

# VECTOR

## **APL98 Reports & Pictures ... plus**

- Jim Lucas reviews K-lite 36
- Another APL Crossword! 58
- Smith on HTML Forms and Servers 105
- Anne's Material Mountain 119
- Cannon on the Dyalog Session 138



*The Journal of the  
British APL Association*

A Specialist Group of the British Computer Society

**ISSN 0955-1433**

**[www.vector.org.uk](http://www.vector.org.uk)**

**Vol.15 No.1 August 1998**

## Contributions

All contributions to VECTOR may be sent to the Journal Editor at the address on the inside back cover. Letters and articles are welcome on any topic of interest to the APL community. These do not need to be limited to APL themes, nor must they be supportive of the language. Articles should be accompanied by as much visual material as possible (b/w or colour prints welcome). Unless otherwise specified, each item will be considered for publication as a personal statement by the author. The Editor accepts no responsibility for the contents of sustaining members' news, or advertising.

Please supply as much material as possible in machine-readable form, ideally as a simple ASCII text file on an IBM PC compatible diskette (any format). APL code can be accepted as camera-ready copy, in workspaces from I-APL, APL+Win, IBM APL2/PC or Dyalog APL/W, or in documents from Windows Write (use the APL2741 TrueType font, available free from Vector Production), and MS Word.

Except where indicated, items in VECTOR may be freely reprinted with appropriate acknowledgement. Please inform the Editor of your intention to re-use material from VECTOR.

## Membership Rates 1998-99

| Category                                        | Fee  | Vectors | Passes |
|-------------------------------------------------|------|---------|--------|
| UK Private                                      | £12  | 1       | 1      |
| Overseas Private                                | £14  | 1       | 1      |
| (Supplement for Airmail, not needed for Europe) | £4   |         |        |
| UK Corporate Membership                         | £100 | 10      | 5      |
| Overseas Corporate                              | £135 | 10      |        |
| Sustaining                                      | £430 | 10      | 5      |
| Non-voting Member (Student, OAP, unemployed)    | £6   | 1       | 1      |

The membership year normally runs from 1st May to 30th April. Applications for membership should be made to the Administrator using the form on the inside back page of VECTOR. Passes are required for entry to some association events, and for voting at the Annual General Meeting. Applications for student membership will be accepted on a recommendation from the course supervisor. Overseas membership rates cover VECTOR surface mail, and may be paid in sterling, or by Visa, Mastercard or JCB, at the prevailing exchange rate.

Corporate membership is offered to organisations where APL is in professional use. Corporate members receive 10 copies of VECTOR, and are offered group attendance at association meetings. A contact person must be identified for all communications.

Sustaining membership is offered to companies trading in APL products; this is seen as a method of promoting the growth of APL interest and activity. As well as receiving public acknowledgement for their sponsorship, sustaining members receive bulk copies of VECTOR, and are offered news listings in each issue.

## Advertising

Advertisements in VECTOR should be submitted in typeset camera-ready format (A4 or A5) with a 20mm blank border after reduction. Illustrations should be photographs (b/w or colour prints) or line drawings. Rates (excl VAT) are £250 per full page, £125 for half-page or less (there is a £75 surcharge per page if spot colour is required).

Deadlines for bookings and copy are given under the Quick Reference Diary. Advertisements should be booked with, and sent to: Gill Smith, Brook House, Gilling East, YORK YO62 4JJ. Tel: 01439-788385 Email: [apl385@compuserve.com](mailto:apl385@compuserve.com)

# Contents

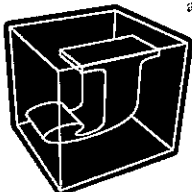
|                                           |                       | Page |
|-------------------------------------------|-----------------------|------|
| <b>Editorial</b>                          | Stefano Lanzavecchia  | 3    |
| <b>APL NEWS</b>                           |                       |      |
| Quick Reference Diary                     |                       | 5    |
| Correspondence                            |                       | 7    |
| British APL Association News              |                       | 11   |
| News from Sustaining Members              |                       | 18   |
| APL Product Guide                         | Gill Smith            | 22   |
| K-lite First Impressions                  | Jim Lucas             | 36   |
| It's APL, but not as we Know It           | Ajay Askoolum         | 47   |
| <b>The Education Vector</b>               |                       |      |
| Zark Newsletter Extracts                  | edited by Jon Sandles | 51   |
| J-ottings 17                              | Norman Thomson        | 60   |
| <b>RECENT MEETINGS — APL98</b>            |                       |      |
| APL98 Conference Reports                  |                       |      |
| A Personal View                           | David Crossley        | 66   |
| Observations                              | Ray Cannon            | 74   |
| The Dyalog OCX Workshop                   | Dyadic Systems        | 81   |
| APL98: New Gems From Old Roots            | Bob Hoekstra          | 97   |
| <b>GENERAL ARTICLES</b>                   |                       |      |
| HTML Basics for APLers — Servers          | Adrian Smith          | 105  |
| APL and the Mountain                      | Anne Wilson           | 119  |
| <b>TECHNICAL SECTION</b>                  |                       |      |
| Hackers' Corner: How Big is that Picture? | Adrian Smith          | 128  |
| Technical Correspondence                  | Sutton, McDonnell     | 130  |
| At Play with J: to Summarize              | Gene McDonnell        | 132  |
| Utilities for the Dyalog Session          | Ray Cannon            | 138  |
| Index to Advertisers                      |                       | 143  |



IT'S SURPRISING  
HOW JSOFTWARE  
MAKES YOUR DATA  
**JUMP OUT!**

J for Windows 95/NT from Iverson Software Inc. provides everything you need to solve problems quickly. You can build complete stand alone Windows applications, or easily integrate J with other programs. For example, use J as an ActiveX control for a web page, an OLE server for Excel,

or a 32-bit DLL called from a C program. J is also available on a variety of other platforms including Windows CE, Mac, Linux, Solaris and AIX. The language is identical in all versions, and programs not making use of platform-specific features will work unchanged on all systems.



*Jina Box*

Strand Software 19235 Covington Court Shorewood, MN 55331  
Tel: (612) 470-7345 Fax: (612) 470-9202  
[www.jsoftware.com](http://www.jsoftware.com)

## Editorial - APL98 and the Future of APL

*by Stefano Lanzavecchia*



APL conferences don't simply stay on your skin, they go deeply inside you and should leave you with strong, confused, irrational, exciting emotions. They are a way to look at the past, at the present and at the future with the aid of many different eyes and glasses. Let's disregard for a second the importance of social events or of the mighty banquet and let's concentrate instead on the contents. I don't believe we need to be reminded all the time that we APLers are doing great, and especially not with a pep-talk or a bloating show, but it's nice to feel that it is

actually happening by looking at what we've achieved, at the way in which we relate to the changing technologies and at what's lined up to make our lives even more interesting.

Maybe a few concrete examples will help understand what I mean. Forget for a moment that I'm currently working for the company (namely Adaytum Software [1]) which opened the show in Roma with a demonstration of what can be done when some competent programmers meet some other skilful marketing people. If you forget it will be easy for you to recognise that my pride is the pride of an APLer who enjoys the success of the language and of the philosophy behind the language. Even though Adaytum Software has still a long way to go, you can see its importance cannot be underestimated. First of all the exposure of its main product to a larger audience than is usually reached by ordinary APL applications is changing the rules of the game. Such an audience requires more and more the product to be compliant with the recognised standards in the look-and-feel of Windows applications. This means the end to the time when typical APL applications could be spotted from far away for their original look. Despite the fact that the word APL is no longer heard with fear or diffidence by the IT departments of large firms (mostly because the young managers have never heard it and they simply think it's something new, another buzzword), the real power of APL will be embodied in shells which no longer betray the origin of the brain.

Second, the success of the company is already causing an internal growth which demands for new recruits to be found at regular intervals. Since the number of APLers is larger than we tend to think, APL-skilled professionals are not so difficult to find. The problem is more that they're all already busy or reluctant to relocate. Plus there is a large number of them who are so old-fashioned and consider Windows and company an unnecessary evil that they would not fit in APL's new suit. I won't go any deeper into the problem for the moment.

Another example I would like to point our attention to is the development of APL interpreters. Never before, the interpreter's vendors have produced upgrades of their products at such a quick rate: new tools for better interoperability (GUI, intra- and inter-machine communication) and more fundamental changes. A truly brilliant display of innovation was shown by Dyadic Systems [2] who presented their new-born release of the interpreter with multithreading capabilities. The implementation is still in its infancy and it will take a while before it becomes a mature feature but even in its current state it's a major leap forward. It would be difficult to clearly identify the importance that in this process of innovation has the drive of the companies which are successfully basing their development on Dyadic's interpreter. Nevertheless it is quite evident that the aforementioned success is quite beneficial.

My final example will relink me to my last editorial. Even a quick scan through the proceedings of the conference will reveal that quite an impressive number of programming languages is referenced in addition to APL and J: some are still array-oriented languages such as K [3], Mathematica, Matlab, but there are also C, C++ and dialects, Forth, Java [4], even Tcl/Tk [5], and finally the bizarre hybrids such as HTML [6] augmented with the scripting languages VBScript [7] and ECMAScript [8]. It seems that I'm not the only one believing in the use of many different knives and it is with pleasure that I register this fact.

## References

- [1] Adaytum Software: <http://www.adaytum.com/>
- [2] Dyadic Systems: <http://www.dyadic.com/>
- [3] Kx Systems: <http://www.kx.com/>
- [4] Java: <http://java.sun.com/>
- [5] Scriptics Corporation: <http://www.scriptics.com/>
- [6] HTML: <http://www.w3.org/MarkUp/>
- [7] VBScript: <http://msdn.microsoft.com/scripting/>
- [8] ECMAScript: <http://www.ecma.ch/stand/ecma-262.htm>

## Quick Reference Diary 1998-2000

| Date              | Venue            | Event                   |
|-------------------|------------------|-------------------------|
| November 1-4 1998 | Orlando, Florida | APL2000 User Conference |
| August 10-14 1999 | Scranton, Penn.  | APL99                   |
| Late July 2000    | Berlin           | APL'00                  |

### APL2000 User Conference

APL2000 Inc is pleased to announce the third annual APL2000 User Conference to be held November 1st through November 4th at the Clarion Plaza Hotel in Orlando Florida. We have planned an exciting program with sessions tailored to APL developers as well as those who manage the development process. In addition, we will be holding sessions of special interest to actuaries and Internet users.

---

### Dates for Future Issues of VECTOR

|              | Vol.15<br>No.2 | Vol.15<br>No.3 | Vol.15<br>No.4 |
|--------------|----------------|----------------|----------------|
| Copy date    | 4th Sept       | 4th Dec        | 5th March      |
| Ad booking   | 11th Sept      | 11th Dec       | 12th March     |
| Ad Copy      | 18th Sept      | 18th Dec       | 19th March     |
| Distribution | October 98     | January 99     | April 99       |
| Edited by    | TBA            |                |                |

---

### Quick Puzzle

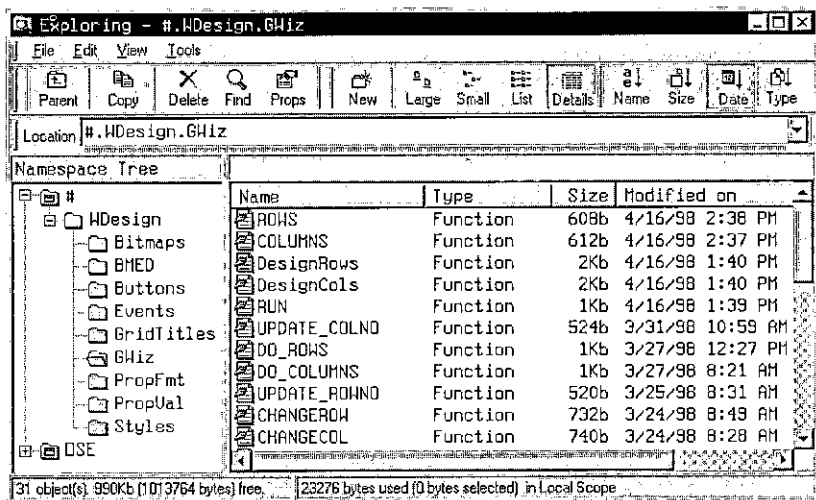
From: Ajay Askoolum

APL usually boasts at least one other way of doing the same thing.

