

# the I.P. Sharp newsletter

JULY/AUGUST 1978  
Volume 6 Number 4

## AN APL USERS MEETING

Hotel Toronto  
Toronto, Canada

September  
18, 19, 20.  
1978

Unlike most APL conferences, there was no call for papers - instead, we invited speakers who we knew had something relevant to say.

Parallel sessions related to specific industries and specific application areas will run for two days. The industries covered include **Banking, Insurance, Resources** and **Finance**, and the application areas will include **Statistics, Graphics, Logistics** and **Financial Planning**.

The first day will be devoted to the presentation and discussion of proposed system enhancements. This will include some new communications facilities designed to make life easier for the APL user, and some short-term and long-term ideas about extensions to the language itself. We expect there will be substantial participation from the audience.

One session, to be chaired by Dick Lathwell, will deal with the problems of migration from an APL service to an in-house time-sharing system. Speakers with the actual experience of making this transition successfully will be happy to display their scars and offer advice.

Two evening tutorial sessions are being staged - "APL System Design" and "Data Base Design in APL".

The final half-day will be a long session dealing with APL standards for documentation and programming. Chaired by Paul Berry, the session will present five papers on various aspects of standards and programming style. The speakers will be Donald McIntyre from Pomona College, Ken Iverson from IBM, Luis Contreras from Southern Methodist University, Alan Harrison from British Nuclear Fuels and Gregory Koutsouros from Xerox. It promises to be an interesting afternoon.

### REGISTER NOW!

All attendees must be **pre-registered**. For those of you coming from out of town: **hotel reservations have to be confirmed by August 1st.**

### IN THIS ISSUE

|  |    |
|--|----|
| APL USERS MEETING.....   | 1  |
| <b>The program:</b> centre insert (in place of the Technical Supplement) |    |
| MAGIC enhancement.....   | 2  |
| Official Airline Guide.....  | 3  |
| MAGIC matrix access.....   | 4  |
| Computer Chess Tournament....  | 4  |
| Batch APL.....   | 5  |
| Session Variables.....   | 6  |
| Roundabout model.....  | 7  |
| AIDS enhancement.....  | 8  |
| Graphics summary.....  | 8  |
| Flow control on ASCII terminals.   | 9  |
| On-line reference card.....  | 9  |
| Sharp in German.....   | 10 |
| Course Schedule .....  | 10 |

## MAGIC ENHANCEMENT: NON-APL OUTPUT

Dave Keith, Toronto

A new feature for MAGIC users has just been released. The primary purpose of the non-APL output option is to allow the use of MAGIC on terminals with non-APL print wheels. One can, for example, use MAGIC on terminals such as the Terminet 300 without having the output look strange. The MAGIC functions *PLOT*, *DISPLAY*, and *TABLE* now may have output converted from lower case to upper case, and special characters such as | that are used in *PLOT* are also converted, say to !. The option setting required is *NOAPL*. The inverse, which has always been the default, is *APL*. It is important to note that the *NOAPL* setting applies **only** to output produced via *PLOT*, *DISPLAY*, or *TABLE*.

Example (from the aviation data base):

```
)LOAD 702 MAGIC .
QUARTERLY,DATED 1 77 TO 4 77
TITLE 'ENPLANED PASSENGERS'
LABEL NAME TRUNKS
HIGHLIGHT
NOAPL
DISPLAY T1,TRUNKS,K110
```

