

the I.P. Sharp newsletter

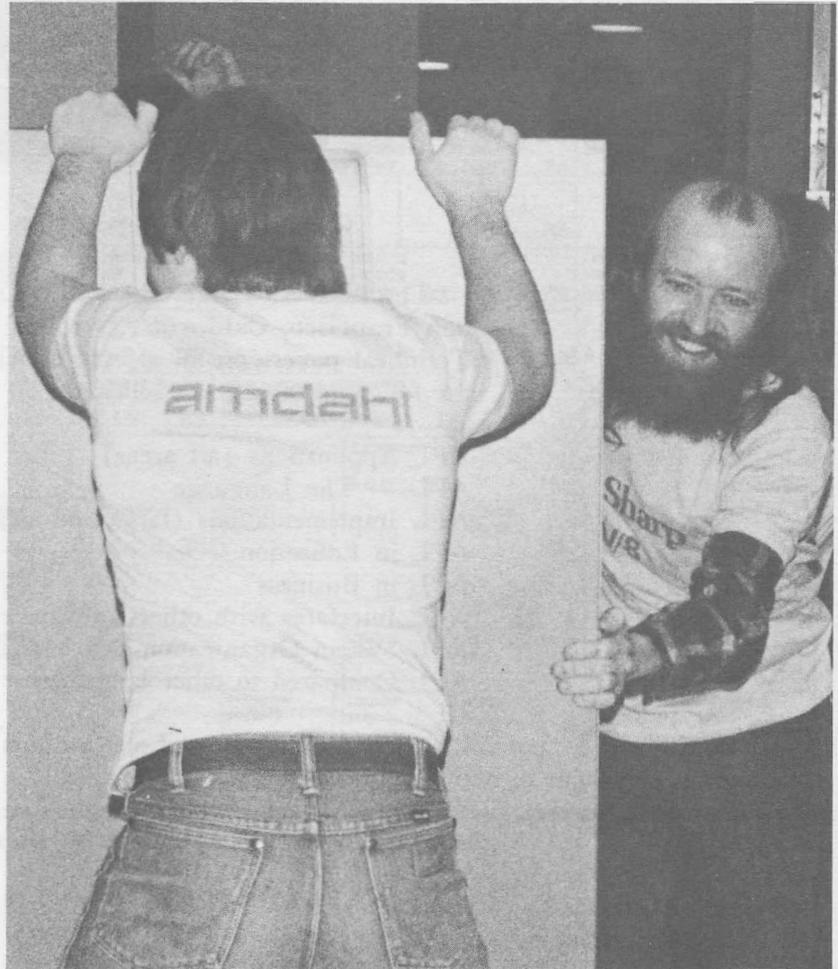
May/June 1980
Vol. 8 Number 3

NEW AMDAHL V/8

One year ago, we made our APL facility the largest in the world by installing a second Amdahl 470/V6-II. That installation was expected to handle the company's rate of growth until the spring of 1981; however, increased demand for computer resources both in North America and Europe has necessitated an upgrade in hardware in the spring of 1980.

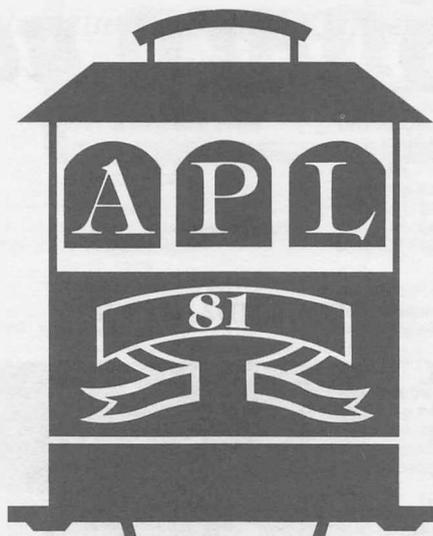
An Amdahl 470/V8, capable of half again as much processing power as a V6, replaced one of Sharp's existing V6's during March. Both machines have 6 megabytes of storage capacity, increasing the existing main memory of the Sharp facility by 4 million bytes. A new Memorex disk system has been installed. The eight double-density 3350-style spindles look like sixteen single-density spindles with 317 megabytes each.

With the added capacity of the Amdahl V8, the Sharp APL computer centre will continue to service our worldwide time-sharing customers with the needed speed, reliability and response time, as well as to allow for increased expansion and growth over the next year.



CONTENTS

New Amdahl V8	1	Private data and MAGIC.....	8
APL81 Call for Papers.....	2	People.....	10
IMPAS.....	3	Network Analysis.....	11
STARS.....	4	Applications library update.....	12
Energy: New DOE data base	5	APL80	12
Seminars planned in Europe.....	5	Course Schedule	13
Sharp literature.....	6		
APL PRESS moves	6	Technical Supplement 26	
APL Standards: ANSC X3 committee	7	SATN vs The Reference Manual	T1
Syracuse conference	7	Branching V	T1



CALL FOR PAPERS

The APL 81 Conference will be held in **San Francisco, California, October 21-23 1981**. Technical papers on all aspects of APL, including the following areas, are solicited.

APL Applications (all areas)
 APL — The Language
 APL Implementations (large and small systems)
 APL in Education
 APL in Business
 APL Interfaces with other software systems
 APL System Organization and Management
 APL Compared to other languages

Both abstracts and full papers will be refereed and authors should submit these documents to the program chairman in accord with the following schedule:

	Due Date	Author Notification
Abstracts	October 1, 1980	December 1, 1980
Full Papers	February 1, 1981	April 15, 1981

Final copies of complete papers must be received by the program chairman by June 15, 1981 for inclusion in the conference proceedings and for presentation at the conference. Authors may submit abstracts using the facilities of SHARP APL; directions may be found in the workspace 1 *APL81*.

Further information may be obtained from either the Program Chairman or from the conference General Chairman, Eugene R. Mannacio, APL81, 900 North Point Street, San Francisco, Ca 94109.

Richard O. Orgass, Program Chairman APL81
 Department of Computer Science
 Virginia Tech
 P.O. Box 17186
 Washington D.C. 20041
 (703) 471-4600

THE INTERNATIONAL MARKETS POTENTIAL ASSESSMENT SYSTEM

IMPAS

As international communications and distribution facilities improve, many multinational firms (and those contemplating expansion into international markets), are interested in assessing the business opportunities available in other nations. As a first step in quantifying these opportunities, The Planning Economics Group, Boston, (P/E), has used SHARP APL to develop a system for the strategic evaluation of international market potential. The International Markets Potential Assessment System, IMPAS, was first developed by P/E to assist a U.S.-based client company in the evaluation of potential for the expansion of their business activities into international markets. IMPAS is now available for other companies interested in similar analyses.

The basic approach of this strategic planning system is to evaluate the attractiveness of international markets on the basis of three general criteria — market size, market growth, and market quality. To support their analyses, users have access to a data base containing from 40-50 economic, demographic, and social indicators for each of 71 countries. The Sharp APL environment has proved to be ideal for IMPAS. Users are able to build and refine their indexes interactively by themselves and have the capability to experiment directly with the data until they achieve satisfactory results. The ease of use and responsiveness of the Sharp APL environment have contributed substantially to the success of the IMPAS system.

The selection of countries for inclusion in the IMPAS data base involved the use of some broad political and economic criteria. Only those countries are included that would provide suitable environments for multinational companies. The IMPAS indicators were drawn from several published sources including the International Monetary Fund, the World Bank, and the United Nations. Data were selected for timeliness, reliability and consistency across the entire sampling of countries. The broad scope of the information in the data base permits evaluation of a wide variety of potential markets and business enterprises.