Given a keyboard with the  $\pm$  (execute) and  $\square$  (quad) keys being inaccessible, how would you convert incoming character scalar integers (including ones specified as blanks) to numbers?

# dyalog APL

The Definitive APL for Windows™



## Dyalog APL Namespaces let you ...

- ⇒ Organise your workspace into **self-contained** sub-systems
- ⇒ **Encapsulate** functions and variables *within* GUI objects
- ⇒ **Isolate** utilities from your application code
- ⇒ Construct **COM/DCOM** object hierarchies
- ⇒ Support **multiple instances** in APL web server applications

That's why Dyalog APL/W remains the **professional** choice. For further information, contact Dyadic or your local distributor today.

Dyadic Systems Limited, Riverside View, Basing Road, Old Basing,  
Basingstoke, Hants. RG24 7AL, United Kingdom.  
Tel:+44 1256 811125 Fax:+44 1256 811130 Email: [sales@dyadic.com](mailto:sales@dyadic.com)

Microsoft is a registered trademark and Windows and the Windows Logo are trademarks of Microsoft Corporation



# CORRESPONDENCE

## A Response to the Responses in Vector...

From: David Piper (davidp@selectst.com)

30 July 1998

Well, it made for an interesting edition of Vector, if nothing else. First, blame the guilty parties — OK *mea culpa*. All transcription errors were entirely my own, I could of course blame the technology (I don't have), or the blinding heat of enthusiasm. But no, when it comes right down to it, I transcribed some of the code wrongly. Much amusement engendered, tongue-in-cheek articles etc. etc. etc. For Phil Last's benefit — the code did run and the timings were genuine...

However let's not throw the baby out with the bath water. The message I was trying (some might say failing) to get across was simply that it's not worth trying to speed up code by taking advantage of some quirks in one (or more) interpreters when there remains a lot you can do simply by making the algorithm more appropriate (to APL in this instance).

In the original article the algorithm implemented worked on a one-at-a-time basis — a sure fire recipe for bad performance in APL. So attention should be focussed on creating a parallel algorithm rather than on tweaking the code. Sometimes — as in this example — the changes required to move from one-at-a-time to many-at-a-time are minimal and the benefits so significant that further tweaks are probably not justified. In other cases creating the parallel algorithm is so difficult (a.k.a. impossible) that we have to end up resorting to tweaks.

Even where parallel algorithms are too costly to justify their development, there are techniques such as loop-hoisting (i.e. ensuring loop invariant lines of code are outside the loop) which should be applied before (potentially) interpreter dependent tweaks. Most experienced practitioners do this sort of thing automatically, so the scope for deriving benefits may be limited — but where they can be applied, such techniques can give rise to substantial benefits.

Claude Henriod makes some interesting and valid points in his response. Maybe it was a mistake to introduce the concept of "style" into my original letter — after all style is subjective. I stick by the use of longer variable names — because I find they make life easier. I plead guilty to the complex path issue — maybe with more thought I would have improved this.

I stoutly defend the “only one assignment per line” rule. The problem with assignments is that they are essentially side-effects. If you adhere to the rule of one-assignment-per-line then you stand a chance of being able to restart any line when it collapses with an error. Assuming of course that side effects have not been created by other actions (like shared variable activity, system function activity or the use of global variables in functions invoked on the line — maybe these are also the basis for important style guidelines). So in the interests of being lazy (something which I find appeals to a lot of people), stick to one assignment per line.

Anyone want to think about diamonds?

David Piper  
11 Lingfield Road  
Evesham  
Worcs WR11 6XG

---

## Honorary Doctorate for Ken Iverson

*reported by Roger Hui*

Ken received a honorary Doctor of Science degree from York University here in Toronto on Thursday, June 11.

### Citation for Dr. Kenneth Iverson

Dr. Kenneth Iverson began his studies in Mathematics as an undergraduate student at Queen’s University and went on to study at Harvard where he earned an MA and Ph.D. It was while doing his graduate work that he became interested in computing and helped to establish the first computer applications program of study at Harvard University. In 1954, when relatively senior members of our current faculty were in elementary school, Dr. Iverson delivered a lecture on Graduate Instruction and Research to Harvard’s “First Conference on Training Personnel for the Computing Machine Field.” The lecture laid out a design for the first graduate course in computing science given at Harvard — a course Dr. Iverson went on to teach.

At that time he also developed a special mathematical notation (now known as the Iverson notation) to assist in writing about computer applications. The notation has been described by one of our faculty colleagues as being "a transparent and illuminating notation for mathematical concepts".

In the early 1960's, having moved to IBM, Dr. Iverson made an historic contribution to computer science by inventing APL, a programming language. APL was revolutionary in at least two respects: first, it was designed to interact with programmers and users through a typewriter terminal, rather than running in batch mode from punched cards or tape. It inspired computer scientists with the possibilities of interactive computing, well before that form of computing was at all common. Second, APL took programming to a higher level of abstraction than previous languages, which abstraction along with its concise notation, allowed enormous increases in programming productivity. Iverson emphasized that a programming language should be designed primarily as tool for expressing human thought, and only secondarily as a tool for instructing a computer. For his work, he was in 1971 made an IBM Fellow, the highest non-managerial position within the corporation. In 1979, he was awarded the A.M. Turing Award from the Association for Computing Machinery — the most prestigious award in computer science.

Dr. Iverson has always placed a great value on sharing knowledge through teaching. He has written several books on teaching mathematics, and APL was originally conceived as a teaching tool. He has continued to be active in his field, and with his son Eric and colleague Roger Hui, he has developed a new language, called "J". This new language is now gaining attention worldwide.

Nominations for Dr. Iverson to be honoured by York University came not only from computer scientists and mathematicians, but also from psychologists and philosophers. One of the latter, noted that if Edna St. Vincent Millay was right in saying that "Euclid alone has looked on beauty bare," it was the case that Kenneth Iverson had "caught glimpses and conveyed them lucidly to the rest of us."

It is not only because York University was one of the first institutions outside IBM to make use of APL that we are particularly proud to honour Kenneth Iverson today. A creative inventor, pedagogical pioneer and breaker of disciplinary boundaries, Dr. Iverson exemplifies our highest ambitions for our own faculty and students. Therefore Mr. Chancellor, I am pleased to request that you confer on Dr. Kenneth Iverson the degree of Doctor of Science, *honoris causa*.

## APL98 Retrospective

From: Paolo Di Chio

September 14th 1998

Dear APL friends, first of all let me thank all of you that came to the APL98 Conference in Rome for the participation and for the great enthusiasm shown. I believe that the Conference as a whole was appreciated very much by all the participants; at least this was my impression from the comments heard during the conference, and from several very kind messages received after the end. Looking at the scientific program, unfortunately I can give no direct feeling, because the many duties I had to satisfy didn't allow me to follow any of the many talks, tutorials and workshops. At any rate I heard rumours that convinced me that the scientific program was quite valuable in any respect: talks were of a good level and people find tutorials and workshops really profitable.

Let me thank here all the fantastic people that with their works, their personal sacrifice and their enthusiasm made the Conference possible; first I would thank the members of Program Committee for the really valuable advice I received; then the Organising Committee fellows that freed me from many heavy tasks and, last but not least, all the delicious girls that, on the buses, at the reception desk and in any other circumstance did their job always with a smile, trying to help anyone with a problem and looking everytime fresh, pretty and available. I remember somebody asked me "Where did you put the ugly girls?"

Moreover I would like to point out one more thing: it seems to me that among all the attendees there was truly a spirit of cooperation and I think that the possibility to talk each other and exchange opinions and experiences has been appreciated by all the participants. Such a possibility was of course because the Conference had a moderate parallelism among the sessions. I firmly believe that this has to be one of the main goals of a good conference, not only running from one room to the other trying to follow five or six talks at the same time.

One feature of APL98 that seems to me has been proven quite successful is the set of social events. I think everybody enjoyed the gala dinner (someone told me that he didn't eat in such a way even at his wedding party), but I'm sure that people liked also the musical evening and the Rome by night tour. The accompanying person program also earned a great success. I'm very sorry for all the people that left before Friday evening, because they missed the Survivors' bruschetta and wine: I think that this has been the most enjoyable event of all: thanks to the Jim Lucas music (maybe due also to the flood of excellent Frascati wine) everybody had a very good time tasting some typical Roman food, in spite

of the painful attempt of myself to play guitar and, worst of all, singing Roman songs.

Of course there were some shadows: my main disappointment was the low number of attendees; perhaps this happened because of the poor communication and publicity for the Conference and the delay in the availability of the Program.

Although I said that the organisation of a conference should be done at least once in everybody lifetime, but not twice, if you want to have a long life, after this experience I'm pleased to be self-contradictory: I truly hope that sooner or later I'd have the honour and the onus to host the APL community once again in Italy.

To close these few words let me tell you the most important target APL98 Conference achieved as far as my personal life is concerned: before it I had just some APL colleagues and acquaintances, now I can safely affirm that I have a lot of true friends all over the world. Thanks also for that.

---

## British APL Association - Annual General Meeting (RSS, London)

### Minutes of the Annual General Meeting

The Meeting opened at 1.30 pm on 22 May 1998.

1. The minutes of the last AGM, which were published in Vector 14.1, were taken as read.
2. The chairman reported on the year. There had been a problem when the Editor of Vector was overloaded but we now appear to be over it.

The Association is financially sound. The membership is dropping — very slowly, it is true — but members should all try to increase it. A larger membership would be beneficial to all. He concluded by thanking the people who had been particularly helpful to the Association this year.

3. No other member of the committee wished to make a report, and there were no questions anyone wished to ask of committee members.
4. The Treasurer circulated copies of the accounts and these were accepted without question.

5. The Treasurer proposed and the Chairman seconded the motion that John Sullivan be appointed auditor for next year. John indicated that he was willing and was appointed *nem con*.
6. The Chairman announced the Committee for 1998-1999 would be:

|                            |                           |
|----------------------------|---------------------------|
| Chairman . . . . .         | Anthony Camacho           |
| Secretary . . . . .        | Ajay Askoolum             |
| Treasurer . . . . .        | Nicholas Small (co-opted) |
| Editor . . . . .           | Vacancy.                  |
| Projects & Publicity . . . | Dr Alan Mayer             |
| Activities . . . . .       | Jon Sandles               |
| Webmaster . . . . .        | Ray Cannon                |
7. The meeting was asked whether anyone wished to raise any other business. No-one did so the Chairman declared the meeting over at 1.55 pm.
8. The AGM was attended by 18 people including the Committee.

### Chairman's Report on the BAA Year 1997/8

The main benefit we provide to our members is Vector, which for fourteen years has been the best APL journal in the world. We would like to hold more meetings, such as the vendor forum this afternoon, but not many members have been keen to attend or to help organise them and so this is the first meeting since the last AGM. Furthermore we have had prospective speakers, that you might have liked to hear, refuse our offer of a platform on the grounds that the number of people we could expect to sit and listen was too few to justify the effort. You may hear about more meetings from Jon Sandles.

The Chairman's job is to hold things together when there are difficulties and we have had some difficulties; my efforts at holding things together were begun rather late because of problems at home, but I believe things are running well again now. I had better explain.

Duncan found he had taken on too much and that being Editor of Vector could not be combined with a new baby. I realised this and did not call the working group together to help out as soon as I should have done. This was because my mother, who is 86, fell down stairs last October and fractured her skull, requiring me to visit her in a Hastings hospital every week-end for a month or so and then my wife broke her ankle on 18 November which more or less prevented the two of us travelling any great distance until the plaster came off in January. It was thus not until 20 January that I started to get a meeting together and we actually met on 7 February. It is a tribute to the Vector working group that we have recovered so well since then.

We have published the October Vector and the January Vector; the April Vector has gone to press with Stefano Lanzavecchia as guest editor and will be given to every delegate at APL 98 and the July Vector, with reports of this meeting and of APL 98 in Rome will be published as soon as we can after the conference reports have been edited

— probably early in August. We intend to publish the October Vector this year on time and to keep to schedule from then on. One consequence of the delay has been that this year there have only been three Vectors to pay for instead of four so our accounts look good.

As reported in Vector 14.1, the Committee decided to stop sending out the 16 page section of Vector known as The Education Vector to the I-APL mailing list. All were agreed that it was no longer doing any significant good in promoting APL in Education. This has also significantly reduced our expenditure. We would be glad to take any educational initiatives that we can find volunteers to help with, but at present we have no proposals.

