Zurich
I.P. Sharp AG
Fortunagasse 15
CH-8001 Zurich, Switzerland
(01) 211 84 24

I.P. Sharp communications network

APL OPERATOR VOICE (416) 363-2051 COMMUNICATIONS (416) 363-1832

Our private, packet-switched network connects with the Value Added Networks—Datapac, Datex-P, PSS, Telenet, Telepac, Transpac, and Tymnet—to provide access from the 46 countries listed below:

• Argentina • Australia • Austria • Bahamas • Bahrain • Barbados • Belgium • Bermuda • Brazil • Canada • Chile • Denmark • Dominican Republic • Finland • France • French Antilles • Germany • Great Britain • Hong Kong • Hungary • Ireland • Israel • Italy • Japan • Korea • Kuwait • Luxembourg • Mexico • The Netherlands • New Zealand • Norway • The Philippines • Portugal • Puerto Rico • Qatar • Republic of South Africa • Saudi Arabia • Scotland • Singapore • Spain • Sweden • Switzerland • Taiwan • United Arab Emirates • U.S.A. • Virgin Islands

SHARP APL is accessible from over 600 places via a local telephone call. Please ask your nearest I.P. Sharp office or representative for a complete list of access points and access procedures. Our private network also connects with the worldwide Telex network via the Amsterdam and Rochester, New York nodes.