The methodology developed by P/E for IMPAS incorporates a preliminary screening process designed to narrow the range of foreign markets under consideration to a manageable number. The degree of precision in this screening process is such that it permits identification of the fifteen or twenty leading prospects for further detailed examination and analysis.

Using IMPAS, objective measures of the desired strategic planning criteria can be selected, weighted, and combined into a composite index of overall market attractiveness. Several APL functions permit on-line analysis of the indicators in the IMPAS data base. The system permits easy construction of composite indexes, high-to-low rankings of the data, sorting by country and region, and report writing.

Additional information on IMPAS may be obtained by contacting **Dr. William Springer, (Director of Strategic Planning Services), The Planning Economics Group, Boston, 300 Unicorn Park, Woburn Massachusetts 01801, telephone (617) 933-8740; or Ashok Kondapi, I.P. Sharp Associates, Boston.**

STARS — A Storage and Retrieval Subsystem

Jane Yates, Calgary

STARS is designed to aid in the building of customized decision support models. If you have ever had a customized model implemented for you, or implemented one yourself, or if you are considering one, STARS will interest you.

STARS was developed by I.P. Sharp in Calgary, where it was first used to implement models for companies in the energy industry. It has evolved over several years from experience in writing project feasibility models, corporate models, and many other financial decision tools. The general applicability of STARS to modelling in any industry was quickly realized.

There are three aspects of the development of a computer-based modelling system:

1. data input, data storage, and data recall
2. calculations
3. reports

In a situation that requires a customized model, the last two aspects are unique. They are the essence of the model itself; an application programmer should be free to focus his attention on them, without consideration of the "house-keeping" details of the first aspect. STARS achieves this through the provision of a "case" framework for the input, editing, storage and retrieval of data.

Now, what exactly constitutes a "case" in STARS? Well, suppose your firm is developing a corporate model. You might want to examine closely what happens to the company under general recessionary economic conditions. This would involve certain assumptions about inflation rates, cost of capital, consumer demand, as well as your particular corporate data. This interrelated set of assumptions and data, capsulized in the input to the model, would comprise a "case". Another case might be comprised of the data and assumptions in a favourable economic climate.

You want an easy-to-learn input procedure, with no redundant specifications; you want a powerful editing mechanism; you want to be able to identify a case, run reports using it, change it and store it for future reference. In order to perform "what-if" analyses, you probably want to store a considerable number of such cases. STARS is a system which provides all of these capabilities.

STARS has already been used in applications in the resource industries such as:

- project feasibility models for oil and gas exploration companies
- a corporate model for a land development company
- a heavy oil project feasibility model for an energy company
- a financial model for a bank
- a mining tax simulation model
- a corporate model for a diversified transportation company
- a cash flow model for a real estate development company
- a budget model for a mining company

STARS has saved time and money. In addition, it has provided a consistent framework for financial analysts who may use more than one modelling system.

If you feel that STARS could help your organization, please ask your local Sharp representative for further information.

NEW INTERNATIONAL PETROLEUM DATA BASE

Lib Gibson, Toronto

The Energy and Natural Resources Support Group has recently put up a new data base containing data from Table 6 of the **International Petroleum Annual**, published by the U.S. Department of Energy. Covering two hundred countries, Table 6 reports Domestic Production, Imports, Exports, Stock Changes, Refinery Input of Crude, Refinery Output (considered equal to Refinery Imports), Imports, Exports, and Final Demand for Refined Petroleum Products, and Bunkers.

IPA6 lends itself well to analysis of broad trends in the petroleum field worldwide. Available annually from 1960, it offers enough history for some long term perspective. For instance, the example (colour insert) shows what might be considered a "petroleum profile" of four countries, highlighting the differences in petroleum supply. The blue section of the plot represents net imports, that is imports minus exports; net exporting countries have negative values here. The red section represents domestic production.

**ENERGY DATA BASE SEMINARS
IN EUROPE**

During May and June, a series of Energy Data Base Seminars are being given in Europe. These seminars have been organized for two purposes: to demonstrate to people who would be interested in energy data bases how they can be used to obtain information about international balances in energy supply and demand; and to stimulate some lively discussion among a group of interested participants as to future directions in this area.

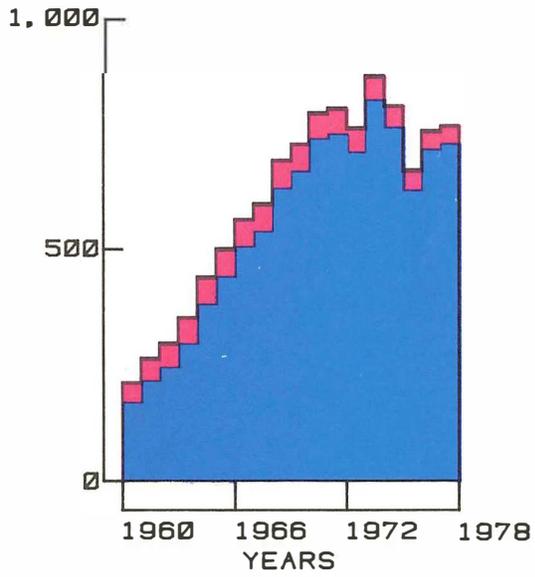
We have invited people from petroleum companies, government agencies, industry organizations, petrochemical companies, consulting firms, the petroleum press, and financial and banking institutions. The seminar will be relatively informal; we'll have a computer terminal and a four-colour plotter on hand for a live demonstration. There is no charge for the seminar, and you are welcome to bring along a colleague when you come. The language of the seminar will be English. Of course, representatives from our local office are fluent in your language.

The schedule is as follows:

London, May 20
Paris, May 22
Hamburg, May 29
Vienna, June 3
The Hague, June 5
Oslo, June 11

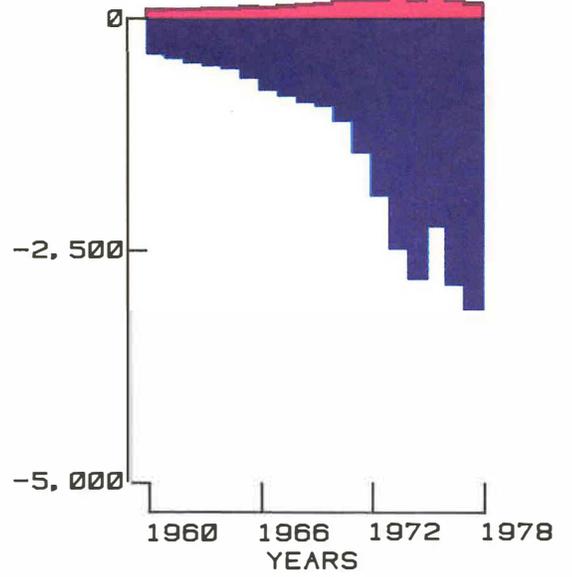
If you or a colleague would like to attend, please get in touch with your local I.P. Sharp representative.