## ENPLANED PASSENGERS

|                           | 1ST/77    | 2ND/77    | 3RD/77    | 4TH/77    | YEAR/77    |
|---------------------------|-----------|-----------|-----------|-----------|------------|
| AAS AMERICAN AIRLINES     | 5,306,087 | 6,077,939 | 6,762,987 | 6,147,473 | 24,294,486 |
| BNS BRANIFF INTERNATIONAL | 2,350,105 | 2,437,086 | 2,466,044 | 2,561,051 | 9,814,286  |
| COS CONTINENTAL AIR LINES | 1,766,849 | 2,004,339 | 2,327,862 | 2,149,260 | 8,248,310  |
| DLS DELTA AIR LINES       | 7,211,009 | 7,577,383 | 7,844,674 | 7,919,927 | 30,552,993 |
| EAS EASTERN AIR LINES     | 7,914,248 | 7,840,861 | 7,830,704 | 7,996,829 | 31,582,642 |
| NAS NATIONAL AIRLINES     | 1,622,142 | 1,577,682 | 1,529,788 | 1,495,958 | 6,225,570  |
| NWS NORTHWEST AIRLINES    | 2,409,042 | 2,556,031 | 2,756,164 | 2,633,571 | 10,354,803 |
| PAS PAN AM WORLD AIRWAYS  | 1,636,656 | 1,939,985 | 2,213,056 | 1,858,298 | 7,647,995  |
| TWS TRANS WORLD AIRLINES  | 3,717,825 | 4,534,461 | 5,117,615 | 4,406,740 | 17,776,641 |
| UAS UNITED AIR LINES      | 7,642,413 | 8,600,264 | 9,643,551 | 8,498,085 | 34,384,313 |
| WAS WESTERN AIRLINES      | 2,245,861 | 2,269,049 | 2,605,462 | 2,274,851 | 9,395,223  |

In an effort to keep MAGIC users aware of new features, the workspace 702 *MAGICNEWS* has been created. By loading this workspace and entering *NEWS* a six page description of those features not described in the current "MAGIC for Time Series Analysis" manual is available. A new manual describing all of MAGIC and its application to the several public and private data bases on the SHARP APL system will be available in a few months.

More on page 4.



## OFFICIAL AIRLINE GUIDE

Bob Dabrowski, Toronto

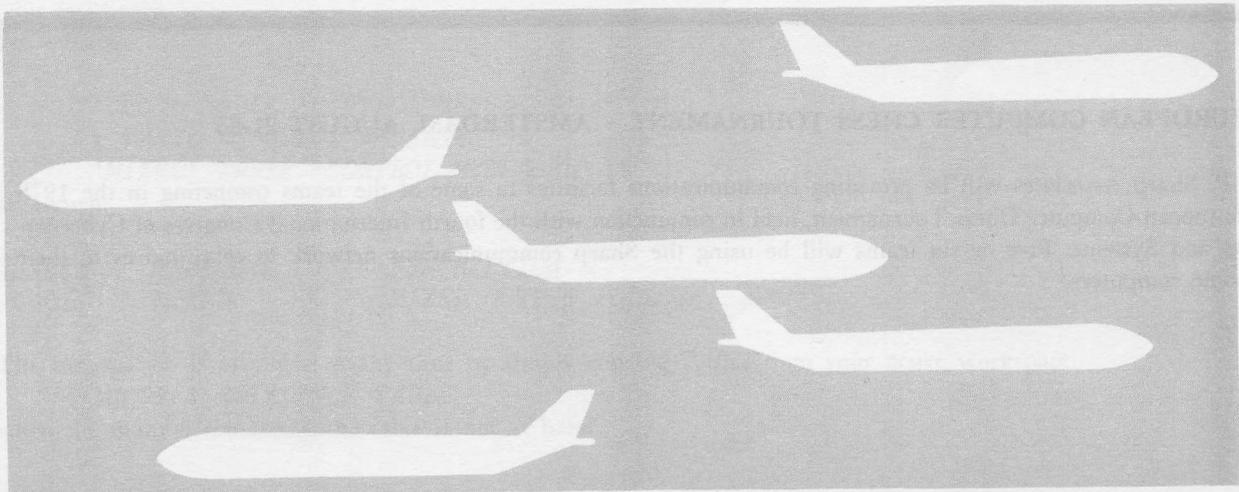
The Official Airline Guide is a publication of The Reuben H. Donnelley Corporation that contains information on all scheduled flights everywhere in the world. It is used as a standard reference throughout the airline industry. Most of you are familiar with it as a means of booking flights; travel agents use it to determine itineraries. There are about 600 airlines that fly almost 150,000 separate flights each month to more than 6000 cities. The data of this publication, the OAG Data Base, is being added to the extensive aviation data bases already available.