Membership continues to reduce, slowly it is true, but the trend is a long-standing one. The number of Vectors we send out is currently 22 less than it was last year and the number of individual members is seven less than it was this time last year. Nearly 60% of our members live abroad. The administrative assistant's report on membership will be printed in full in Vector so anyone who is interested can see the details.

We are going to send a recent Vector to all those who have failed to renew their membership in the last two or three years with a message asking them to renew. I would also urge all members to encourage someone new to join. We welcomed 33 new members in the year and if each individual member were to recruit a new member next year we would welcome nearly 400 instead of 33. A larger membership might enable us to hold well attended meetings, would improve the incentive to write for Vector and maybe would stimulate new activities. It is strongly in your interests to encourage the APL Association to grow.

The Association owes its thanks to the people who have helped us this year:

1. the members of the committee, in particular Ajay Askoolum, our secretary, and Nicholas Small, our Treasurer, and Jon Sandles for organising this afternoon's meeting;
2. the Vector working group, in particular Stefano Lanzavecchia for editing 14.4, Richard Weber & family for hosting two Vector meetings and Adrian Smith for the extra work he has had to do coping with changes of editor and a guest editor who lives abroad;
3. all the authors who have written for Vector; and
4. our two administrative assistants, Rowena Small (membership) and Gill Smith (Vector). I discovered that the rate of pay for the job had not been changed since the jobs were created and so this year their pay was increased to £10 an hour.

Anthony Camacho, 21 May 1998.

## Administrative Assistant's Report on Membership

Membership at 30.4.97 (previous year's figures in brackets)

|               | UK        |           | FOREIGN   |           | TOTAL     |           |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|
|               | Number    | Vectors   | Number    | Vectors   | Number    | Vectors   |
| Sustaining    | 4 (5)     | 21 (41)   | 4 (4)     | 78 (78)   | 8 (9)     | 99 (119)  |
| Corporate     | 10 (10)   | 66*(67)   | 3 (3)     | 30*(30)   | 13 (13)   | 96*(97)   |
| Corp. Ind.    | 30 (29)   | 30*(29)   | 1 (1)     | 1*(1)     | 31 (30)   | 31*(30)   |
| Individual    | 158 (155) | 156 (153) | 228 (238) | 228 (238) | 386 (393) | 384 (391) |
| Non-voting    | 14 (9)    | 13 (8)    |           |           | 14 (9)    | 13 (8)    |
| Life          | 1 (1)     | 1 (1)     | 1 (1)     | 1 (1)     | 2 (2)     | 2 (2)     |
| Library       | 4 (3)     | 4 (3)     | 5 (6)     | 5 (6)     | 9 (9)     | 9 (9)     |
| Russians      |           |           | 11 (11)   | 11 (11)   | 11 (11)   | 11 (11)   |
| APL Groups    |           |           | 14 (14)   | 42 (42)   | 14 (14)   | 42 (42)   |
| TOTAL VECTORS |           |           |           |           | 687 (709) |           |

\* These figures together make the total subscribed for by UK Corporate members.

N.B. Renewal notices for November and February went out late (10 people not yet renewed).

### New Members

|    |             |    |          |             |    |
|----|-------------|----|----------|-------------|----|
| UK | Individuals | 16 | Overseas | Individuals | 17 |
|    | Non-voting  | 4  |          |             |    |

We have a new corporate member for 1998/99 — Medilab, Denmark.

# British APL Association — Summary of Annual Accounts 1997/98

*from Nicholas Small (Hon Treas)*

## Summary of income and expenditure/receipts and payments:

|                                | 1997/98<br>(R&P)<br>£ | 1997/98<br>(I&E)<br>£ | 1996/97<br>(I&E)<br>£ |
|--------------------------------|-----------------------|-----------------------|-----------------------|
| <b>Income/Receipts</b>         |                       |                       |                       |
| Subscriptions                  | 11487                 | 11021                 | 11364                 |
| BCS services                   | 0                     | 0                     | 106                   |
| Bank interest                  | 3322                  | 3322                  | 3022                  |
| Vector advertising (incl.VAT)  | 3515                  | 3471                  | 4605                  |
| Conferences & seminars         | 0                     | 0                     | 0                     |
| Other                          | 108                   | 108                   | 351                   |
| <b>Total receipts</b>          | <b>18431</b>          | <b>17922</b>          | <b>19448</b>          |
| <b>Expenditure/Payments</b>    |                       |                       |                       |
| Meetings                       | 0                     | 0                     | 358                   |
| Administration                 | 1695                  | 1152                  | 2135                  |
| BCS services                   | 0                     | 0                     | 106                   |
| Vector production and despatch | 14385                 | 14584                 | 15315                 |
| Education Vector supplement    | 931                   | 108                   | 3446                  |
| Projects                       | 569                   | 501                   | 1117                  |
| Conferences and seminars       | 0                     | 0                     | 0                     |
| Other                          | 347                   | 308                   | 670                   |
| <b>Total payments</b>          | <b>17927</b>          | <b>16653</b>          | <b>23148</b>          |
| <b>Assets summary:</b>         |                       |                       |                       |
| Bank and other balances        | 52525                 | 52021                 |                       |
| Debtors                        | 2071                  | 2188                  |                       |
| Creditors                      | (5270)                | (6151)                |                       |
| <b>Net assets</b>              | <b>49327</b>          | <b>48059</b>          |                       |

## Notes

The BCS have informed me that interest on the Gold account was £432.47 more than originally stated at the AGM. This affects total income, bank balance and net assets — these are the only figures revised.

Pence figures have been omitted, so columns may not add exactly.

The value of stocks of Vector and the Software Library have not been assessed, nor has the value of the Association's computing hardware and software.

For 1997/98, figures are shown both as income and expenditure, i.e. revenues strictly relating to the activities of that year, and as receipts and payments, i.e. what goes in and out of our bank account. The comparative figures for earlier years relate to income and expenditure.

Note that from 1997/98 VAT collection payments are made net of any rebate, so VAT rebates no longer appear under Income/Receipts, Other.

## ***Vector Back Numbers***

Back numbers of Vector are available from:

British APL Association,  
c/o Gill Smith,  
Brook House, Gilling East,  
YORK YO6 4JJ

Price in UK: £10 per complete volume (4 issues);  
£12 (overseas); £16 (airmail) including postage.

## Adjustments to R&amp;P for 1997/98 to Obtain I&amp;E

£

**Add to income:**

|               |          |                                            |
|---------------|----------|--------------------------------------------|
| Subscriptions | -644.50  | late for 1996/97                           |
|               | 411.75   | paid in 1996/97                            |
|               | 561.25   | outstanding 1997/98                        |
|               | -794.00  | advance payments for 1998/99               |
| Vector ads    | -1043.75 | from Vol 13 (incl VAT except for 13.4)     |
|               | 1000.00  | outstanding for Vol 14 (excl VAT for 14.4) |

**Add to expenditure:**

|           |          |                                          |
|-----------|----------|------------------------------------------|
| Vector    | -3538.48 | production of 13.4                       |
|           | -462.91  | late expenses claims, 1996/97            |
|           | 3400.00  | production of 14.4 (estimate)            |
|           | 800.00   | late expenses claims, 1997/98 (estimate) |
| Ed Vector | -823.06  | for 13.4                                 |
| Projects  | -67.80   | Vectors to J conference, 1996/97         |
| Admin     | -642.20  | late expenses claims, 1996/97            |
|           | 100.00   | Outstanding expenses claims (estimate)   |
| Other     | -100.40  | VAT, 1996/97 Q4                          |
|           | 71.58    | VAT, 1997/98 Q4                          |
|           | 10.20    | VAT refund, 1997/98                      |
|           | -104.00  | Vector storage 1996/97                   |
|           | 104.00   | Vector storage 1997/98                   |

# News from Sustaining Members

## Dyadic Systems Ltd

Dyadic is pleased to announce Dyalog APL/W Version 8.2. Although this version is not scheduled for general release until 4th quarter 1998, it was exposed to hands-on use by APL98 workshop attendees and came through with flying colours. Version 8.2 contains two highly significant major enhancements, namely multi-threading and the ability to write ActiveX controls in APL, as well as many other new features and extensions.

## Multi-Threading

Multi-threading allows you to execute more than one APL function (or indeed any expression) at the same time. This allows you to perform background processing, such as printing, database retrieval, database update, calculations, and so forth while at the same time perform other interactive tasks.

Multi-threading may improve throughput and responsiveness and is an essential ingredient of an APL web server.

A thread is created by calling a function *asynchronously*, using the new primitive operator *spawn*: &.

Any thread can spawn any number of sub-threads, subject only to workspace availability. This implies a hierarchy in which a thread is said to be a child thread of its parent thread. Each thread has a unique thread number, with which, for example, its presence can be monitored or its execution terminated.

Threads are implemented by the following new language elements:

- A primitive operator called *spawn*: &.
- System functions: `⌈TID`, `⌈TNUMS`, `⌈TKILL`, `⌈TSYNC`.
- An extension to the GUI Event syntax to allow asynchronous call-backs.
- A control structure: `:Hold`.
- System command: `)HOLDS`.

## APL ActiveX Controls

The new *ActiveXControl* object allows you to package a Dyalog APL application as an ActiveX Control. To run your application, all the user needs is your control and a copy of the Dyalog APL dynamic link library.

Packaged as an ActiveX control, your Dyalog APL GUI application may be included in a program written in Microsoft Visual Basic, Microsoft Excel, Borland Delphi, or in any other host application that supports ActiveX.

This feature also allows your Dyalog APL application to be run *inside a web page* hosted by Microsoft Internet Explorer, or by any other web browser that supports ActiveX. With Version 8.2, it is now possible to implement and distribute applets, written in Dyalog APL, that are accessible to any user with Internet Explorer on the World Wide Web. This capability offers enormously expanded potential for the exploitation of APL in application development.

## Other Enhancements

Other enhancements include the following:

- Additional APL2 language features
- Permanent session log
- *:With* control structure
- CoolBar object
- Splitter object
- Calendar object
- TabControl object
- Colour printing of APL functions
- Improved Bitmap support
- PNG and GIF support
- Web Server workspace

## More Information

Further details are available at <http://www.dyadic.com>

## APL2000 Inc

### Major Features of APL+Win version 3.0:

**COM Support:** Allows an APL application to be called from other Windows development tools using Microsoft's component object model (COM). A developer can have an APL calculation engine driven by a front-end written in Visual Basic, or call APL from Excel or Java. Allows one APL session to manage multiple other APL sessions.

**Network Interface:** TCP/IP support allows Internet enabled APL applications. Also included are sample workspaces that demonstrate: FTP client and server; ITalk — Internet chat facility; HTTP server and utilities; Simple chat/draw across network; ECHO client and server; SMTP email client.

**ODBC Server:** Allows workspace variables in an APL application from native or component files to appear as an ODBC-compliant database to any ODBC client. Included with APL+ODBC are the client driver and a sample server workspace.

**Media Control:** A media player control for AVI, ANI, WAV, and MIDI files.

**Revised On-line Help System:** Easier access to any object's properties, methods, and events.

### Control Improvements

Improved support for ActiveX controls; embedded objects in RichEdit controls; new seltext property for Edit controls; tooltips on disabled objects; tooltips on labels; tooltipwidth — allows multi-line tooltips; onMouseEnter and onMouseLeave events; mouse events on labels; in-place editing of Treeview items; in-place editing of Listview items; order items and columns in Listview; place pictures in Listview.

### Other Improvements:

Squad — INDEX function; Iota and epsilon faster with nested or heterogeneous arguments; open the function editor at a given line; Ctrl-T toggles between editor and session.

**New APL Utilities:** New workspaces with functions to: operate the Audio CD player and Multimedia Player (.avi, .mpg, .mid, etc.); print a tree or listview (in colour); print APL functions in colour; decode Windows error messages; find all Windows API calls in a workspace with their aliases and parameters; find DLL dependencies in a module; clean up INI and ADF files.

Miscellaneous: MS Word 6 .DOT file templates for easier insertion of APL characters into Word documents.

## Causeway Graphical Systems Ltd

The first thing we must do is send our best wishes for the future to Duncan and Jon, who have moved on to join Adaytum Software's growing team at York. From being Causeway developers, they are now full-time Causeway users — anyone who heard Morten Kromberg's plenary session at Rome will know that the core of Adaytum Planning is rapidly being rewritten around the Causeway technology.