GERMANY



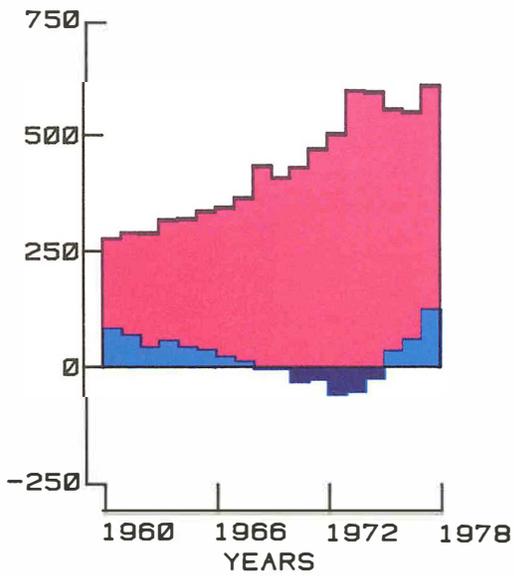
— DOMESTIC SUPPLY
 ■ PRODUCTION
 ■ NET IMPORTS

SAUDI ARABIA



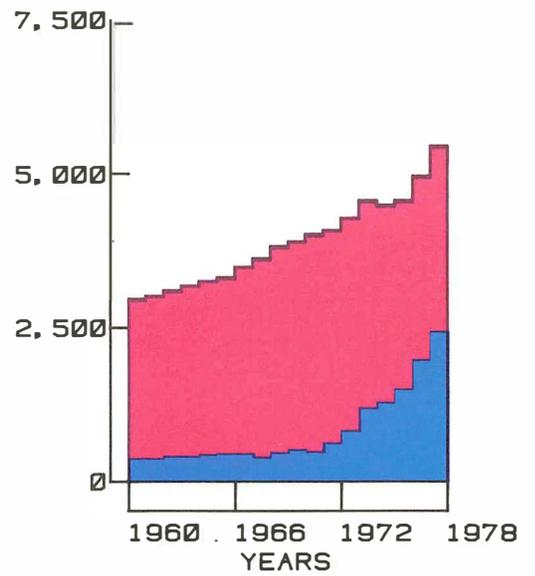
— DOMESTIC SUPPLY
 ■ PRODUCTION
 ■ NET IMPORTS

CANADA



— DOMESTIC SUPPLY
 ■ PRODUCTION
 ■ NET IMPORTS

U. S. A.



— DOMESTIC SUPPLY
 ■ PRODUCTION
 ■ NET IMPORTS

UNITS ARE MILLIONS OF BARRELS

READER'S CORNER

SHARP LITERATURE

Tony de Lucovich, Amsterdam

A new system for storing the titles of literature available from I.P. Sharp Associates is now available in workspace 707 *MANUALS*. At present, data is stored for the following 'categories' of literature:

TECDOC Technical Documentation
BROCHURE Publicity Brochures
FACT Fact Sheets
EXTERNAL External Publications
FUTURE

A 'record' is stored for each publication, made up of all or some of the following 'fields', depending on category:

Title
 Author and publisher - if not IPISA
 Date of publication
 Number of pages
 Cost - see below
 In-stock indicator

The main capabilities of the system are:

- * **List all records for all or selected categories.** For example,
LIST TECDOC, BROCHURE
- * **Print an on-line version of the Publication Order Form.**
- * **Inform you, when you load the workspace, if any records have been added, updated or deleted since you last used the system.**
- * **Inform you if an old printout from this system is still up to date.** - you re-enter the timestamp that is printed at the end of every printout. For example,
80 2 29 UPDATED ALL
- * **Search all or selected categories for a given character string,** for example,
'MAGIC' LIST ALL
- * **Costs on all printouts may be converted to your local currency, using exchange rates built into the system.**

The input and update of the data for the present categories is the responsibility of the Publications Group in Toronto. For more information, please see the *DESCRIBE* in the workspace.

APL PRESS

APL PRESS has moved from the East Coast to the West Coast. Formerly in Pleasantville, N.Y., it is now situated at 220 California Ave., Suite 201, Palo Alto, Ca. 94306. Its new manager is **Gene McDonnell**, who is also chairman of STAPL, an editor of APL Quote Quad, and a member of the APL Development Group at I.P. Sharp Associates.

APL PRESS is primarily a publisher of books which use APL in the exposition of a topic, including many of the textbooks written by Dr. K.E. Iverson. It also publishes a newsletter called APL NEWS.

BESTSELLER!

"Max was a genius with computers. He would sit before the console board and play the machines like a virtuoso. It did not matter what language the computer had been taught, for Max spoke all of them. He talked to digital computers and low-level and high-level language computers. He was at ease with FORTRAN and FORTRAN IV, the giant IBM 370's and the PDP 10's and 11's and ALGOL 68.

He was at home with COBOL, programmed for business, and BASIC, used by the police, and the high speed APL, which conversed solely in charts and graphs. Max talked LISP and APT, and the PL-1. He held conversations in the binary code, and questioned the arithmetic units and the CPU units, and the high-speed printer answered his questions at the rate of eleven hundred lines a minute. The giant computers had spent their lives sucking up information like insatiable pumps, storing it, analyzing it, remembering it, and now they were spewing it out in Max's ear, whispering their secrets to him in their secluded air-conditioned crypts."

Sydney Sheldon, Bloodline

©William Morrow and Company,
 Inc., New York 1978
 Reprinted by permission.

PROJECT TO DEVELOP A STANDARD APL

American National Standards Committee, Computers & Information Processing, ANSC-X3

The American National Standards Committee X3 has approved a proposal to develop a standard for the APL language. The project was proposed by STAPL (Sigplan Technical Committee on APL), an ACM special interest group that also publishes APL Quote Quad, and sponsors APL workshops and APL conferences.

The standard is intended to significantly reduce the need for conversion of APL programs when transferring them between systems and to reduce the retraining of programmers. To meet this goal, the standard must cover most of the features of APL systems that are commonly used in practice without constraining evolution of the language. The newly formed technical committee X3J10, is expected to consider the following topics, among others: the APL character set, the syntax of the language, primitive functions, communication with terminals, APL system functions, APL system commands, APL system variables, scope rules for variables, calls to programs, communication with programs written in other languages, a file system, error messages, and the process of defining a function. The proposal states that formal notation will be used as needed so that the standard will be unambiguous. The formal notation will, however, be supplemented by a more readable informal description that will point out easily overlooked consequences of the definition. The time projected to complete development and approval is three years.

An initial organizational meeting of the technical committee is scheduled for May 28th in Washington DC. Persons wishing to participate in the work who meet the requirements (technical knowledge and willingness and ability to participate) should plan to attend. For more information, contact **Clark Wiedmann**, Convenor, University Computing Center, Graduate Research Center, University of Massachusetts, Amherst, Ma 01003, (413) 545-1500.

A CELEBRATION OF APL

Lael Kirk, Rochester

Syracuse University has recently installed SHARP APL. In celebration of this event they held a one day conference on March 28 for people in the central New York area. The focus of the day was solving practical problems in education, business and industry. Invited speakers included Lib Gibson and Ken Iverson from I.P. Sharp. Lib talked about maintaining and accessing large data bases in APL. She showed many examples of reports and graphs from Sharp's oil data bases. Ken introduced some of his ideas on teaching APL and gave examples of materials he has been using in some experimental courses at Sharp offices.

Other speakers at the conference were Ken Jones from IBM (Programmer productivity through the use of APL), Craig Bell from Sybron (Using APL for corporate decision making) and Professor Gunter Wessel from Syracuse University (How APL helps me handle paperwork for a heavily enrolled course).

The success of the conference was due to the efforts and enthusiasm of Phyllis Kent and Garth Foster at S.U.