As a first step, a summary of monthly schedules will be developed. This summary will contain basic information on all direct flights operating everywhere in the world. Examples of data to be found are class of traffic, operating days, meals served, arrival, departure and block times. In addition, the Civil Aeronautics Board QSI, Quality of Service Index, will be included for each flight. The QSI is a measure of the service provided on any flight and takes into account the equipment type number of intermediate stops and the frequency of service during the month. The QSI will be calculated historically so that broad comparisons of service growth on a specific market (for example) will be quite easy to make. Previously such comparisons were difficult because they involved a great deal of time-consuming manual research.

One feature of the OAG data base that should prove of interest is the cross-reference ability. Users will be able to interrogate the data base to answer complicated questions on many flights. Questions such as "What are the routes of all 747SP flights through Denver, Colorado?" can be answered with a single access to the cross-reference function.

At first there will be no historical data and the data base will start at the middle of the current year. We will add to this later. The ability to add data is restricted somewhat by disc space available but we will extend the data base to contain at least two or three years of data. Future plans also include the development of functions to examine the connection activity for flights between any two points, and a set of functions to manipulate carrier schedules on a day to day basis.

The data base and the information contained in it are the property of The R.H. Donnelley Corp. of Chicago and all users are required to pay a subscription fee for use of the data.



**MAGIC ENHANCEMENTS: MATRIX ACCESS TO CANSIM**

Clive Edwards, Toronto

Access to CANSIM data is now available by matrix number as well as by individual series number. Type *MAT* followed by the matrix number as the right argument to the *CANSIM* function.

**Example:**     `DISPLAY CANSIM 'MAT7140'`

To avoid retrieving too many series when using this option you may also request a count of the number of series retrieved. The use of *TO* is also permitted when using the *MAT* option, and individual series may be intermixed with *MAT*.

**Example:**     `R←CANSIM '?MAT1 TO MAT3'`

This would result in a message: *NN SERIES FOUND* (where *NN* is the number of series), and *R* would contain the actual series numbers in a form suitable for use as the right argument to the *CANSIM* function if you then wish to display the series.

**Example:**     `DISPLAY CANSIM R`

*AUTOTITLE* (*NOAUTOTITLE*): this new function will automatically retrieve the matrix description from the CANSIM base when you are using the *MAT* option for a *DISPLAY* or *TABLE*. If you are accessing more than one matrix, the title will be the description of the last matrix accessed. Please note that at this time not all matrix descriptions are available in the file - these will be added as each of the matrices is updated.

**AUTOLABEL CHANGES**

A slight change has been made to the form of the labels when *AUTOLABEL* is used. *AUTOLABEL* will now cause the series number as well as the description to be displayed. This is particularly useful when using the *MAT* option to access CANSIM data, or when a number of series specified in the right argument are not available.

The *AUTOLABEL* option has also been extended for use with the USCPI data base, as well as the banks data bases. Although series descriptions are not available for USCPI data, use of *AUTOLABEL* will cause the city code and series number to be displayed as the label. When using *AUTOLABEL* with *MBANK*, *QBANK* and *YBANK*, and many banks are specified but only one fact is displayed, the labels displayed will be the bank names instead of the name of the data item.

**EUROPEAN COMPUTER CHESS TOURNAMENT - AMSTERDAM, AUGUST 21-25**

I.P. Sharp Associates will be providing communications facilities to some of the teams competing in the 1978 European Computer Chess Tournament, held in conjunction with the fourth International Congress of Cybernetics and Systems. Five or six teams will be using the Sharp communications network to relay moves to their home computers.

# Batch APL

## SHARP TAKEN TO TASK FOR LOW CPU RATES

Doug Forkes, Toronto

If you're not making use of N- and B-tasks, SHARP APL has a nice surprise in store for you. CPU rates for B-tasks are 40% lower than T-tasks. N-tasks are nearly 30% cheaper than T-tasks.