The Causeway product set is now fully documented, stable and readily available from our web site. The *CausewayPro* manual and tutorial have been fully revised and distributed to existing customers. *RainPro* has been updated to include:

- horizontal bar charts
- bitmaps and metafiles placed anywhere on a chart
- graphical editing and drill-down capabilities

*NewLeaf* (for Dyalog APL) has also gained many new features, for example:

- automatic horizontal folding of wide tables, with key columns retained on each page
- 'groups' which define collections of text and charts and prevent unhelpful page-breaks
- html and csv output in parallel to the report production
- 'overlays' to allow arbitrarily complex master pages

Serial numbers after 97042 entitle you to a free upgrade, so please download the software (including updated help) from the web site and contact us or your local dealer for the password and an updated user manual. Users with serial numbers prior to this may upgrade for a fee of £50 per product.

In common with most software companies, we are now up to our ears in work! Marc is in Basel fully occupied with TCP/IP sockets and SQL, Adrian and JoHo are busy with project work for Adaytum and SAP, and we have a major mainframe-to-Dyalog conversion job to start on from October. Please use email where you can for support calls, and try to help us out with clear and simple diagnostic information. If we are out of the office when you phone, leave us a short message and someone will get back to you as soon as they can.

# The Vector Product Guide

*compiled by Gill Smith*

VECTOR's exclusive Product Guide aims to provide readers with useful information about sources of APL hardware, software and services. We welcome any comments readers may have on its usefulness and any suggestions for improvements.

We reserve the right to edit material supplied for reasons of space or to ensure a fair market coverage. The listings are not restricted to UK companies and international suppliers are welcome to take advantage of these pages.

For convenience to readers, the product list has been divided into the following groups ('poa' indicates 'price on application'):

- Complete Systems (Hardware & Software)
- APL and J Interpreters
- APL-based Packages
- Consultancy
- Other Products
- Overseas Associations
- Vendor Addresses
- World Wide Web and FTP Sites

Every effort has been made to avoid errors in these listings but no responsibility can be taken by the working group for mistakes or omissions.

We also welcome information on APL clubs and groups throughout the world.

*Your listing here is absolutely free, will be updated on request, and is also carried on the Vector web site, with a hotlink to your own site. It is the most complete and most used APL address book in the world.  
Please help us keep it up to date!*

All contributions and updates to the Vector Product Guide should be sent to Gill Smith, at Brook House, Gilling East, York, YO6 4JJ. Tel: 01439-788385, Email: [apl385@compuserve.com](mailto:apl385@compuserve.com)

## COMPLETE APL SYSTEMS

| COMPANY | PRODUCT           | PRICES(£) | DETAILS                                                                                                                                                                      |
|---------|-------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dyadic  | IBM RS/6000 MD320 | 11,736    | APL POWERstation (Greyscale) 27.5 MIPS, 7.4 Mflops RISC Processor 8Mb RAM, 120Mb Disk<br>19" 1280x1024 Greyscale Graph Display AIX, OSF Motif, Dyalog APL (1-user)           |
|         | IBM RS/6000 MD320 | 13,817    | APL POWERstation (Colour) 27.5 MIPS, 7.4 Mflops RISC Processor 8Mb RAM, 120Mb Disk<br>16" 1280x1024 Colour Graphics Display AIX, OSF Motif, Dyalog APL (1-user)              |
|         | IBM RS/6000 MD320 | 22,656    | Advanced APL POWERstation 27.5 MIPS, 7.4 Mflops RISC Processor 16Mb RAM, 320Mb Disk, 150Mb Tape<br>16" 1280x1024 Colour Graphics Display AIX, OSF Motif, Dyalog APL (1-user) |
|         | IBM RS/6000 MD520 | 37,114    | APL POWERsystem (8-users) 27.5 MIPS, 7.4 Mflops RISC Processor 16Mb RAM, 320Mb Disk, 150Mb Tape CD-ROM Drive, 16 Ports<br>AIX, Dyalog APL (2-8 user licence)                 |
|         | IBM RS/6000 MD530 | 72,054    | APL POWERsystem (16-users) 34.5 MIPS, 10.9 Mflops RISC Processor 32Mb RAM, 1.34Gb Disk, 2.3Gb Tape CD-ROM Drive, 16 Ports<br>AIX, Dyalog APL (8+ user licence)               |
|         | IBM RS/6000 MD540 | 122,842   | APL POWERsystem (32-users) 41 MIPS, 13 Mflops RISC Processor 64Mb RAM, 1.7Gb Disk, 2.3Gb Tape CD-ROM Drive, 32 Ports<br>AIX, Dyalog APL (8+ user licence)                    |
| Optima  | IBM Compatible    | poa       | Complete networked or stand-alone solutions including configuration installation, maintenance and commissioning.                                                             |

## APL INTERPRETERS

| COMPANY                         | PRODUCT                  | PRICES(£)          | DETAILS                                                                                                                                                             |
|---------------------------------|--------------------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| APL Software                    | APL*Plus/PC Release 10   | 450                | STSC's APL for IBM PCs & compatibles. Upgrades from earlier releases also available.                                                                                |
|                                 | Run-time                 | poa                | Closed version of APL*Plus/PC which prevents user exposure to APL                                                                                                   |
|                                 | APL*Plus II              | 1,395              | All the features of mainframe APL*Plus for your 386PC!                                                                                                              |
|                                 | Run-time<br>Dyalog APL   | poa<br>1000-10,000 | 2nd generation APL for Unix systems                                                                                                                                 |
|                                 | APL2/PC                  | poa                | IBM's APL 2 for the PC.                                                                                                                                             |
| Beautiful Systems               | Dyalog APL/W for Windows | poa                | US Distributor of Dyalog APL products from Dyadic.                                                                                                                  |
|                                 | Dyalog APL for Unix      | poa                | See Dyadic listing for product details.                                                                                                                             |
| The Bloomsbury Software Company | APL+PC Version 11        | 250                | Upgrade to version 11 gives free runtime (£120 from any version)                                                                                                    |
|                                 | APL+Win v3.0             | 1350               | A 32-bit Windows-hosted Interpreter that runs under all Windows platforms including Windows 95. Note: Upgrade for £350 from version 1.8 or 2, £920 from version 1.0 |
|                                 | Migration to APL+Win     | 1060               | from APL*PLUS/PC APL*PLUS/DOS any version.<br>from earlier versions of APL*PLUS II                                                                                  |
|                                 | APL+DOS                  | 1250               | APL*PLUS II DOS is renamed to APL+DOS.                                                                                                                              |
|                                 | Migration to APL+DOS     | 760                | from APL*PLUS/PC                                                                                                                                                    |
|                                 | APL Link                 | 200                | Database Access                                                                                                                                                     |
|                                 | APL Link Pro             | 500                |                                                                                                                                                                     |

|                                                            |                                |            |                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------|--------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                            | APL*PLUS II for UNIX           | poa        | APL2000's 2nd generation APL for all major Sparc and Risc Unix workstations.                                                                                                                                                                                        |
|                                                            | APL*PLUS VMS                   | poa        | 2nd generation APL for DEC VAX computers under VMS.                                                                                                                                                                                                                 |
|                                                            | APL*PLUS Mainframe             | poa        | Enhances VS APL with many high performance, high productivity features. For VM/CMS and MVS/TSO offers simple upgrade from VS APL.                                                                                                                                   |
| Dinsoft Oy                                                 | Dyalog APL/W for Windows       | poa        | Finnish distributor of Dyalog APL products.                                                                                                                                                                                                                         |
|                                                            | Dyalog APL for Unix            | poa        | See Dyadic's listing for product details.                                                                                                                                                                                                                           |
| Dyadic                                                     | Dyalog APL for DOS/386         | 995        | Second generation APL for DOS. Runs in 32-bit mode, supports very large workspaces. Unique "window-based" APL Development Environment and Screen Manager. Requires 386/486 based PC or PS/2, at least 2Mb RAM, EGA or VGA, DOS 3.3 or later.                        |
|                                                            | Dyalog APL/W for Windows       | 995        | As above, plus object-based GUI development tools. Requires Windows 3.0 or later.                                                                                                                                                                                   |
|                                                            | Dyalog APL for Unix            | 995-12,000 | Second generation APL for Unix systems. Available for Altos, Apollo, Bull, Dec, HP, IBM 6150, IBM RS/6000, Masscomp, Pyramid, NCR, Sun and Unisys machines, and for PCs and PC/2s running Xenix or AIX. Oracle Interface available for IBM, Sun and Xenix versions. |
| IAC/Human Interfaces                                       |                                |            |                                                                                                                                                                                                                                                                     |
|                                                            | I-APL/Mac                      | 13         | Macintosh version of I-APL.                                                                                                                                                                                                                                         |
| I-APL Ltd                                                  | I-APL/PC or clones             | 8          | ISO conforming Interpreter. Supplied only with manual (see 'Other Products' for accompanying books).                                                                                                                                                                |
|                                                            | I-APL/BBC Master               | 8          | As above                                                                                                                                                                                                                                                            |
|                                                            | I-APL/Archimedes               | 8          | As above                                                                                                                                                                                                                                                            |
|                                                            | Strand Software Inc            |            | Strand Software Inc has the sole selling rights to Iverson Software Inc products. I-APL stocks a few of these (mainly APLWIN and the personal J products and books), but is no longer an agent.                                                                     |
| IBM APL Products                                           | TryAPL2                        | free       | APL2 for educational or demonstration use. Write, fax or Email to APL Products; specify disk size desired.                                                                                                                                                          |
|                                                            | APL2 PC (US Version)           | \$630      | Product No. 5799-PGG, PRPQ Number RJ0411. Order from 1-800-IBM-CALL.                                                                                                                                                                                                |
|                                                            | APL2 PC (European Version)     | £348       | Product No. 5604-260. Part number 38F1753. From all IBM dealers, including MicroAPL.                                                                                                                                                                                |
|                                                            | APL2 for OS/2 Entry Edition    | \$185      | Part No 89G1555.                                                                                                                                                                                                                                                    |
|                                                            | APL2 for OS/2 Advanced Edition | \$650      | Part No 89G1697. Contains all facilities of the Entry Edition plus: DB2 Interface; co-operative processing TCP/IP interface; tools for writing APIs; TIME facility                                                                                                  |
|                                                            | APL2 for Windows version 1.0   | \$1500     | Product No. 5639-d46, part number 4229558.                                                                                                                                                                                                                          |
|                                                            | APL2 for Sun Solaris           | \$1500     | Product No. 5648-065.                                                                                                                                                                                                                                               |
|                                                            | APL2 for AIX 6000              | poa        | Product No. 5785-012.                                                                                                                                                                                                                                               |
|                                                            | APL2 Version 2                 | poa        | Product No. 5688-228. Full APL2 system for S/370 and S/390                                                                                                                                                                                                          |
|                                                            | APL2 Application Env't Vn2     | poa        | Product No. 5688-229. Runtime environment for APL2 packages                                                                                                                                                                                                         |
| Insight Systems                                            | APL*PLUS/PC                    | poa        | APL systems marketed and supported ...                                                                                                                                                                                                                              |
|                                                            | Dyalog APL                     | poa        | from: Dyadic, Manugistics, IBM                                                                                                                                                                                                                                      |
|                                                            | APL2                           | poa        | under: Windows, OS2 and Unix                                                                                                                                                                                                                                        |
| Iverson Software Inc. J on the Web online registration ... |                                |            |                                                                                                                                                                                                                                                                     |
|                                                            | J Educational Edition          | \$50       |                                                                                                                                                                                                                                                                     |
|                                                            | J Standard Edition             | \$150      |                                                                                                                                                                                                                                                                     |
|                                                            | J Professional                 | \$325      |                                                                                                                                                                                                                                                                     |
| Books and accessories (discounts for reg users)            |                                |            |                                                                                                                                                                                                                                                                     |
|                                                            | J Dictionary                   | \$50       |                                                                                                                                                                                                                                                                     |