MAGIC INTERFACE WITH PRIVATE DATA BASES

Private data bases with four levels of structure can interface with—or be driven by—MAGIC. A minibase allows you to coordinate your private data with public data, using MAGIC as the common language. MAGIC is expanding, as ever, in response to user needs. A new MAGIC manual is currently in production, and should be available shortly. The following session illustrates some of the capabilities of the system. Please contact **David Steinbrook**, I.P. Sharp, Philadelphia for more information.

A <MINIBASE> IS A PRIVATE FILE FOR STORAGE OF TIMESERIES DATA ORGANIZED WITHIN FOUR LEVELS:

MINILISTS 'SAMPLE'

SAMPLE WED 2 APR 1980

PERNMS	COMMODS	ZONES	CATS
365 DAILY	1 DIST	100 ROTTERDAM SPOT	HIGH
52 WEEKLY	2 RESID	200 GULF COAST	LOW
	3 KERO		
	4 DIESEL		

<SAMPLE> IS A MINIBASE WITH TWO PERIODICITIES, FOR EACH OF WHICH THERE ARE FOUR COMMODITIES. FOR EACH COMMODITY, AS MANY AS TWO REGIONS, EACH WITH AS MANY AS TWO CATEGORIES MAY BE STORED. YOU CAN ADD OR DROP ENTRIES AT ANY OF THESE LEVELS.

THE TIMEPOINTS WITHIN EACH COMMODITY MAY VARY:

MINIDATES 'SAMPLE'

SAMPLE WED 2 APR 1980

	365 DAILY	52 WEEKLY
DIST	791112 791203	791005 791123
RESID	791112 791203	791005 791123
KERO	791112 791203	791005 791123
DIESEL	791113 791116	0 0

JUST AS WE SET A TIMEFRAME BEFORE ACCESS, WE SET ONE PRIOR TO UPDATE AS WELL:

CLEAR DAILY DATED AT 12 11 79
'DIESEL' MINIUPDATE 'SAMPLE'

365 DAILY DIESEL FOR MON 12 NOV 1979

RS

?

ENTER HIGH LOW VALUES FOR:

RS

17 16

GC

13

2 ELEMENT(S), PLEASE

GC

14 13

DISPLAY [Y/N]?

Y

	HIGH	LOW
1 RS	17.00	16.00
2 GC	14.00	13.00

OK [Y/N]?

Y

<SAMPLE> UPDATED 365 DAILY FOR DIESEL MON 12 NOV 1979

CHOOSE COMMOD [SPACE,RETURN TO EXIT]

?

COMMODS ARE:

1 DIST 2 RESID 3 KERO 4 DIESEL

CHOOSE COMMOD [SPACE,RETURN TO EXIT]

Technical Supplement-26

SATNS and THE SHARP APL REFERENCE MANUAL

The Sharp APL Technical Notes, as defined in SATN-0, fill the gap between "no documentation" and "good documentation". The Reference Manual has indeed provided good documentation for many of the original SATN titles. The current SATN index below shows the twelve titles that remain in workspace 1 *SATN*.

SATN-0	SATN Introduction (1.1.76)
SATN-2	Control Messages (20.10.79 Rev.4)
SATN-4	N-tasks and B-tasks (1.4.78 Rev.2)
SATN-5	Batch APL (1.2.80 Rev.3)
SATN-8	<i>HSPRINT</i> (1.3.79 Rev.2)
SATN-9	Usage Inquiry System (1.8.78 Rev.1)
SATN-10	<i>SORTREQ</i> (1.6.78 Rev.2)
SATN-19	Fileprint (1.1.77)
SATN-22	APL Workspace Transfer (11.3.80 Rev.3)
SATN-23	Comparison Tolerance (15.7.78 Rev.1)
SATN-28	Terminal Control (11.7.77)
SATN-29	System Time and Timestamps (15.6.78)

BRANCHING IN SHARP APL, PART V

Robert Metzger, Rochester

This series has suggested a number of programming methods which replace the use of the branch arrow. These methods are:

- 1) Use boolean conditional statements
- 2) Use compression
- 3) Use the execute function
- 4) Use APL's built-in loops
- 5) Use subfunctions
- 6) Use recursion

This final segment describes the circumstances in which branching is appropriate. It also suggests means of making effective use of the branch arrow.

Let's review the purpose of the branch arrow. It causes the lines of a function to be executed in an order other than the order they are listed in. There are only two basic ways to change the order of execution.

1. To do a set of statements repeatedly while some condition is true.
2. To do one set of statements instead of another, based upon some condition.

The default order of execution which APL provides is **SEQUENTIAL**. The other two types are called **ITERATIVE** and **ALTERNATIVE**. All patterns of branching fall into either the **ITERATIVE** or **ALTERNATIVE** categories.

When is it appropriate to use **ITERATIVE** structures built with branch arrows? If all of the data to be processed by a function is not in the active workspace at the same time, it is necessary to use branching. The data may not all be in the active workspace because it won't fit. Or, it could be provided in pieces to the function from a source outside the active workspace.

If data won't fit in the active workspace, we normally store it in several file components. Then we loop through the components, processing each one in turn. Data which is not all available at the same time can come to a function from two sources. A person may be entering the data at a terminal. Another program may also provide data in this way. Such a program could be another APL program in a different active workspace. It could also be a program completely outside of APL. It is unusual to write loops in APL if one of these circumstances does not hold.

When is it appropriate to use **ALTERNATIVE** structures built with branch arrows? If an **ALTERNATIVE** sequence can not be conveniently expressed with one of the methods suggested in the previous articles, it is appropriate to use branching. Frequently this occurs when each of the alternatives must be expressed in several APL statements.

It is a provable theorem that any program in any computer language may be constructed using one type of each of the **SEQUENTIAL**, **ITERATIVE**, and **ALTERNATIVE STRUCTURES**. In practice, we use several common variants of these structures to ease our programming effort.

There are four types of **ITERATIVE** structures which are commonly used. Three of these are distinguished by where the controlling condition is placed. This statement can be put before, after, or within the set of statements to be executed. The fourth type is a special case of the first.

The loop with the condition up front is known in several programming languages as a 'WHILE' loop. The concept is expressed thus: **WHILE** condition is true, **DO** execution statements. This can be written in APL as

```
LOOP: →( condition is false )/END
      execution statements
      →LOOP
END:
```

We normally use this type of loop, unless special circumstances arise.

The loop with the condition at the end is known in some programming languages as 'UNTIL' loop. The concept is expressed as: **DO** execution statements **UNTIL** condition is false. This can be written in APL as the following:

```
LOOP: execution statements
      →( condition is true )/LOOP
```

We do not normally use the **UNTIL** loop because it has a significant drawback. The **WHILE** loop can gracefully 'do nothing', whereas the execution statements in the **UNTIL** loop will always be executed at least once. This will cause a program error if it is not possible to execute once. An example of this is a batch APL task which runs every night to process a file of transactions built up during the day. If there are no transactions in the file (perhaps it was a holiday), the **WHILE** loop would complete successfully, but the **UNTIL** loop would terminate with a **FILE INDEX ERROR**.

The loop with the condition in the middle is sometimes called a **CYCLE**. The concept is expressed as: **DO** some statements, **LEAVE** the cycle if a condition is true, otherwise **DO** some more statements and **CYCLE** back to the beginning. This can be written in APL thus:

```
CYCLE: execution statements
      →( condition is true )/END
      more execution statements
      →CYCLE
END:
```

We normally use a cycle when data is being provided in parts to the program from a source outside the active workspace. This most commonly occurs when data is being entered by a person at a terminal. Properly designed interactive dialogues allow a user to end data entry by entering a 'STOP' command.