Of course an N- or B-task can't print a report. The report must be dumped to file and later printed by a T-task or highspeed printed. At a terminal a typical report costs about \$1.00 per page. A highspeed printout costs \$.10 per page.

**What is a typical B-task application?** Consider the following simple (and hypothetical) system.

A user has a master file available on-line for both inquiry and update. To facilitate inquiries the file is sorted, and the inquiry functions use a binary search technique. File updates go on concurrent with inquiries. Obviously it is inefficient to re-sort the file every time an update is made; restructuring the file while inquiries are being made is also going to upset the binary search.

Therefore a sequential holding file is built from the update transactions. Inquiries are made by searching all of the holding file prior to making a binary search of the master. Each evening, a B-task is dispatched to perform the following operations:

- a) Catenate the holding file onto the master file
- b) Submit a sort request to produce a new sorted master file
- c) Submit a B-task request to be dispatched after the sort is complete.

The second B-task performs the following operations:

- d) Erases the original master
- e) Renames the sorted master to be the new master
- f) Submits a B-task for tomorrow's master update.

A new holding file is created each day. The previous one is not erased. All holding files from the previous month are erased after the first Friday of the next month. This guarantees that SHARP APL backup procedures will automatically provide complete system backup.

**N-tasks** can be used to fragment the processing of a B-task to reduce the risk of failure. For example a B-task may wish to invert a matrix, but not wish to be interrupted by a possible unpredictable *DOMAIN ERROR* resulting from a singular matrix. A solution is to relegate the operation to an N-task that will pass back the result in a file component. If the N-task disappears from `□RUNS`, the matrix was singular. The imposed delay of `□RUN` makes this procedure impractical on a large scale, of course.

A new manual, "Batch Tasks in SHARP APL", is available from your local Sharp representative.

**CHANGE IN SESSION VARIABLES:  $\square SP$  and  $\square HT$ , but not  $\square PW$**

Paul Berry, Palo Alto

During March 1978 we added two new system variables to SHARP APL,  $\square SP$  and  $\square HT$ . They became the first members of a new class of variables, referred to as **session variables**.

The distinctive thing about a session variable is that its value is able to last throughout a session. In particular, it survives the loading of a new workspace, or the command `)CLEAR`. This makes it possible for one active workspace to pass information to its successor.  $\square SP$  was provided for just that purpose.

A session variable is a variable like any other. You can assign a value to it, you can localize it, you can save it in a workspace. The one thing that is different is this: every time a new active workspace is created (whether at sign-on, or following a clear or load) the very first thing the system does (even before executing the latent expression) is to respecify the visible (most local) value of each session variable. The most local value of the session variable in your previous active workspace is copied forward to become the most local value in the new active workspace. At sign-on, of course, there is no preceding active workspace, and so each session variable (like other system variables) is set to a standard default value for a T-task, and at  $\square RUN$  time for an N- or B-task.

This behaviour of session variables has an important consequence. While a saved workspace may contain a session variable, you never get to see its most local value when you load the workspace because (before anything else happens in the workspace), the value that the session variable previously had in your active workspace is copied forward, and replaces whatever value was in the workspace you loaded. In fact, that's the idea: session variables are not intended for information saved in workspaces, but for information characteristic of your work session.

It seemed obvious that  $\square HT$ , which describes to the system where the tab stops have been set on your terminal, should also be a session variable. You don't want to have to set tabs again each time you load a workspace - the command `)TABS` set tabs for the session.

It had seemed to us that  $\square PW$ , which controls the maximum width of a display, should also be in that class. Our experience indicated that  $\square PW$  was used mainly as a way of matching your output to the type of terminal you were using at a particular session. Many CRTs have a width limited to 80 characters. Some printing terminals accommodate 120 characters, some 130, and some 156. So we made  $\square PW$  a session variable too --- a mistake.