|                     |                                    |           |                                                                                                                                                                |
|---------------------|------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | J User Manual                      | \$50      |                                                                                                                                                                |
|                     | J Phrases                          | \$40      |                                                                                                                                                                |
|                     | J Primer                           | \$40      |                                                                                                                                                                |
|                     | Set of the above 4 books           | \$160     |                                                                                                                                                                |
|                     | Concrete Math                      | \$40      |                                                                                                                                                                |
|                     | Fractals, Visualization & J        | \$50      |                                                                                                                                                                |
|                     | Exploring Math                     | \$50      |                                                                                                                                                                |
|                     | J User Conference Proceedings      | \$35      |                                                                                                                                                                |
|                     | Mugs, T-shirts, Mousepads          | \$10 each |                                                                                                                                                                |
|                     | APLWIN                             | \$70      | For 386/PC under Windows 3.1                                                                                                                                   |
| J Austria           | J                                  | poa       | Distributor for Austria and Switzerland                                                                                                                        |
|                     | Dyalog APL                         | poa       | Distributor                                                                                                                                                    |
|                     | Causeway Products                  | poa       | Distributor                                                                                                                                                    |
|                     | Structural Analysis Software       | poa       | Complete package by IG Zenkner&Handel to perform structural analysis/engineering calculations. Also suitable for dynamic problems, e.g. earthquake simulation. |
| Lescasse Consulting | APL+PC                             | poa       | Lescasse Consulting is the exclusive APL2000 distributor in France and also                                                                                    |
|                     | APL+Unix                           | poa       | distributes in Switzerland and Belgium. Call for price quotes.                                                                                                 |
|                     | APL+DOS                            | poa       |                                                                                                                                                                |
|                     | APL+Win                            | poa       |                                                                                                                                                                |
|                     | Dyalog APL/W                       | poa       | French distributor for Dyalog                                                                                                                                  |
| MasterWork Software | Manugistics Products and ISI       | poa       | New Zealand distributor                                                                                                                                        |
| MicroAPL            | APL68000 Level I                   | 2000      | First generation APL with numerous enhancements. Multi-user version (Unix, Mirage, MCS).                                                                       |
|                     | APL68000 Level II                  | 2500      | Second generation APL. Nested arrays, user defined operators, selective specification etc. Multi-user version (Unix, Mirage, MCS)                              |
|                     | APL68000/X                         | 1500-6000 | Second-generation APL. Nested arrays, user defined operators, selective specification, etc. Multi-user AIX version with full OSF/Motif support.                |
|                     | APL68000 Level I<br>Mac, ST, Amiga | 87        | First generation APL. Single user, full windowing interface, software floating point support.                                                                  |
|                     | Mac, Amiga                         | 260       | First generation APL. Single user, full windowing interface, hardware floating point.                                                                          |
|                     | APL68000 Level II<br>ST            | 170       | Second generation APL. Full windowing interface, software floating point support.                                                                              |
|                     | Amiga                              | 260       | Second generation APL. Full windowing interface. Hardware and software floating point support.                                                                 |
|                     | Mac                                | 520       | Second generation APL. Full windowing interface. Hardware and software floating point support.                                                                 |
|                     | APL*PLUS Rel 10                    | 450       |                                                                                                                                                                |
|                     | APL*PLUS II V 4.0                  | 1395      |                                                                                                                                                                |
| Oasis Systems       | Dyalog APL                         | poa       | Dyadic Systems                                                                                                                                                 |
|                     | APL*PLUS                           | poa       | Manugistics                                                                                                                                                    |
|                     | APL68000                           | poa       | MicroAPL Ltd                                                                                                                                                   |
|                     | APL2                               | poa       | IBM                                                                                                                                                            |
| Optima              | Dyalog APL/W                       | 995       | Fully fledged Windows development environment.                                                                                                                 |

|                    |                                  |     |                                                                                       |
|--------------------|----------------------------------|-----|---------------------------------------------------------------------------------------|
| RE Time Tracker Oy | APL+PC (APL*PLUS/PC)             | poa | Complete APL+ and Statgraphics product range and links to various 3rd party products. |
|                    | APL+DOS (APL*PLUS II)            |     |                                                                                       |
|                    | APL+Win (APL*PLUS III), APL+Link |     |                                                                                       |
|                    | APL+UNIX                         |     |                                                                                       |
|                    | APL*PLUS Sharefile               |     |                                                                                       |
| Soliton Associates | SHARP APL for MVS                | poa | for IBM MVS mainframes                                                                |
|                    | SHARP APL for Unix               | poa | for IBM RS/6000 and Sun SPARC                                                         |
| Strand Software    | Canada                           |     |                                                                                       |
|                    | All APL*PLUS Products            | poa | All APL*PLUS products including upgrades and educational.                             |
|                    | Dyadic and ISI products          | poa |                                                                                       |
|                    | USA                              |     |                                                                                       |
|                    | Dyadic and ISI products          | poa |                                                                                       |

## APL PACKAGES

| COMPANY                                        | PRODUCT                                                                                                                  | PRICES(£)   | DETAILS                                                                                                                                                                                                                             |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADAPTA Software                                | MPS - Master Production Scheduling<br>FBS - Forecasting and Budgeting System<br>DRP - Distribution Requirements Planning | pos         |                                                                                                                                                                                                                                     |
| Adaptable Systems                              | FLAIR                                                                                                                    | poa         | Finite loader and interactive rescheduler. Customisable full-function scheduling system. (Available outside Australia by special arrangement only.)                                                                                 |
| Adaytum Software                               | Adaytum Planning                                                                                                         | poa         | Full-featured Budgeting and Financial Planning system for medium to large enterprises.                                                                                                                                              |
| The APL Group                                  | Qualed                                                                                                                   | \$1500-4000 | Electronic Data Interchange (EDI) translation software for the PC, with strict compliance checking.                                                                                                                                 |
| APL Software Ltd<br>(mainframe)                | RDS                                                                                                                      | poa         | Relation Data Base System                                                                                                                                                                                                           |
|                                                | IPLS                                                                                                                     | poa         | Project Management System                                                                                                                                                                                                           |
|                                                | REGGPAK                                                                                                                  | poa         | Regression Analysis Package                                                                                                                                                                                                         |
| (microcomputer)                                | POWERTOOLS                                                                                                               | 295         | Assembler written replacement function for commonly used CPU-consuming APL functions, includes a Forms Processor.                                                                                                                   |
|                                                | REGGPAK                                                                                                                  | poa         | Regression Analysis Package                                                                                                                                                                                                         |
|                                                | RDS                                                                                                                      | 990         | Relational Database System                                                                                                                                                                                                          |
| Beautiful Systems                              | ASF_FILE                                                                                                                 | \$399       | Dyalog APL/W auxiliary processor for access to APL*PLUS/PC APL component files (*.ASF).                                                                                                                                             |
|                                                | NAT_FILE                                                                                                                 | \$299       | Dyalog APL/W auxiliary processor which emulates the APL*PLUS/PC quad-N native file subsystem for access to the DOS file system.                                                                                                     |
|                                                | DBF_FILE                                                                                                                 | \$299       | Dyalog APL/W auxiliary processor for efficient block mode access to dBASE format files. Designed to get large amounts of data in and out of dBASE. Not suited for random access to small amounts of data (it does not handle keys). |
|                                                | SF_READ                                                                                                                  | poa         | Dyalog APL/W functions to read APL*PLUS data objects of any type or structure from *.SF style component files created by APL*PLUS II or III.                                                                                        |
| The Bloomsbury Software Company<br>(for VSAPL) | Enhancements & Sharefile                                                                                                 | poa         | Component files, quad-functions & nested arrays for VSAPL under VM/CMS & MVS/TSO                                                                                                                                                    |
|                                                | Compiler                                                                                                                 | poa         | The First APL compiler!                                                                                                                                                                                                             |

|                      |                                             |           |                                                                                                                                                                                                                                                                                  |
|----------------------|---------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (for APL2)           | Sharefile/AP                                | poa       | STSC's shared access component file system for APL2. Comparable to all APL*PLUS file systems: multi-user storage of APL2 arrays with efficient disk usage.                                                                                                                       |
| Causeway             | CausewayPro for Dyalog/W                    | 400/\$600 | Causeway application development platform for Dyalog APL/W.                                                                                                                                                                                                                      |
|                      | RainPro Business Graphics                   | 250       | The ultimate graphics toolkit for the APL developer. Adds 3D charting capability, Web publishing and clipboard support to the shareware product. Charts can be included in <i>NewLeaf</i> reports. Functionally compatible across Dyalog/W and APL+Win.                          |
|                      | NewLeaf for Dyalog and +Win                 | 400       | Frame-based reporting tool with comprehensive table-generation and text-flow support. Offers multiple master-page capability, bitmap wrap-around and on-screen preview with pan and zoom. Fully supported on Dyalog/W and APL+Win (1.8 and above).                               |
| Cinerea AB           | ORCHART                                     | 250       | Organization chart package for IBM APL2/PC. Full & heavily commented source code included - free integration into other applications. NB: ASCII output with line-drawing (semi-graphic) characters for boxes.                                                                    |
| CODEWORK             | HELM                                        | poa       | Decision Support system for top management. Handles large multi-dimensional tables, data analysis, EIS presentations; generates HTML and Latex output. Platforms: DOS, APL+II, Windows 3.1/95 for Dyalog APL, LAN support. Ideal for APL customisation, more than 100 installed. |
| H.M.W.               | 4XTRA                                       | poa       | Front-end Foreign Exchange dealing / pos keeping                                                                                                                                                                                                                                 |
|                      | Arbitrage                                   | poa       | Arbitrage modelling                                                                                                                                                                                                                                                              |
|                      | Basket                                      | poa       | Basket currency modelling                                                                                                                                                                                                                                                        |
|                      | Menu-Bar                                    | poa       | pull-down menu for APL*PLUS/PC                                                                                                                                                                                                                                                   |
| HRH Systems          | APL Utilities                               | poa       | Software to transfer workspaces between APL*PLUS and Sharp, and between APL*PLUS and I-APL. Software to import IBM .ATF files to APL*PLUS.                                                                                                                                       |
|                      | APL*PLUS Utilities                          |           | Public domain software, unlock locked fns, a user-friendly alternative to locking, fns of mathematical physics, menus, and others.                                                                                                                                               |
| IAC/Human Interfaces | SPARKS                                      | poa       | Educational simulation of electric circuit (for Apple Mac.)                                                                                                                                                                                                                      |
|                      | EPIDEMIC                                    | poa       | Educational simulation of spreading infection (for Apple Mac.)                                                                                                                                                                                                                   |
|                      | COINS                                       | poa       | Educational simulation (KS3) of coin-tossing experiment with simple stats (for Apple Mac.)                                                                                                                                                                                       |
|                      | FIBONA                                      | poa       | Educational simulation of Fibonacci's rabbits (for Apple Mac.)                                                                                                                                                                                                                   |
| I-APL Ltd            | Educational workspaces                      | 5         | PC format disks with the examples from: Thomson, Espinasse (Kits 1-4), Kromberg, Jizba & FinnAPL. All the examples to save your fingers!                                                                                                                                         |
| IBM APL Products     | A Graphical Statistical System              | \$250     | for DOS, Product Number 5764-009                                                                                                                                                                                                                                                 |
|                      | (AGSS)                                      | \$500     | for Workstations (OS/2, Aix, Solaris), Product Number 6764-092                                                                                                                                                                                                                   |
|                      |                                             | \$2500    | for CMS, Product Number 5764-011                                                                                                                                                                                                                                                 |
| INFOSTROY            | APL*PLUS/Xbase Interface (II/386 Version 2) | \$198     | Complete package written in C. Comparable with the data, index & memo files of FoxPro, dBASE, & Clipper. Multi-user support. No DBMS license required.                                                                                                                           |
|                      | (DLL Version 1)                             | \$198     | The same in a DLL form! Gives your Windows applications all advantages of DLLs.                                                                                                                                                                                                  |
| Insight Systems      | IUTILS/XP                                   | 20-95     | Cross-platform utility library including simple OS calls (DIR, COPY, DEL, RENAME) and DATE functions. For APL*PLUS II, APL2 and Dyalog APL under Windows, OS/2 and Unix.                                                                                                         |
|                      | ASI                                         | 95        | APL Spreadsheet Interface. "Device-Independent" spreadsheet driver supporting Excel, 123 and Quattro-Pro for Dyalog APL/W                                                                                                                                                        |
|                      | WinCom                                      | 95        | Asynchronous comms package for Dyalog APL/W                                                                                                                                                                                                                                      |
|                      | S2D,22D,X2X                                 | poa       | Advanced APL syntax analysis and conversion packages from Sharp and APL2 to Dyalog, and between any two APLs                                                                                                                                                                     |