The user will enter a line of data and the program will determine if the entry is a STOP command. If it is not, the entry is processed, and the cycle is repeated. A similar situation occurs when data is being passed from one processor to another through a shared variable. The receiving processor will access the shared variable and determine if the value is an 'end of file' or 'end of transmission' signal. If it is not, the value is processed, and the cycle is repeated.

The fourth type of loop is the count-controlled loop. Such a loop is executed a specified number of times. The iteration number is stored in a counter variable. Such loops can count up to or down from an initial value to a final value. These loops can be built similarly to either WHILE loops or UNTIL loops. It is preferable to put their condition statement at the front for the same reason that WHILE loops are preferable to UNTIL loops. In this way they can be executed 0 or more times, instead of 1 or more times. A count-controlled loop which counts up to a limit would be set up as follows:

```
COUNTER← (lower limit) - step
LIMIT← upper limit
LOOP:COUNTER←COUNTER+ step
→(LIMIT<COUNTER)/END
  execution statements
→LOOP
END:
```

One feature of this arrangement is incrementing the loop counter at the beginning of the loop. Dr. Iverson recommended this style of incrementing and testing at the head of a loop in **A Programming Language** [p. 6]. We have already explained the advantage of placing the decision at the head of the loop. When a counter is incremented at the end of loop, it is easy for the incrementing step to be ignored if branching inside the loop becomes complicated. Incrementing at the beginning avoids this problem. In order to assure that the incrementing and testing are done together, some people even put them in one statement:

```
LOOP:→(LIMIT<COUNTER←COUNTER+
      step)/END
```

A common use of a count-controlled loop is to process the components in a file serially.

There are three types of ALTERNATIVE structures which are commonly used. These are distinguished by the number of different sets of execution statements which may be chosen.

The standard ALTERNATIVE structure is frequently called the IF-THEN-ELSE. Either one set of statements or the other is done, depending on the value of the condition statement. Such a construct is built in APL as follows:

```
→( condition is false )/ELSE
  THEN: execution statements
→END
  ELSE: other execution statements
END:
```

The IF-THEN structure is a variant of IF-THEN-ELSE in which one of the sets of execution statements is empty. We build such constructs in APL as follows:

```
→( condition is false )/END
  execution statements
END:
```

The structure known as a CASE provides for executing one set of statements chosen from a number of possible sets. CASE constructs can be built with nested IF-THEN-ELSE constructs, but are easier to understand when built with a boolean vector compressed over a vector of labels. An example is given below:

```
CASE:→(1 0 -1=×DATA)/POSITIVE,
      ZERO,NEGATIVE
  POSITIVE: execution statements
→END
  ZERO: execution statements
→END
  NEGATIVE: execution statements
END:
```

You may have noticed that we have followed several conventions in constructing all of the various branching statements in this series. These conventions can help you make your functions easier to understand and easier to change.

WRITE CONDITIONAL BRANCHES IN THE SAME WAY

Always writing conditional branches the same way makes programs easier to read. The reader knows what to expect. It also makes you more productive as an APL programmer. You will write whole statements without thinking about how to say what you need to. Writing in phrases or idioms is the way you write in a natural language. There are a number of ways of writing conditional branches:

- (condition)ρ line
- (condition)+ line
- (condition)/ line
- line ×₁ condition
- line *IF* condition

(See ws 999 *BRANCHING* or Part II of this series for the definition of *IF*)

Choose the method you feel is easiest to read—all of them are good. The CPU usage differences do not become noticeable until you do several thousand branches. If you have a program which executes several thousand branches, you have much bigger things to worry about.

BRANCH TO LINE LABELS

Do not branch to absolute line numbers or $\square LC$ expressions. The reason for this convention is to make programs easier to change. Branch statements which refer to absolute line numbers or $\square LC$ may need changing every time you change other parts of the program. If you insert or delete a line between the line on which the branch statement occurs and the line which is the branch target, the statement becomes invalid. So either change two or more lines every time you want to change one line, or throw away your now invalid programs. Labels are labor saving devices — use them.

CHOOSE LABEL NAMES THE SAME WAY

Once again, adopting a convention makes programs easier to read and quicker to write. There are several approaches to creating label names:

1) the FORTRAN approach. Labels are a letter (typically *L*) followed by increasing integer values: L05, L10, L15, One advantage of this approach is that the labels are short. Another is they can be re-used in other programs (saving symbol table entries). The disadvantage is that they mean nothing.

2) The ALGOL approach. Labels indicate the type of structure they constitute. In order to distinguish multiple occurrences of the same structure in a single program, a digit may be suffixed to labels. The labels for the end of a structure are the letters of the structure name reversed. Some examples:

```
IF - FI
IF - THEN - ELSE
CASE - ESAC
```

The main advantage of this approach is that the labels indicate the control structures in the program. The same labels can also be re-used in other programs. The disadvantage is that you can easily mistype the distinguishing digit.

3) The contextual approach. Labels indicate what is happening on the line labeled. They act as executable comments. The chief advantage is that the purpose of the labelled lines is easier to see. The disadvantages are that the labels are longer, and are not as often re-usable in other programs. Some examples:

```
PRINTΔLOOP, , VERIFY, INVALIDΔDATA
```

KEEP A TEST AND ITS RELATED ACTIONS TOGETHER

Some APL users let conditional branches and the execution statements which are related to them drift apart. The phrase 'spaghetti code' refers to programs in which the executable statements are tied together with a net of intertwining branches. The less we alter APL's sequential execution, the easier our programs are to read. Thus we keep the branch and the statements together so that the program reads from top to bottom as much as possible. The ideal program always has the statements of the THEN part follow the IF branch, and the statements of the ELSE part follow the THEN part.

The most common symptoms of the 'Driftwood Syndrome' (floating branches) is branching around branch statements. The following program fragment is an example:

```
→(~ test )/LABEL1
→LABEL2
```

If the test and action were kept together, this could be rewritten as the following:

```
→( test )/LABEL2
LABEL1:
```

These suggestions on how to branch end this series of articles. We have presented a number of alternatives to branching. It is fair to say that when you are writing APL programs, it is frequently helpful to stifle the urge to branch. The search for an alternative may provide you with a more elegant, efficient, or effective program. When you must branch, abiding by a few simple conventions will make your programs more reliable and understandable.

<SAMPLE> UPDATED 365 DAILY FOR DIESEL MON 12 NOV 1979

UPDATE MAY BE BY REGION AS WELL AS BY COMMODITY. UPDATE FOR A TIMEPOINT ALREADY REPRESENTED CALLS FOR REVISION:

FRAME

DAILY DATED AT 12 11 79

'DIST' MINIUPDATE 'SAMPLE'

◇ STANDARD ZONES? [Y/N]

Y

365 DAILY DIST FOR MON 12 NOV 1979

◇ DISPLAY [Y/N]?

Y

	HIGH	LOW
1 RS	15.00	14.00

◇ OK [Y/N]?

N

◇ PICK A ROW

1

◇ ENTER HIGH LOW FOR RS

15.5 14.5

◇ DISPLAY [Y/N]?

N

◇ OK [Y/N]?