We hadn't allowed for the fact that a fair number of application workspaces, some in widespread use, were making use of the value of  $\square PW$  stored in the workspace to control the format of a report. The programs, having been written before there were such things as session variables, relied on the value stored in the workspace. After the change to session variables, many users would sign on (and thereby receive a default value of  $\square PW$ ), load their workspace, and start work, only to find that the default value assigned at the beginning of the session was now being copied forward to replace the value saved in the workspace. And, naturally, those users weren't pleased with the effect! What we had intended as an additional convenience, and had seemed to work out that way in the tests we ran, was for them a nuisance.

After that, we spent a lot of effort trying to come up with a design for  $\square PW$  that would let us have it both ways. But all the proposals seemed too complicated. We have concluded, therefore, that  $\square PW$  should **not** have been made a session variable, and it was returned to its former status with a default value of 132.

To all our users who were inconvenienced by our efforts to improve  $\square PW$ , we extend our very sincere apologies. We're confident that the conversion back will be welcome, and will require of you little if any change or effort.

## LETTER FROM LONDON

---

 ROUNABOUT THE POPULATION EXPLOSION
 

---

Valerie Lusmore, London

Basildon, about 35 miles to the east of London, is one of the new towns built from scratch after the war, to spread both population and industry away from central London. The Basildon Development Corporation was formed to build and run the town for the first few years, and will hand it over to the Local Authority in five years.

APL users in Basildon, otherwise known as the Transportation and Planning Research Unit of the Development Corporation, are themselves having a "population explosion". They've just added three new users to their original two, and new applications are pouring off the terminal at a great rate. Tony Budd and Mac Mackenzie are always coming up with bright ideas and the new recruits Pete, Chris and Tony C. have added their share.

Tony and Mac were introduced to APL about four years ago by the man who sold them a 2741-type terminal; quietly, mostly on their own, they have developed population models to work out how many and what kind of new houses are needed, they have recently reorganised all the local bus-routes, and analysed the results of several surveys done by the County Council. APL is ideal for their needs as they can quickly and easily develop a set of functions to cope with a variety of problems. Although they would hardly describe themselves as "computer people" they have had quite a lot of experience with using computers and packages from a variety of computer suppliers. One in particular, a very large transportation package from IBM, is used to predict the volumes of traffic on roads. Tony reckons they debugged most of the obscure nooks and crannies of this general package single-handed! That is why they like APL so much - you aren't spending your time fitting your problem to a package that doesn't quite solve it, but tackle the problem directly, and of course, there aren't any bugs caused by someone else - so it's much easier to sort out what is happening.

A problem that they keep coming back to is that of the behaviour of traffic at junctions --- perhaps because Essex has some of the most overcrowded roads in the country. The number of cars that can pass through a junction per hour, or how long a vehicle must wait in a queue (which, of course, backs up to the last junction, and so on), is critical.

Traffic control, junctions, and when and whether a vehicle can join from another stream have been extensively studied in North America. The roundabout (or traffic circle) is a peculiarly European problem and has not been studied much at all. This type of junction is uncommon in the United States because the rule of giving way to traffic on one's right slows down the traffic already on the circle, increases congestion and prevents a smooth flow. This was in fact the situation in the U.K. until, in the 1950's, the precedence rule was changed so that traffic flows smoothly. At Basildon we built some small models to test how the American theories of gap acceptance could be applied to the British situation and, within a few hours, had produced some interesting models. These take into account the space between cars, their speed, and the variable reaction time for different drivers.

Modelling in APL certainly provides a better approach than building the road and then testing it "on-line"! Further work produced more interesting ideas - but budget restrictions forced other problems to be given priority. Perhaps though, in time, they will produce the all-singing, all-dancing model of a roundabout and then, when more roads are developed in their neck of the woods, they'll have a better way to tackle the question of "Will this road be adequate for the traffic?"