|                     |                                  |         |                                                                                                                                                                                               |
|---------------------|----------------------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | SQAPL Client                     | poa     | Interface from APL*PLUS II, APL2 and Dyalog (Windows, OS/2 or Unix) to most SQL databases over most networks.                                                                                 |
|                     | SQAPL Server                     | poa     | Makes APL*PLUS II, APL2 or Dyalog APL (Unix) available as SequelLink servers. Can be called from SQAPL clients or other applications such as Excel, C++, Smalltalk, Visual Basic.             |
| JAD Software        | JAD SMS                          | 150-500 | Software management system for APL*PLUS II based on hierarchical databases; includes full-screen interface and stand-alone functions. Price depends on number of users.                       |
| Lescasse Consulting | APL+Win Monthly Training Program | \$500   | Download 50+ page document about APL+ programming each month. You also get one or more workspaces full of re-usable APL code and sometimes additional files or products.                      |
|                     | Advanced Windows Programming ... | \$95    | 200-page book plus companion disk on interfacing APL and Delphi. Contains full coverage of Delphi-2, +Win and Dyalog.                                                                         |
|                     | DLL parser for +Win              | \$250   | Parse any Visual Basic DLL declaration file into a set of quadNA definitions. Turn constants and structures into APL variables. Available for APL+Win and Dyalog/W.                           |
|                     | Delphi Forms Translator          | \$195   | Design forms with Delphi and turn them automatically into APL programs which recreate the same form (+Win and Dyalog/W).                                                                      |
|                     | APL+Link Pro                     | poa     | ODBC interface for APL+Win                                                                                                                                                                    |
|                     | SQAPL Pro                        | poa     | ODBC interface for Dyalog APL/W                                                                                                                                                               |
|                     | RainPro                          | poa     | Highly customisable 2D and 3D publication graphics for APL+Win and Dyalog APL/W                                                                                                               |
|                     | NewLeaf                          | poa     | Page layout and printing tools for APL+Win and Dyalog                                                                                                                                         |
|                     | GraphX and ChartFX               | poa     | High-quality business graphics for APL+Win                                                                                                                                                    |
|                     | Formula One and Dyalog APL       | \$95    | 100-page book + companion disk on how to use the Formula One VBX with Dyalog APL/W                                                                                                            |
| Lingo Allegro       | FRESCO Business Graphics         | poa     | Fast and easy business graphics DLL                                                                                                                                                           |
|                     | AP126/PC                         | poa     | GDDM interface for Dyalog APL/W                                                                                                                                                               |
|                     | AP127/PC                         | poa     | ODBC interface for Dyalog APL/W                                                                                                                                                               |
|                     | AP119/PC                         | poa     | TCP/IP interface for Dyalog APL/W                                                                                                                                                             |
|                     | FACS                             | poa     | EMMA-like interface to DB2 or ODBC databases                                                                                                                                                  |
|                     | ISAP                             | poa     | MS Windows DLL for calling Dyalog APL/W from web pages via MS Internet Information Server. Visit <a href="http://www.lingo.com">www.lingo.com</a> for a demo.                                 |
| RE Time Tracker Oy  | UIT/W                            | poa     | Comprehensive high-level Windows User Interface library for APL+Win and +II v 5.1. Comprehensive spreadsheets, replicated fields, special field types, etc. 16 and 32 bit versions available. |
|                     | AJGRAPH                          | poa     | Graphpak-compatible 2D graphics package for +Win and +DOS. Includes multi-window support, print and metafile support. No DLLs required.                                                       |
|                     | ECCO PRO with APL                | poa     | Leading group and personal information management system with comprehensive customising. Supplied with sample +Win workspace to interface to ECCO databases via DDE.                          |
|                     | NEWT TCP/IP SDK with APL         | poa     | Lead TCP/IP SDK with interfaces to all protocols. Supplied on 3 CD ROMs together with a sample +Win workspace.                                                                                |
|                     | DB+                              | poa     | Database interface for APL+DOS under Windows. Allows combining character-based APL applications with ODBC-compliant databases such as Oracle and SQL-server.                                  |
| Soliton Associates  | LOGOS                            | poa     | Application Development Environment                                                                                                                                                           |
|                     | MAILBOX                          | poa     | Electronic Mail                                                                                                                                                                               |
|                     | VIEWPOINT                        | poa     | Report generator with interfaces to DB2 and MVS data                                                                                                                                          |
| Warwick University  | BATS                             | 250     | Menu driven system for time series analysis and forecasting using Bayesian Dynamic modelling. Price is reduced to £35 for academic institutions.                                              |
|                     | FAB                              | free    | Training program for the above.                                                                                                                                                               |

|      |                             |                    |                                                                                                  |
|------|-----------------------------|--------------------|--------------------------------------------------------------------------------------------------|
| Zark | APL Tutor (PC)              | \$299              | APL computer-based training. Available for APL*PLUS PC & APL*PLUS II. Demo disk \$10.            |
|      | APL Tutor (MF)              | \$5000             | Mainframe version.                                                                               |
|      | Zark ACE                    | \$99               | APL continuing education. APL tutor news and hotline phone support.                              |
|      | APL Advanced Techniques.... | \$59.95            | 488pp. book, (ISBN 0-9619067-07) Including 2-disk set of utility functions (APL*PLUS PC format). |
|      | Communications              | \$200 pc, \$500 mf | Move workspaces or files between APL environments.                                               |

## APL CONSULTANCY AND DEVELOPMENT

| COMPANY              | PRODUCT                  | PRICES (£)  | DETAILS                                                                                                                                                                                          |
|----------------------|--------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adfee                | Consultancy              | poa         | Development, maintenance, conversion, migration, documentation, of APL products in all APL environments                                                                                          |
| Ajay Askoolom        | Consultancy              | poa         | APL+Win development and migration of actuarial, financial, mathematical applications.                                                                                                            |
| Andrews              | Consultancy              | poa         | APL programming and analysis, Year-2000 legacy systems, algorithms, tree-processing.                                                                                                             |
| APL Solutions Inc    | Consultancy              | poa         | APL systems design, development, maintenance, documentation, testing and training. Providing APL solutions since 1969.                                                                           |
| AUSCAN Software      | Consultancy              | poa         | APL software development, training                                                                                                                                                               |
| Bloomsbury Software  | Consultancy              | 300-750+VAT |                                                                                                                                                                                                  |
| Camacho              | Consultancy              | poa         | Manuals; feasibility reports and estimates; analysis and programming; APL and MS Windows applications; Sharp, (Si APL, APL*PLUS, APL2/PC and other APLs spoken. Fixed price systems a speciality |
| Ray Cannon           | Consultancy              | poa         | APL, C, Assembler, Windows, Graphics: PC and mainframe                                                                                                                                           |
| Causeway             | Consultancy and Training | poa         | On-site training for Causeway, RainPro and NewLeaf. Customisation and enhancement to meet local needs. Code review and pre-implementation check of Causeway applications.                        |
| Paul Chapman         | Consultancy              | 250-500     | 24-hour programmer: APL, Smalltalk, C; Windows front end design a speciality.                                                                                                                    |
| CODEWORK             | Consultancy              | poa         | Development, maintenance, migration, documentation of APL applications. Speciality: info systems for top executives, internet applications.                                                      |
| Dinosoft Oy          | Consultancy              | poa         | Specialised in very large databases.                                                                                                                                                             |
| Dyadic               | Consultancy              | poa         | APL and Unix system design, consultancy, programming and training.                                                                                                                               |
| Evestic AB           | Consultancy              | poa         | Excellent track record from 15+ years of APL applications in banking, insurance, and education services. All dialects, platforms and project phases. SQL expertise.                              |
| General Software     | Consultancy              | from 120    |                                                                                                                                                                                                  |
| Godin London Inc     | Software Development     | poa         | We have applications in the food manufacturing field, travel agency and airline bookings field and in product lease management.                                                                  |
| Entropy Software Ltd | Consulting               | poa         | Company reporting, business graphics, Windows applications with Dyalog APL/W.                                                                                                                    |
| H.M.W.               | Consultancy              | poa         | System design consultancy, programming. HMW specialize in banking and prototyping work.                                                                                                          |
| Hoekstra Systems Ltd | Consultancy              | poa         | APL consultancy, programming, etc. Also UNIX system administration                                                                                                                               |
| Michael Hughes       | Consultancy              | poa         | Consultant with 10+ years experience with various APL interpreters and C.                                                                                                                        |

|                                  |               |                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                           |
|----------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IAC/Human Interfaces Consultancy |               | poa                                                                                                                                                                                                                                 | APL on Macintosh & PC. HCI design. VDU ergonomics: EC/Health & Safety compliance.                                                                                                                                                                                                         |
|                                  | Documentation | 100-200                                                                                                                                                                                                                             | On-line assistance, product demos & mock-ups, manual writing; foreign language software localization.                                                                                                                                                                                     |
|                                  | Training      | poa                                                                                                                                                                                                                                 | Using I-APL for courseware & distance learning materials; Mac programming in C, APL & HyperCard.                                                                                                                                                                                          |
| INFOSTROY                        | Consultancy   | poa                                                                                                                                                                                                                                 | Moving applications between platforms. Client/server development. Multilingual user interface.                                                                                                                                                                                            |
| Insight Systems                  | Consultancy   | poa                                                                                                                                                                                                                                 | Experts in APL conversions between any combination of: APL*PLUS, APL2, Dyalog APL and Sharp APL. We are also experienced right-sizers, comfortable with networks and relational databases (that also means when NOT to use SQL) and client/server development in APL, C and Visual Basic. |
| JAD Software                     | Consultancy   | poa                                                                                                                                                                                                                                 | Systems design and development, project management, technical manuals, financial and actuarial expertise in APL.                                                                                                                                                                          |
| Phil Last                        | Consultancy   | poa                                                                                                                                                                                                                                 | APL consultancy, modelling and programming.                                                                                                                                                                                                                                               |
| Lescasse Consulting              | Consultancy   | poa                                                                                                                                                                                                                                 | A range of consultants, experts in Windows programming, with APL+Win and Dyalog APL/W. More than 100 major APL applications already developed. We all have additional expertise in Formula One and Delphi.                                                                                |
| Lingo Allegro                    | Consultancy   | poa                                                                                                                                                                                                                                 | General APL consulting, internet website development, migration and downsizing, performance tuning, education and training.                                                                                                                                                               |
| Mackay Kinloch Ltd               | Consultancy   | 200-400                                                                                                                                                                                                                             | Design, analysis and programming for banking, insurance, financial planning and modelling, corporate performance and legal reporting                                                                                                                                                      |
| MicroAPL                         | Consultancy   | poa                                                                                                                                                                                                                                 | Technical & applications consultancy.                                                                                                                                                                                                                                                     |
| Milinta Inc                      | poa           | Design, development, maintenance, conversion, documentation in all APLs, most APs and some specific Sharp products (LOGOS, ViewPoint, Retrieve). Experience in multi-user, multi-task systems, databases, Y2K, Windows programming. |                                                                                                                                                                                                                                                                                           |
| Ellis Morgan                     | Consultancy   | 250-500                                                                                                                                                                                                                             | Business Forecasting & APL Systems.                                                                                                                                                                                                                                                       |
| Oasis Systems                    | Consultancy   | poa                                                                                                                                                                                                                                 | Expertise in APL system design, Project management, conversion, migration, tuning; for all APL versions (10+ years experience)                                                                                                                                                            |
| Object Oriented Ltd              | Consultancy   | poa                                                                                                                                                                                                                                 | General APL consulting, code recycling — mainframe to PC, performance tuning.                                                                                                                                                                                                             |
| Optima                           | Consultancy   | poa                                                                                                                                                                                                                                 | A range of consultants specialising in all areas of pharmaceutical, industrial and financial systems with 5-15 yrs experience on both PC and mainframe.                                                                                                                                   |
| RadSys Technologies              | Consultancy   | poa                                                                                                                                                                                                                                 | Areas of expertise: financial systems, risk analysis systems, healthcare systems.                                                                                                                                                                                                         |
| RE Time Tracker Oy               | Consultancy   | poa                                                                                                                                                                                                                                 | APL application conversions, APL Windows interfaces, APL to APL-level interfacing to any system under Windows, TCP/IP network and database connectivity.                                                                                                                                  |
| Rex Swain                        | Consultancy   | poa                                                                                                                                                                                                                                 | Independent consultant, 20 years experience. Custom software development & training, PC and/or mainframe.                                                                                                                                                                                 |
| Rochester Group                  | Consultancy   | poa                                                                                                                                                                                                                                 | Specialise in MIS using Sharp APL                                                                                                                                                                                                                                                         |
| Shepp & Associates               | Consultancy   | poa                                                                                                                                                                                                                                 | APL applications development and consulting, especially in the travel industry, especially on small computers. 25 years experience in APL programming.                                                                                                                                    |
| Snake Island Research Inc        | Consultancy   | poa                                                                                                                                                                                                                                 | APL interpreter and compiler enhancements, intrinsic functions, performance consulting. APL parallel compiler APEX is giving very good initial performance tests with convolution somewhat faster than FORTRAN.                                                                           |

|                   |             |     |                                                                                                                                                                                                                                                             |
|-------------------|-------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Strand Software   | Consultancy | poa | Advice on migrating to and from all flavours of APL and hardware platforms. Full-screen interface implementation, APL utilities, benchmarking, efficiency analysis, actuarial software, system development tools, valuation, pricing and modelling systems. |
| Sykes Systems Inc | Consultancy | poa | Complete APL services specialising in audit, optimisation and conversion of APL systems. Excellent design skills. All dialects and platforms. 17-23 years experience.                                                                                       |
| Stephen Wynn      | Consultancy | poa | Most experience of financial planning, and mathematical areas: operational research, quality control, experimental design.                                                                                                                                  |