Y

<SAMPLE> UPDATED 365 DAILY FOR DIST MON 12 NOV 1979

◇ CHOOSE COMMOD [SPACE,RETURN TO EXIT]

<SAMPLE> UPDATED 365 DAILY FOR DIST MON 12 NOV 1979

CLEAR ◇ (15) DAILY DATED 12 11 79 TO 16 11 79
TIMESERIES

PUT 'DIST,DIESEL' SAMPLE 'RS' DISPLAY ABOVE

FILE: SAMPLE

	12NOV79	13NOV79	14NOV79	15NOV79	16NOV79
RSH DIST	15.50	13.00	14.00	13.00	14.00
RSL DIST	14.50	13.00	14.00	15.00	16.00
RSH DIESEL	17.00	0.13	0.14	0.13	0.14
RSL DIESEL	16.00	0.14	0.15	0.14	0.15

TO MAINTAIN WEEKLY DATA, WE CONSOLIDATE THE CHANGES.

AVERAGES

5 WEEKLY DATED AT 16 11 79

'DIST,DIESEL' MINICONSOL 'SAMPLE'

<SAMPLE> CONSOLIDATED 52 WEEKLY FOR DIST FRI 16 NOV 1979

<SAMPLE> CONSOLIDATED 52 WEEKLY FOR DIESEL FRI 16 NOV 1979

OTHER FEATURES OF THE MINIBASE PACKAGE LET YOU SPLICE DATA FROM WORKSPACE TO MINIBASE, CARRY-FORWARD DATA FROM THE PREVIOUS ENTRY, GROUP COMMODITIES OR REGIONS, EXPAND DATE-OF-CHANGE DATA, AND OTHERS.

COMMENT

From Phil Chastney —

Part of the appeal of childrens' rhymes is their constancy. One example I heard for the first time recently seemed to be the better part of one hundred years old, judging from internal local references. Looking so far back into the dawn of history was obviously quite a strain on the comprehension of the little one who told me the rhyme: her next contribution was nothing like so old, however. It went —

I'm a little robot, short and square,
Got no toes, got no hair.

When you want the answer to your sums,
Press a button, out it comes.

It is no doubt interesting to see new technology penetrating the collective consciousness, but it surely deserves better treatment? The rate of one major misconception per line must be something of a record.

While one might object to

- the confusion of computers with robots,
- the anthropoidal image of computers and robots, and
- the assumption that computers are primarily numeric in application,

it is that last line that really rankles.

Doesn't the programmer deserve a mention?

SHARP PEOPLE

PALO ALTO

Paul Jackson has joined the APL Development Group. Paul began working in APL in 1969 at Trinity University in San Antonio, Texas, where he was a computer science instructor. In 1972 he moved to Dallas, Texas, and joined Dallas County Community College District, a local County Government Agency which controls the administration and support services for seven community colleges in Dallas County. He established a large educational use of APL and then was made Manager of Systems. For the last three years he has been managing a 210-terminal network which is used by the College District to meet all of its educating and administrative needs in APL.

Paul Jackson



OSLO

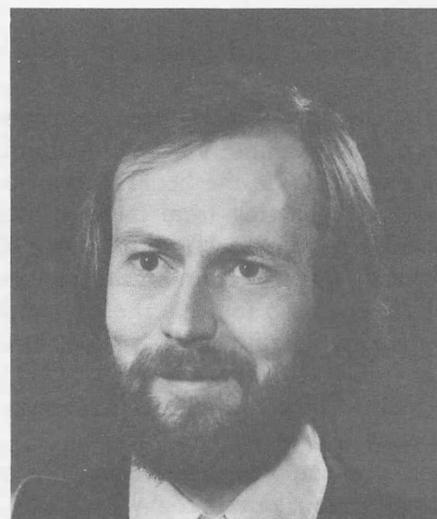
SØNNER AV NORGE

Scott Remborg has now come full circle. He graduated from the Oslo Institute of Business Administration (Bedriftsøkonomisk Institutt) with a B.Comm., majoring in Management Science and in his own words, 'spent too much of my time programming'.

Encouraged by a Canadian wife, he went on to take a Masters degree in Business Administration at the University of Alberta in Edmonton, where he majored in Finance. Upon graduating he joined the Calgary office, where he has primarily been involved with building financial models for the energy industry.

Finally, in March, Scott returned to his home town, and joins **Carol Aitchison** in our office there.

Scott Remborg





Sandra Doubleday joined **Mike Powell** in the Victoria office in April. She has a BA in Mathematics from Queen's University and an MA in Business from Sheffield. She has been using APL in business and government, most recently with the B.C. Central Statistics Bureau, and is particularly interested in the expanding use of data bases.

GOOD NEWS FOR PROJECT MANAGERS

Project Network Analysis Packages

We now offer two different approaches to Project Network Analysis (sometimes called PERT or CPM). Each of the two systems has been designed to meet the needs of a different kind of user. It is also possible to transfer data between the two systems quickly and easily, and to use them with MABRA, Sharp's powerful and easy to use data management system.

There are two workspaces, *SNAP* and *FASTNET*. *SNAP* offers a very comprehensive range of facilities which are particularly suited to multi-project scheduling. *FASTNET* is aimed primarily at users who want a quick, cost effective means of processing small networks, or managing single projects.

Both packages have very versatile and flexible data input, update, calculation and reporting facilities. Users can add their own data fields in addition to those required for the critical path calculations. Reports can be presented in tabular or bar chart form and there is provision for resource histogram and S-curve generation. Both packages have access to *DRAWNET*, a facility for drawing a network diagram from the data held on file.

SNAP has facilities for resource aggregation, hammock activities, and milestone reports, but full resource scheduling is not available as yet. *FASTNET* has a resource scheduling facility which we plan to make available to *SNAP* users in the not too distant future. Both systems are designed to make it easy for users to extend their capabilities to suit their own requirements or to enable customized systems to be quickly and easily produced by I.P. Sharp consultants.

Where the two packages perform the same or similar tasks we have endeavoured to use the same or similar commands. A user of one system should therefore have little difficulty in understanding how to use the other.

These packages were both produced in the U.K.—*SNAP* was designed and written by Ken Chakahata and has been in use in pre-release version for some time in library 544. *FASTNET* was produced more recently by Nick Telfer, who also wrote the *DRAWNET* network drawing facility. *SNAP* is expected to be released in the next month or so when the manual is completed. It will then be available in library 48. *FASTNET* and *DRAWNET* can be expected to follow some time later in the year.

For information about these packages please contact Nick Telfer in the London office.

ON LINE APPLICATIONS LIBRARY

Applications Library programs, in workspaces 1 through 499, are constantly being updated. Please make a note of the following change:

CHANGED: 55 *RETRIEVE* The state-setting function Δ *ERRORTYPE* takes a new argument. It allows the use of \square *SIGNAL* and \square *TRAP* for communication of error conditions between *RETRIEVE* and the application system in which it is embedded. More detailed information is in Δ *DESCRIBE* in the workspace.

APL80 Conference
Leewenhorst Congres Centre, near Leyden,
The Netherlands.

This International conference on APL is organized by the Leyden University Computing Centre (CRI), under the Dutch Computer Society (NGI) and the European Cooperation in Informatics (ECI). Central address: APL80 Registration, CRI, Postboks 9512, 2300 RA Leyden, The Netherlands.