## CORPORATE PLANNING - AIDS CHEAPER WITH NEW TECHNIQUE

Cost reductions of up to 50% have been reported by users of a new AIDS *INTERFACE* workspace. *INTERFACE* allows planners to make use of conventional APL program statements for entering data, executing models, making inquiries and producing reports, and so bypassing all of the interactive AIDS commands normally used for such operations (i.e. *OVERRIDE*, *SIMULATE*, *CONSOLIDATE*, *ID*, *REPORT*).

The procedure for using *INTERFACE* involves opening and activating the required AIDS data and program files by means of the workspace 10113 *AIDS*. Workspace 10113 *INT* is then loaded and commands *REFDIVISION/REFSUMMARY*, *REFCASE*, *TIME*, *READ*, *STORE* and *WRITE* are used in conjunction with APL program statements to access the data directly.

All commands are described in the "AIDS Users Manual".

## SHARP APL GRAPHICS

A manual and a brochure are now available. The package provides:

**PICTURES:** The **Basic Graphics** system, in workspaces 3 *GRAPHICS* and 3 *GRAFIX*, provides tools to aid graphic input, create pictures (graphic objects) and manipulate them. A distinctive feature of SHARP APL Graphics is that you can talk about an entire picture as a single object, as if it were a scalar, or form a new picture by joining several separate ones much as you can join several scalars and refer to them collectively as a vector. This ability provides great elegance, simplicity and flexibility. Specialized graphics systems such as plotting and the picture editor described below, grew out of the Basic Graphics package.

**PLOTS:** Graphs are an excellent way to summarize large quantities of data. Trends and impressions are perceived at a glance, whereas the same data displayed in numeric tables would be difficult to grasp. This package (in workspace 3 *GRAFPLLOT*) provides the facilities to produce high quality graphs of the type you want.

**AN EDITOR:** It is easy to enter drawings and then edit them, or to edit drawings that have already been entered. You can display drawings on a screen, and interactively revise, extend or combine them. All this is interactive, so you can see at once the result of your efforts (load workspace 3 *GREDIT*).

**DEVICE INDEPENDENCE:** How you talk about your drawing is independent of the way the system controls your terminal. That's an important design feature of the SHARP APL graphics system. Exploiting the powers of graphic devices may require a complex sequence of codes to be transmitted between the system and the terminal, but you don't need to know about that aspect since it's done for you automatically. Moreover, the same fundamental vocabulary applies across the board in graphic work, regardless of the device you're using. (Of course, there are some actions - such as specifying the colour of ink - that aren't going to have any effect on a terminal that doesn't have various inks.)

The underlying structure is this: In each workspace, the function that causes a picture to be drawn is called *DRAW*. But before you start work, you identify the type of terminal you're using now. This generates subprograms used by *DRAW* so that, when you're ready to display a drawing, the system makes appropriate use of the device you actually have. For example, at a storage tube with point-to-point plotting, the system may be able to draw a complicated figure with a few long lines that flash clear across the screen. At a terminal that's fundamentally a typewriter, the same figure might require hundreds of dots, each carefully positioned beside the next. But either way, you don't get involved in these details; they're taken care of automatically.

---

**COMMUNICATIONS NETWORK: FLOW CONTROL ON ASCII TERMINALS**

Tony De Lucovitch and Fred Perkins, London

Flow Control support for all ASCII terminals (like the AJ860 and the Xerox Diablo 1620) is now operational in all locations except Toronto. During output, **control S** (simultaneously) will cause output to stop immediately. Then you can adjust the paper, read a VDU screen, or whatever. Hitting **control Q** subsequently will cause output to resume at the exact character that was interrupted. DC3 and DC1 correspond to Ctrl S and Ctrl Q respectively. Use of the break key is unaffected (and will also cancel the effect of a control S entirely). If control S doesn't stop output, it is likely that the terminal buffer is emptying.

---

**PUBLIC LIBRARY - AN ON-LINE REFERENCE CARD**

Ed Stubbs, Toronto

Workspace 1 *REFERENCE* provides on-line information about SHARP APL in the form of an expanded reference card - and contains several tables such as:

- a list of file access permission codes,
- a summary of *HSPRINT* specifications,
- control messages, etc.