## OTHER PRODUCTS

| COMPANY                  | PRODUCT                                                      | PRICES (£)  | DETAILS                                                                                                                                                                         |
|--------------------------|--------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adfee                    | Employment                                                   | poa         | Contractors and permanent employees                                                                                                                                             |
| APL-385                  | Typefaces                                                    | poa         | Variants of the APL2741 typeface available to specification.                                                                                                                    |
| Bloomsbury Software      | Training                                                     | poa         | Contact the company for details.                                                                                                                                                |
| ComLog                   | Comic-Logger                                                 | \$25.95+p&p | APL*PLUS II comic-book inventory system. Shareware version available on America OnLine.                                                                                         |
| HMW                      | Employment                                                   | poa         | Contractors and permanent employees placed.                                                                                                                                     |
| HRH Systems              | APL lessons                                                  |             | On-screen interactive APL lessons for APL*PLUS, TryAPL2, Sharp and I-APL — in English or French.                                                                                |
|                          | The BBS/APL                                                  | n/a         | <i>The BBS/APL is no longer available.</i> For an email catalogue of its archive files (and how to get them) email <a href="mailto:clck.hoit@juno.com">clck.hoit@juno.com</a> . |
| I-APL Ltd                | An APL Tutorial                                              | 3           | 45pp by Alvord & Thomson                                                                                                                                                        |
|                          | An Encyclopaedia of APL (2d Ed)                              | 6           | 228pp by Helzer                                                                                                                                                                 |
|                          | APL in Social Studies                                        | 3           | 36pp by Traberman                                                                                                                                                               |
|                          | I-APL Instruction Manual (2d Ed)                             | 3           | 55pp by Camacho & Ziemann                                                                                                                                                       |
|                          | APL Programs for the Mathematics Classroom (Springer-Verlag) | 16          | 185pp by Thomson                                                                                                                                                                |
|                          | Programming in J                                             | 10          | 75pp by Ken Iverson                                                                                                                                                             |
|                          | Aritmetic                                                    | 12          | 118pp by Ken Iverson                                                                                                                                                            |
|                          | Tangible math                                                | 8           | 36pp by Ken Iverson                                                                                                                                                             |
|                          | Sharp APL Reference Manual                                   | 42          | 349pp by Berry                                                                                                                                                                  |
|                          | APL Press Books                                              | poa         | A comprehensive selection of early APL literature                                                                                                                               |
|                          | <i>Please note there is a packing charge of £3 per order</i> |             |                                                                                                                                                                                 |
| Oasis Systems            | Training                                                     | poa         | introductory courses in APL<br>Advanced courses for different APL versions                                                                                                      |
| Renaissance Data Systems | Booksellers                                                  |             | The widest range of APL books available anywhere. See Vector advertisements.                                                                                                    |
| Soliton Associates       | MVSLINK                                                      | poa         | Interface from Sharp APL (Unix & MVS) to non-APL data and software in the MVS environment.                                                                                      |
|                          | SSQL                                                         | poa         | High-performance DB2 interface for Sharp APL (Unix and MVS).                                                                                                                    |

## OVERSEAS ASSOCIATIONS

| GROUP                                           | LOCATION          | JOURNAL             | OTHER SERVICES                       | Ann.Sub.                         |
|-------------------------------------------------|-------------------|---------------------|--------------------------------------|----------------------------------|
| APL Bay Area                                    | USA N. California | APLBUG              | Monthly Meetings (2nd Monday)        | \$20                             |
| APL Club Austria                                | Austria           |                     | Quarterly Meetings                   | 200AS(indiv), 1000AS(corp)       |
| APL Club Germany                                | Germany           | APL Journal         | Semi-annual meetings                 | DM60                             |
| Ass. Francophone pour la promotion d'APL France |                   | Les Nouvelles d'APL |                                      | FF350 (private) FF2800 (Company) |
| BACUS                                           | Belgium           | APL-CAM             | Conferences & Seminars               | £18 (\$30)                       |
| Capital PCUG                                    | Washington, D.C.  | Monitor             | Monthly meetings, occasional classes | free                             |

|                  |                   |                      |                                                               |                                         |
|------------------|-------------------|----------------------|---------------------------------------------------------------|-----------------------------------------|
| Danish SIG       | Denmark           |                      |                                                               |                                         |
| Dutch APL Assoc. | Holland           | Vector provided      | Mini-congress, APL ShareWare Initiative                       |                                         |
| FinnAPL          | Helsinki, Finland | FinnAPL Newsletter   | Seminars on APL                                               | 100FRM(private), 30(student), 1000 (Co) |
| Japan APL Assoc  | Tokyo             | APL Journal          | Monthly meetings (4th Sat)                                    | 10,000yen to join                       |
| Rome/Italy SIG   | Roma, Italy       |                      |                                                               |                                         |
| RusAPL           | Moscow, Russia    | APL Club             | Seminars and Annual Meeting                                   | 100,000R (students 20,000)              |
| SE APL Users Grp | Atlanta, Georgia  | SEAPL Newsletter     | Quarterly meetings                                            | \$10                                    |
| SovAPL           | Obninsk, Russia   |                      |                                                               |                                         |
| SwedAPL          | Sweden            | SwedAPL Nytt         | Semi-annual meetings, seminars                                | SEK 75                                  |
| SWAPL            | Texas, USA        | SWAPL                |                                                               | \$18                                    |
| Swiss APL (SAUG) | Bern              | Part of Qlty SI-Info |                                                               | SF60 (SI) + SF20 (SAUG)                 |
| Toronto SIG      | Toronto, Canada   | Glinrne Arrays!      | Monthly Meetings, APL skills database, J SIG, Toronto Toolkit | \$25                                    |

## ADDRESSES

## ORGANISATION CONTACT

## ADDRESS, TELEPHONE, FAX, EMAIL etc.

|                                                 |                     |                                                                                                                                       |
|-------------------------------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| ADAPTA Software GmbH                            | Michael Baas        | Marienhoehe 86, 25451 Quickborn, Germany Tel: +49 4106 60977<br>Fax: +49 4106 67869 Email: 101523.1757@compuserve.com                 |
| Adaptable Systems                               | Lois & Richard Hill | 49 First Street, Black Rock 3193, Australia.<br>Tel: +61 3 9589 5578 Fax: +61 3 9589 3220 Email: adsys@ibm.net                        |
| Adatum Software                                 | Douglas Howley      | 13 Great George Street, BRISTOL BS1 5RR UK Tel: 0117-921 5555                                                                         |
| Adfee                                           | Bernard Smoor       | Dorpsstraat 50, 4128 BZ Lexmond, Netherlands.<br>Tel: +31 347 342 337 Fax: +31 347 342 342 Email: adfee@concepts.nl                   |
| Ajay Askoolum                                   | Ajay Askoolum       | 42 Hanworth Road, Redhill, Surrey RH1 5HT<br>Tel: 01737 771643 Email: 106173.3347@compuserve.com                                      |
| Andrews                                         | Dr Anne D Wilson    | 12 Thorny Hills, Kendal, Cumbria LA9 7AL, UK Tel: 01539-731205                                                                        |
| APL-385                                         | Adrian Smith        | Brook House, Gilling East, York YO62 4JJ UK. Tel: 01439-788395<br>Fax: 01439-788194 Email: 100331.644@compuserve.com                  |
| APL Bay Area APLBUG                             | Curtis Jones (Sec)  | 228 South 15th Street, San Jose, CA 95112-2150, USA<br>Tel: +1 (408) 292-4060 Email: jonesca@vnet.ibm.com                             |
| APL Club Austria                                | Harald F. Nelson    | c/o N-TECH, Siebenbrunnfeldg. 4-6, A-1050 Wien, Austria.<br>Tel: +43 1 5458033 Fax: +43 1 5458063-17                                  |
| APL Club Germany                                | Dieter Lattemann    | Rheinstraße 23, D-69190 Walldorf, Germany.<br>Tel: +49 6227-63489 Compuserve: 100332.1461                                             |
| The APL Group Inc                               | Stuart Sawabini     | 644 Danbury Road, WILTON, CT 06897 USA. Tel: +1 (203) 762-3933<br>Fax: +1 (203) 762-2108 Email: ssawabini@aol.com, eshaw@aplgroup.com |
| APL Solutions Inc                               | Eric Landau         | 1107 Dale Drive, Silver Spring, MD 20910-1607 USA<br>Tel: +1 (301) 589-4621 Fax: +1 (301) 589-4618 Email: elandau@cais.com            |
| Association Francophone pour la promotion d'APL | Ludmila Lemagnen    | 174 Boulevard de Charonne, F-75020 Paris, FRANCE<br>Email: lemagnen@aol.com                                                           |
| AUSCAN Software Ltd                             | Richard Procter     | PO Box 39, Mansfield, Ontario L0N 1M0 Canada<br>Tel: +1-705-434-1239 Email: rjp@interlog.com                                          |
| BACUS                                           | Joseph De Kerf      | Roolnberg 72, B-2570 Duffel, Belgium. Tel: +32 15 31 47 24                                                                            |
| Beautiful Systems, Inc.                         | Jim Goff            | 308 Old York Road, Suite 5, Jenkintown, PA 19046, USA<br>Tel: +1 (215) 886-2836; Fax: +1 (215) 886-4888                               |
| Bloomsbury Software                             | Peter Day           | 3-6 Alfred Place, Bloomsbury, London WC1E 7EB UK. Tel: 0171-436 9481<br>Fax: 0171-436 0524 Email: pd@bloomsbury-software.co.uk        |
| Camacho                                         | Anthony Camacho     | 11 Auburn Road, Redland, Bristol BS6 6LS UK. Tel: 0117-9730036.<br>email: acamacho@dx.compulink.co.uk Reuternet (Sharp): ACAM         |
| Ray Cannon                                      |                     | 21 Woodbridge Rd, Blackwater, Camberley, Surrey GU17 0BS UK<br>Tel: 01252-874897 Email: 100430.740@compuserve.com                     |
| Paul Chapman                                    |                     | 51B Lambs Conduit Street, London WC1N 3NB UK.<br>Tel: 0171-404 5401. Compuserve: 100343.3210                                          |
| Causeway Graphics                               | Adrian Smith        | The Maltings, Castlegate, MALTON, North Yorks YO17 7DP UK                                                                             |