The provisional program is summarized below. (Only the first authors are listed — for titles and any further information please contact APL80, CRI at the above address):

Monday June 24th:

Invited speaker, **Dr.K.E.Iverson**, 'Notation as a tool of thought'

Use of Language—G.A.Blaauw; F.Cesarini; W.Rutten

Methodology—G.R.Mayforth; D.G.I.Mathieson; K.S.Andersen

Simulation—A.Wellbrock; Th.v.d.Bruele; W.K.Giloi

APL-systems—M.Udo; J.D.Sybalsky; G.H.Foster

Programming Techniques—G.R.Mayforth; B.R.Martin; S.B.Jaffe

Invited speaker, **A.Kay**, 'General Programming, APL and Small Talk'

Wednesday June 25:

Invited speaker, **P.S.Abrams**, 'Large Applications in APL'

Design of language—A.M.Marcum; G.Martin; E.McDonnell; J.deKerf;

N.D.Thomson; R.C.Metzger

Applications—C.Witteveen; R.Pinanzi; D.Owen; G.Elshoud; H.Koppelaar;

J.T.Hardin

Thursday June 26:

Data Base applications—A.Vaissiere; F.Schmidt; E.van Ingen; E.Mahood

Education—A.Spoelstra; J.Moonen; L.Alvord; H.A.Peelle

Data Analysis—P.J.Lewi; M.Dell'Oca; R.Erbe

Special Topics—W.K.Giloi; Th.W.Cook; R.H.Swain

Invited speaker, **R.Tisseraad**, 'International Standard of APL'

Friday June 27th:

Invited speaker, **A.J. Perlis**, 'Construction of APL Processors'

COURSE SCHEDULE

SEMINARS and SPECIAL COURSES

Aberdeen

Appreciation of APL, June 19

Boston

Report Formatting, May 12
Event Trapping, May 27

Coventry (half-day seminars)

Formatting with SHARP APL, July 12
Event Trapping, May 10
N- and B-tasks, June 14
Plotting, August 9

Gloucester

Appreciation of APL, May 12

London

Advanced APL, June 23
APL Review, May 27, June 9
Appreciation of APL, May 8, June 13
Use a Terminal, June 5
Introduction to SNAP, May 1
Advanced SNAP, July 3

Montreal, (English and French)

AIDS, May 20
Planning System Design, May 6
Shared Variables, May 8

Ottawa

Batch (Non-Terminal) Tasks, August 21
Box-Jenkins, June 19, October 23
Highspeed Printing, July 24, November 20
Input Validation (Idiot Proofing), November 19
MABRA, November 18
MAGIC, June 17, October 21
Plotting in SHARP APL, August 19
Regression Analysis, July 22

Rochester

Advanced Features Workshops, (4 half-days):
Package Variables, May 8, August 7
Batch APL, May 15, August 14
Event Trapping, May 22, August 21
Shared Variables, May 29, August 28
Advanced Methods Workshops (full days):
Interactive Dialogue and Report Formatting, Jul.10, Oct.2
Data Structures and File Design, Jul.17, Oct.9
Program Development, Debugging and Testing, Jul.24, Oct.16
Efficient APL Programming, Jul.31, Oct.23

Toronto

Advanced APL Coding Techniques
for Actuarial Applications, May 1
APL for Managers, May 19, June 30
Box-Jenkins, May 8
Forecasting Methods, May 9
MAGIC for Time Series Analysis, June 6
Regression Analysis, June 17
Report Formatting, June 20
Saving Money with N- and B-tasks, June 18
SUPERPLOT & MAGIC, June 16
Text Editing, May 15-16
The Sharp System:
A Practical Approach, May 8, June 5

Warrington

Appreciation of APL, May 30

INTRODUCTION TO APL

Aberdeen
(3 days)
May 21-23

Atlanta
June 16-20
September 15-29
December 8-12

Coventry
(3 day)
May 20-22
July 22-24

London
(3 day)
May 13-15
June 16-18

Gloucester
(3 day)

Montreal
(English)
May 13-15

Ottawa
(non-programmers)
June 3-8
August 5-7
October 7-8
(programmers)
May 6-8
July 8-10
September 2-4

Paris
(3 jours, francais)
Juin 11-13
Septembre 3-5

Rochester
(6 days)
May 12-14/19-21
June 16-18/23-25
July 21-23/27-29
August 11-13/18-20

Toronto
(3 day)
May 12-14
May 27-29
June 9-11
June 23-25

Warrington
(3 day)
June 25-27

Zurich
(German)
May 12-14 June 2-4 June 9-11

INTERMEDIATE APL

Atlanta, May 7-8

Edmonton, May 27, 28, 29

London, May 28, June 10

Montreal, (English or French), May 27-29

Ottawa, May 20-21, September 16-17

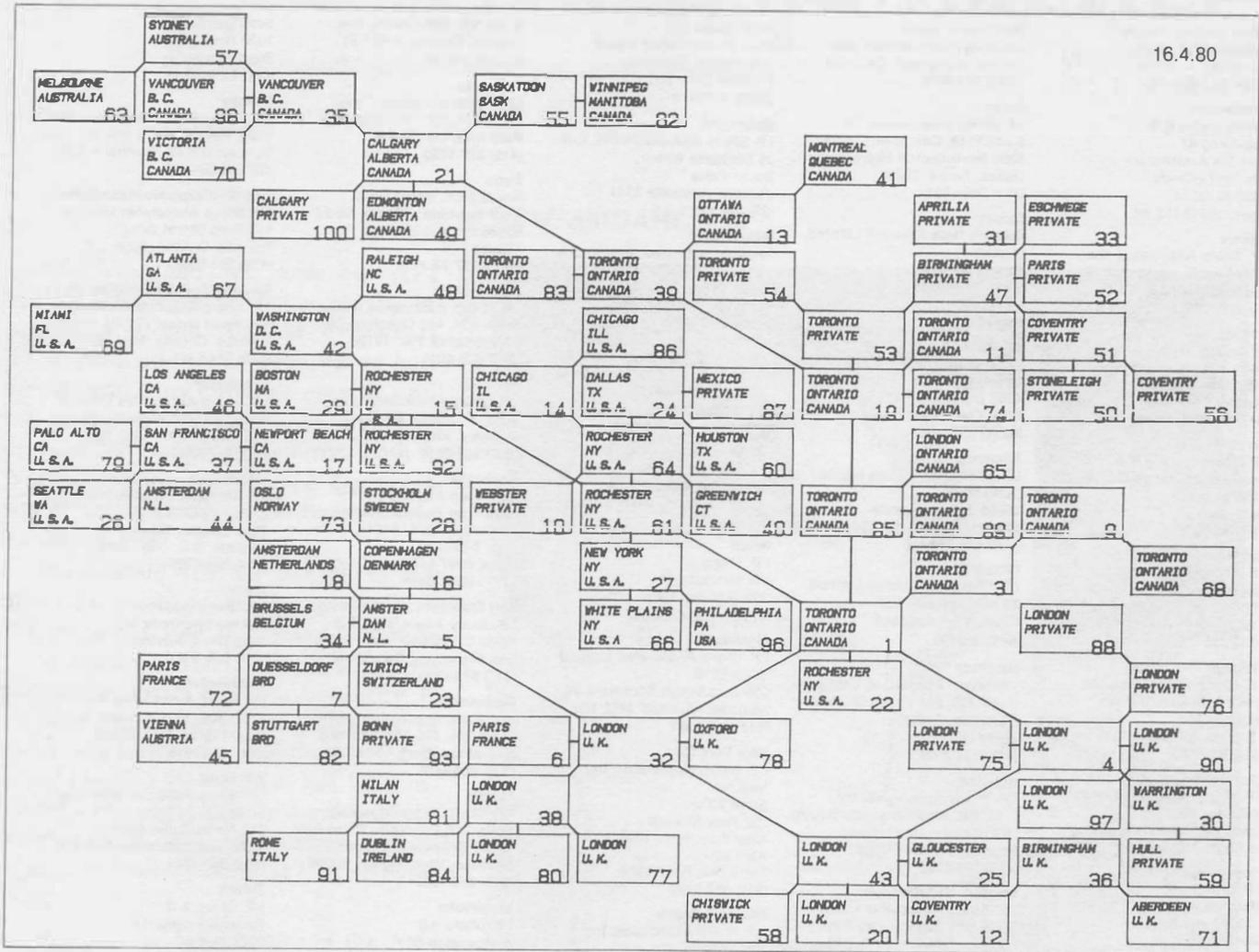
Rochester, (4 days) June 19,20,26,27 and September 4,5,11,12