The information is intended for reference by those who already understand the use of the system and not as tutorial material for new users. The main function in the workspace is *TABLE*. It takes as right argument a character string indicating the name of the table - or enough of the beginning of the name to identify it uniquely. Supplying an unknown name such as *TABLE ' '* elicits a list of the current tables. Unknown names are appended to a file and may be used as suggestions for new tables.

You may (but do not have to) provide a left argument to search the table indicated in the right argument. The explicit result will then be much smaller than the whole table. This capability is especially useful when tables are large, and when you know exactly what information you require. If there is no left argument, the whole table is printed.

For example,

```
'0001' TABLE 'CONTROL MESSAGES' or
1 TABLE 'CONTROL MESSAGES' or
'MIXED' TABLE 'CONTROL MESSAGES'
```

all return:

| <u>NUMBER</u> | <u>MNEMONIC</u> | <u>DESCRIPTION</u>   |
|---------------|-----------------|----------------------|
| 0001          | MIXED           | MIXED OUTPUT FOLLOWS |

The information is available at any time by simply copying *TABLE* into your active workspace:

```
)COPY 1 REFERENCE TABLE
```

when the manual containing the table is not at hand.

**A TOUCH OF GERMAN:**

Margarete Buch



**Margarete Buch** studied English and German at the University of Mannheim in Germany, and subsequently worked as a teacher in Switzerland for two and a half years. Since joining the Zurich office earlier this year, Margarete has translated some Sharp documents into German (including the Reference Card, and the File System manual), and is in the throes of proofreading the "Einfuehrung in SHARP APL". She is becoming fluent in APL also, "for it is not the German or English language that most interests me, it is APL".

**COURSE SCHEDULE**

**Seminars:**

|                  |   |                       |        |        |
|------------------|---|-----------------------|--------|--------|
| BOSTON           | 'Designing APL Business Systems'              | Jun 5                 | Jul 10 | Aug 7  |
|                  | 'Implementing APL Business Systems'           | Jun 12                | Jul 17 | Aug 14 |
|                  | 'Interactive APL Business Systems'            | Jun 19                | Jul 24 | Aug 21 |
|                  | 'Using Arrays, Packages and Files'            | Jun 26                | Jul 31 | Aug 28 |
| CALGARY          | 'MAGIC for Time Series Analysis'              | On request            |        |        |
|                  | 'Saving Money with N-tasks and B-tasks'       | On request            |        |        |
| EDMONTON         | 'Forecasting in SHARP APL'                    | On request            |        |        |
|                  | 'Origins of APL (video-tape)'                 | On request            |        |        |
|                  | 'Regression Analysis in APL'                  | On request            |        |        |
| MINNEAPOLIS      | 'APL in Financial Modelling'                  | On request            |        |        |
|                  | 'What is APL?'                                | On request            |        |        |
| OTTAWA           | 'New Features of SHARP APL'                   |                       | Jun 20 | Jul 18 |
|                  | 'Advanced File Design and Usage'              |                       | Jun 21 | Jul 19 |
|                  | 'Input Considerations (Idiot proofing)' -a.m. |                       | Jun 22 | Jul 20 |
|                  | 'Program Design for Highspeed Printing' -p.m. |                       | Jun 22 | Jul 20 |
|                  | 'Non-terminal tasks'                          | Jun 23                | Jul 21 |        |
|                  | 'MAGIC for Time Series Analysis'              | Jun 8                 | Jul 6  |        |
| TORONTO          | 'Data Base Design'                            | Jul 27                | Sep 27 |        |
|                  | 'Actuarial APL Techniques'                    | Jun 13                |        | Sep 19 |
|                  | 'AIDS' (one and a half days)                  | Jun 27                |        | Sep 26 |
|                  | 'Mathematics of Mortgages and Bonds'          | Jul 19                |        | Sep 13 |
|                  | 'Graphics'                                    | Aug 17                |        |        |
|                  | 'Plotting with SHARP APL'                     | Jul 18                |        | Sep 21 |
|                  | 'Report Formatting with SHARP APL'            | Jun 6                 | Jul 11 | Aug 8  |
|                  | 'Box-Jenkins'                                 | Jun 1                 |        | Sep 8  |
|                  | 'Regression Analysis'                         | Jun 22                | Aug 23 |        |
|                  | 'Forecasting Methods'                         | Jul 26                |        | Sep 11 |
|                  | 'Saving Money with N-tasks and B-tasks'       | Jul 13                |        | Sep 14 |
|                  | U.K. (LONDON)                                 | 'Appreciation of APL' | Jun 22 | Oct 12 |
| 'Systems Design' |   | Sep 7-8               |        |        |
| (WARRINGTON)     | 'Appreciation of APL'                         | Jul 25                |        |        |

| <b>Introduction to APL:</b>                | JUN              | JUL   | AUG   | SEP   | OCT   | NOV   |
|--|------------------|-------|-------|-------|-------|-------|
| BOSTON                                     | 20-23            | 25-18 | 22-25 |       |       |       |
| CALGARY (5 sessions)                       |                  |       |       |       |       |       |
| EDMONTON (5 sessions)                      | 20,21,27,28,Jul5 |       |       |       |       |       |
| MINNEAPOLIS                                | 13-15            | 18-20 |       |       |       |       |
| (6 half-day sessions)                      | 20-22            | 25-27 |       |       |       |       |
| MONTREAL (6 sessions once a week at 5-7pm) |                  |       |       |       |       |       |
| OTTAWA                                     | 05-09            | 03-07 | 07-11 | 11-15 |       |       |
| ROCHESTER                                  | 19-23            | 17-21 | 21-24 | 25-29 |       |       |
| TORONTO                                    | 19-21            | 03-05 | 14-16 | 05-07 |       |       |
|  |                  | 24-26 |       |       |       |       |
| U.K. (LONDON)                              | 12-14            | 10-12 | 14-16 | 11-13 |       |       |
| (WARRINGTON)                               | 26-28            |       | 21-23 |       |       |       |
| VANCOUVER                                  |                  | 04-06 | 07-09 | 18-20 | 16-18 | 20-22 |
| WINNIPEG                                   |                  |       | 22-25 |       | 10-13 |       |

**Intermediate:**

|                  |              |        |          |  |  |  |
|------------------|--------------|--------|----------|--|--|--|
| OTTAWA           | Jun 19       | Jul 17 |          |  |  |  |
| TORONTO (2 days) | Jul 31-Aug 1 |        | Sep 28-9 |  |  |  |
| WINNIPEG         | Oct 24-27    |        |          |  |  |  |

**Advanced:**

|  |  |            |               |           |  |  |
|--|--|------------|---------------|-----------|--|--|
| BOSTON   |  | Jun 8-9    | Jul 13-14     | Aug 10-11 |  |  |
| MINNEAPOLIS 'Advanced Techniques in APL'               |  | On request | (3 half-days) |           |  |  |
| TORONTO 'Advanced APL and Efficient Coding Techniques' |  | Jun 29-30  |               | Sep 18-19 |  |  |
| U.K. 'APL System Design'                               |  | Jul 6-7    | Sep 7-8       |           |  |  |

**Special Courses:**

|                                  |                                       |         |  |  |  |
|----------------------------------|---------------------------------------|---------|--|--|--|
| TORONTO 'APL for Managers'       | Jun 29-30                             | Aug 2-3 |  |  |  |
| LONDON 'Appreciation of APL'     | On request                            |         |  |  |  |
| WARRINGTON 'Appreciation of APL' | Jul 25                                |         |  |  |  |
| AMSTERDAM 'Introduction to APL'  | In Dutch, Sep 20,21,22 & Nov 22,23,24 |         |  |  |  |
| DUESSELDORF/VIENNA/ZURICH        | In German, scheduled on demand        |         |  |  |  |

**UPDATE**

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

Name: \_\_\_\_\_

Co.: \_\_\_\_\_

Address: \_\_\_\_\_