Toronto, June 2-4

Zurich, (German), May 27-29, June 16-18

Network Topology

16.4.80



UPDATE

Label number: _____

- Please amend my mailing address as indicated.
- Add to your mailing list the following name(s).
- Send me a SHARP APL publications order form.
- Please add my name to the Energy Newsletter list.
- Please add my name to the Aviation Newsletter list.

Name: _____

Title: _____

Co.: _____

Address: _____

Telephone: _____



International Branch Offices

Aberdeen

I.P. Sharp Associates Limited
5 Bon Accord Crescent
Aberdeen AB 12DH
Scotland
(0224) 25298

Amsterdam

Intersystems B.V.
Kabelweg 47
1014 BA Amsterdam
The Netherlands
(020) 86 80 11
Telex: 18795 ITS NL

Atlanta

I.P. Sharp Associates, Inc.
5000 Snapfinger Woods Dr.
Decatur, Georgia 30035
(404) 987-2301

Boston

I.P. Sharp Associates, Inc.
Suite 415
148 State Street
Boston, Massachusetts 02109
(617) 523-2506

Brussels

I.P. Sharp Europe S.A.
Avenue General de Gaulle, 39
1050 Bruxelles
Belgique
(02) 649 99 77

Calgary

I.P. Sharp Associates Limited
Suite 2660, Scotia Centre
700-2nd Street S.W.
Calgary, Alberta T2P 2W2
(403) 265-7730

Chicago

I.P. Sharp Associates, Inc.
2 North Riverside Plaza
Suite 1736
Chicago, Illinois 60606
(312) 648-1730

Cleveland

I.P. Sharp Associates, Inc.
(216) 431-6861
(local call, switched through
to Rochester office.)

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, England CV1 1EA
(0203) 21486/7

Dallas

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

Dublin

Gamma Data Systems Limited
(Agent)
Dollard House
Wellington Quay
Dublin 2, Ireland
(01) 711 877

Düsseldorf

I.P. Sharp GmbH
Leostrasse 62A
4000 Düsseldorf 11
West Germany
(0211) 57 50 16

Edmonton

I.P. Sharp Associates Limited
Suite 505
10065 Jasper Avenue
Edmonton, Alberta T5J 3B1
(403) 428-6744

Gloucester

I.P. Sharp Associates Limited
29 Northgate Street
Gloucester, England
(0452) 28106

Hamilton

I.P. Sharp Associates Limited
Suite 308, 237 Locke St. South
Hamilton, Ontario
L8P 4T4
(416) 527-3801

Houston

I.P. Sharp Associates, Inc.
Suite 925, 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) 434-2426

London, England

I.P. Sharp Associates Limited
132 Buckingham Palace Road
London SW1W 9SA
England
(01) 730-0361

Manchester

I.P. Sharp Associates Limited
Paul House
89-91 Buttermarket Street
Warrington, Cheshire
England WA1 2NL
(0925) 50413/4

Melbourne

I.P. Sharp Associates Pty. Ltd.
36 Elizabeth Street
South Yarra
Victoria, Australia 3141
(03) 244-417

Mexico City

Teleinformatica de Mexico S.A.
(Agent)
Mail to:
Arenal N 40, Chimalistac
Mexico 20 D.F., Mexico
Shipments to:
A/C Despachos del Centro S.A.
Aeropuerto Internacional de la
Ciudad de Mexico
(905) 550-8033

Miami

I.P. Sharp Associates, Inc.
Suite D, Kennedy Building
14560 N.W. 60th Avenue
Miami Lakes, Florida 33014
(305) 556-0577

Milan

I.P. Sharp Srl
Via Eustachi 11
20129 Milan, Italy
(2) 271 6541/221 612

Montreal

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

New York City

I.P. Sharp Associates, Inc.
Mail to:
Suite 2004
200 Park Avenue
New York, N.Y. 10017
Also at:
Suite 242, East. Mezz.
(212) 557-1200

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
4. Etasje
Kronprinsesse Märthas Plass 1
Mail to:
Postboks 1470, Vika
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

Société I.P. Sharp SARL
Tour Neptune - Cedex No.20
92086 Paris-la-defense
France
(1) 773 57 77

Philadelphia

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

Phoenix

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

Rochester

I.P. Sharp Associates, Inc.
1200 First Federal Plaza
Rochester, N.Y. 14614
(716) 546-7270
Telex: 0097 8473
0097 8474

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 208, 135 21st Street E.
Saskatoon, Sask. S7K 0B4
(306) 664-4480

Seattle

I.P. Sharp Associates, Inc.
Suite 217, Executive Plaza East
12835 Bellevue-Redmond Road
Bellevue, Washington 98005
(206) 453-1661

Stockholm

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

Stuttgart/Boeblingen

I.P. Sharp GmbH
Schafgasse 3
7030 Boeblingen
West Germany
(070 31) 2 30 14

Sydney

I.P. Sharp Associates Pty. Ltd.
Suite 1342, 175 Pitt Street
Sydney, N.S.W., Australia 2000
(02) 232-5914

Toronto (Corporate Headquarters)

Tour Neptune - Cedex No.20
145 King Street West
Toronto, Ontario M5H 1J8
(416) 364-5361

Toronto (Special Systems Div.)

I.P. Sharp Associates Limited
156 Front Street W., 5th Floor
Toronto, Ontario M5J 1G6
(416) 364-5361

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
Rechte Wienzeile 5/3
1040 Wien, Austria
(222) 576 571

Washington

I.P. Sharp Associates, Inc.
Suite 400, 1835 K Street N.W.
Washington, D.C. 20006
(202) 293-2915

Winnipeg

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

Zurich

I.P. Sharp A.G.
Badenerstrasse 141
8004 Zurich
Switzerland
(1) 241 52 42

SHARP APL Communications Network: Local Access Cities

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

Local dial access is available in all locations listed above. The SHARP APL Communications Network also provides local dial access in:

- Ann Arbor • Birmingham • Buffalo • Des Moines • Des Plaines • Detroit • Ft. Lauderdale • Greenwich (Ct) • Halifax
- Hartford • Hull • Kitchener • Laurel • Liverpool • Los Angeles • Lyndhurst • Minneapolis • Oxford • Raleigh • Red Deer
- Regina • Rome • Santa Ana • Syracuse • Towanda • Ukiah • Warrington • White Plains (NY)

In the United States the SHARP APL Network is interconnected with the networks of TYMNET and TELENET to provide local dial access in more than 250 other cities.