|                           |                                                                                                      |                                                                                                                                                                                                                     |
|---------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Systems Ltd               |                                                                                                      | Tel: 01653-696760 Fax: 01653-697719<br>Email: causeway@compuserve.com                                                                                                                                               |
| Cinerea AB                | Rolf Kornemark                                                                                       | Box 61, S-193 00 Sigtuna, Sweden.<br>Tel/Fax: +46 859 255 421 Email: rolf@cinerea.se                                                                                                                                |
| CODEWORK                  | Mauro Guazzo                                                                                         | Corso Cairoli 32, 10123 Torino, Italy.<br>Tel: +39 11 885168 Fax: +39 11 812 2652 Email: codework@lnrete.it                                                                                                         |
| ComLog Software           | Jeff Pedneau                                                                                         | PO Box 5570, Derwood, MD 20855 USA<br>Tel: +1 (301) 990-7063 Email: jeff@softmed.com                                                                                                                                |
| CPCUG                     | Lynne Sturtz                                                                                         | Capital PC User Group, 51 Monroe Street, Suite PE-2, Rockville, Maryland 20850-2421, USA. Tel: +1 (301) 762-9372 Fax: (301) 762-9375.                                                                               |
| Danish User Group         | Per Gjerlov                                                                                          | Email: gjerlov@ibm.net                                                                                                                                                                                              |
| Dinosoft Oy               | Pertti Kalliojärvi                                                                                   | Lönnrotinkatu 21C, 00120 Helsinki, FINLAND.<br>Tel: +358 9 70028820 Fax: +358 9 70028824 Email: dinosoft@dinosoft.fi                                                                                                |
| Dutch APL Association     | Bernard Smoor (Sec)                                                                                  | Postbus 1341, 3430BH Nieuwegein, Netherlands.<br>Tel: +31 347 342 337 Fax: +31 347 342 342                                                                                                                          |
| Dyadic Systems Ltd.       | Peter Donnelly                                                                                       | Riverside View, Basing Road, Old Basing, Basingstoke, Hants RG24 0AL UK. Tel: 01256-811125 Fax: 01256-811130                                                                                                        |
| Entropy Software Ltd      | George MacLeod                                                                                       | 37 Newhouse Rd, Bovingdon, Herts, HP3 0EJ UK<br>Tel: 01442-834015 Email: 100412.1305@compuserve.com                                                                                                                 |
| Evestic AB                | Olle Evero                                                                                           | Bertellusvagen 12A, S-146 38 Tullinge, Sweden<br>Tel&Fax: +46 778 4410 Email: olle.evero@mailbox.swipnet.se                                                                                                         |
| FinnAPL                   | Olli Paavola                                                                                         | Suomen APL-Yhdistys RY, FinnAPL RF, PL 1005, 00101 Helsinki 10, Finland Email: olli.paavola@pyt.fi                                                                                                                  |
| General Software Ltd      | M.E. Martin                                                                                          | 22 Russell Road, Northolt, Middx, UB5 4QS UK. Tel/fax: 0181-864 9537                                                                                                                                                |
| Godin London Incorporated | Gaëtan Godin                                                                                         | 12 Gerrard St., London, Ontario, Canada N6C 4C5<br>Tel: +1 (519) 679-8290 Fax: +1 (519) 438-6381 Email: info@godin.on.ca                                                                                            |
| H.M.W.Trading Systems Ltd |                                                                                                      | Hamilton House, 1 Temple Avenue, Victoria Embankment, London EC4Y 0HA UK. Tel: 0171-353 9900; Fax: 0171-353 3325; Email: 100020.2632@compuserve.com                                                                 |
| Hoekstra Systems Ltd      | Bob Hoekstra                                                                                         | 5 Thorsden Court, Guildford Road, Woking, Surrey, GU22 7QS UK<br>Tel: 01483-771028 Email: bob.hoekstra@khamlin.demon.co.uk                                                                                          |
| HRH Systems               | Dick Holt                                                                                            | 3802 N Richmond St, Suite 271, Arlington, VA 22207 USA<br>Tel: +1 (703) 528-7624; Fax: +1 (703) 528-7617; Email: dick.holt@juno.org                                                                                 |
| Michael Hughes            |                                                                                                      | 28 Rushton Road, Willbarston, Market Harborough, Leics. LE16 8QL UK.<br>Tel: 01536-770998 Email: 101740.1203@compuserve.com                                                                                         |
| IAC/Human Interfaces      | Ian A. Clark                                                                                         | 9 Hill End, Frosterley, Bishop Auckland, Co. Durham DL13 2SX UK<br>Tel: 01388-526803. Compuserve: 100021.3073                                                                                                       |
| I-APL Ltd                 | Anthony Camacho<br>(for queries, order forms)<br>J C Business Services<br>(for pre-paid orders only) | 11 Auburn Road, Redland, Bristol BS6 6LS UK. Tel: 0117-9760036<br>email: acamacho@clx.compulink.co.uk Reuternet (Sharp): ACAM<br>56 The Crescent, Milton, Weston-super-Mare, Avon, BS22 8DU UK<br>Tel: 01934-625181 |
| IBM APL Products          | Nancy Wheeler                                                                                        | APL Products, IBM Santa Teresa, Dept M46/D12, 555 Bailey Avenue, San Jose CA 95141, USA. Tel: +1 (408) 463-APL2 (=2752)<br>Fax: +1 (408) 463-4488 Email: APL2@vnet.ibm.com Cserve: GO IBMAPL2                       |
| INFOSTROY                 | Alexei Miroshnikov                                                                                   | 3 S. TulenIn Lane, St. Petersburg 191186 Russia.<br>Tel:+7 812 312-2673 Fax:+7 812 311-2184 Email:aim@infostroy.spb.su                                                                                              |
| Insight Systems ApS       | Morten Kromberg                                                                                      | Nordre Strandvej 119C, DK-3150 Hellebæk, Denmark<br>Tel:+45 49 76 20 20 Fax: +45 49 76 20 30 Email: info@insight.dk                                                                                                 |
| Iverson Software Inc.     | Eric Iverson                                                                                         | 33 Major Street, Toronto, Ontario, Canada M5S 2K9 Tel: +1 (416) 925-6096; Fax: +1 (416) 488-7559 Email: info@software.com                                                                                           |
| J Austria                 | Joachim Hoffmann                                                                                     | c/o Causeway Graphical Systems. Email: JoHo@ping.at                                                                                                                                                                 |
| JAD Software              | David Crossley                                                                                       | 580 Eyer Drive, #81 Pickering, Ontario, Canada L1W 3B7<br>Tel: +1 (905) 837-1895 Fax: +1 (905) 831-5172                                                                                                             |
| Japan APL Assoc           | Toshio Nishikawa                                                                                     | 1-8-13 Masujima Buld.6F Higashi Gotanda Shinagawa-ku, Tokyo Japan<br>141-0022 Tel: +81 (03) 3260-0411 Fax: +81 (03) 3280-0418<br>Email: KY00381@niftyserve.org.jp                                                   |
| Phil Last Ltd             | Phil Last                                                                                            | 146 Crossbrook Street, Chestnut, Herts, EN8 8JY UK.<br>Tel: 01992 633807 Fax: 0121 359 0375 Email: phil_last@compuserve.com                                                                                         |
| Lescasse Consulting       | Eric Lescasse                                                                                        | 18 rue de la Belle Feuille, 92100 Boulogne, France Tel: +33.1.46.05.10.76<br>Fax: +33.1.46.04.60.23 Email: eric@lescasse.com                                                                                        |

|                           |                          |                                                                                                                                                                                                                                                  |
|---------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lingo Allegro USA, Inc    | Steven J Halasz          | 1105 Chicago Avenue, Suite 155, Oak Park, IL 60302, USA.<br>Tel: +1 800 546 4821-1 Email: sjhalasz@interaccess.com                                                                                                                               |
| Mackay Kinloch Ltd        | Alastair Kinloch         | 519 Webster's Land, Edinburgh EH1 2RX, Scotland, UK<br>Tel/Fax/Answerphone: 0131 228 3580 Pager/Voice: 01426 98 3858<br>Email: Alastair_Kinloch@compuserve.com                                                                                   |
| Mercia Software Ltd.      | Gareth Brentnall         | Holt Court North, Heneage Street West, Aston Science Park, Birmingham<br>B7 4AX UK. Tel: 0121-359 5096. Fax: 0121-359 0375                                                                                                                       |
| MicroAPL Ltd.             | Richard Nabavi           | South Bank Technopark, 90 London Road, LONDON SE1 6LN UK<br>Tel: 0171-922 8866 Fax: 0171-928 1005<br>Email: MicroAPL@microapl.demon.co.uk                                                                                                        |
| Millint Inc.              | Dan Baronet              | 4215 St-Andre, Montreal, CANADA H2J 2Z3. Tel: +1 (514) 529-1434                                                                                                                                                                                  |
| Ellis Morgan              | Ellis Morgan             | Myrtle Farm, Winchester Road, Stroud, Petersfield, Hants GU32 3PE UK.<br>Tel: 01730-263843 Email: Ellis@myrtfarm.demon.co.uk                                                                                                                     |
| Oasis Systems B.V.        | Theo Zwart, Louis Rijkse | Lekstraat 4, 3433 ZB Nieuwegein, Holland Tel: +31 30 60 66 336<br>Fax: +31 30 60 65 844 Email: oasisbv@pl.net or zwart@oasis.nl                                                                                                                  |
| Object Oriented Ltd       | Walter G. Fil            | Am Grendel 2, CH-8004 Luzern, Switzerland. Tel: 41 41 418 70 70<br>Fax: 41 41 418 70 77 Email: info@object-oriented.com                                                                                                                          |
| Optima Systems Ltd        | Paul Grosvenor           | 115 Brighton Road, Purley, Surrey CR8 4HE UK<br>Tel: 0181-763 2490 Fax: 0181-763 2491                                                                                                                                                            |
| RadSys Technologies AB    | Randolph Schrab          | Lovsångarv. 18, S-756 52 Uppsala, Sweden. Tel: +46 18 32 41 53<br>Fax: +46 708 1996 11 Email: 100564.2544@compuserve.com                                                                                                                         |
| Renaissance Data Systems  | Ed Shaw                  | PO Box 421, Georgetown, CT 06982, USA. Tel: +1 (203) 270-9729                                                                                                                                                                                    |
| RE Time Tracker Oy        | Richard Eller            | Mikonkatu 8 A, 2.krs, PL 363, 00101 Helsinki, Finland.<br>Tel: +358 9-621 3300 Fax: +358 9-621 3378 Email: re@rett.fi                                                                                                                            |
| The Rochester Group Inc.  | Robert Miller            | 600 Park Avenue, Rochester, NY 14607-2926, USA.<br>Tel: +1 (716) 271-1110. Fax: +1 (716) 271-1230                                                                                                                                                |
| Rome/Italy SIG            | Mario Sacco              | Casella Postale 14343, 00100-Roma Trullo, Italy<br>Email: marsac@vnetlibm.com                                                                                                                                                                    |
| RusAPL                    | Boris Makeev             | box 971 (for Makeev), Dmitrovskoe Sh., 2, 127434, Moscow, Russia<br>Tel/fax: +7 95 210-7783 Email: makeev@atom.ai.x-atom.net                                                                                                                     |
| SE APL Users Group        | John Manges              | 413 Comanche Trail, Lawrenceville, GA 30044, USA<br>Tel: +1 (770) 972-3755 Email: seapldoc@aol.com                                                                                                                                               |
| Shepp & Associates LLC    | Andrew Shepp             | 1312 Washington Avenue, 6th Floor St. Louis MO 63103, USA<br>Tel: +1 (314) 621-3272 Fax: +1 (314) 621-4267<br>UK Address: Claridge House, 29 Barnes High St, London SW13 9LW<br>Tel: 0181 8768666 Fax: 0181 8768650 Email: ashepp@compuserve.com |
| Snake Island Research Inc | Bob Bernecky             | 18 Fifth Street, Ward's Island, Toronto, Ontario M5J 2B9 Canada<br>Tel: +1 (416) 203-0854 Fax: +1 (416) 203-6999<br>Email: bernecky@eeecg.toronto.edu                                                                                            |
| SOCAL (South California)  | Roy Sykes Jr             | Sykes Systems Inc, 4649 Willens Ave, Woodland Hills,<br>CA 91364-3812 USA Tel: +1 (818) 222-2759 Fax: +1 (818) 222-9250                                                                                                                          |
| Soliton Associates        | Laurie Howard            | Soliton Associates Ltd, Groot Blankenberg 53,<br>1082 AC Amsterdam, Netherlands<br>Tel: +31 20 646 4475 Fax: +31 20 644 1206 Email: sales@soliton.com                                                                                            |
| SovAPL                    | Alexander Skomorokhov    | PO Box 5061, Obninsk-5, Kaluga Region, Russia<br>Tel: +7 (08439) 31463 Email: askom2@kaluga.romail.com                                                                                                                                           |
| Strand Software Inc       | Anne Faust               | 19235 Covington Court, Shorewood MN 55331 USA<br>Tel: +1 (612) 470-7345 Email: amfaust@aol.com                                                                                                                                                   |
| Rex Swain                 | Rex Swain                | 8 South Street, Washington, CT 06793 USA. Tel: +1 (860) 868-0131<br>Fax: +1 (860) 868-9970 Email: rhswain@acm.org                                                                                                                                |
| SwedAPL                   | Christer Uthleim         | Novator Consulting Group AB, Svärdvägen 11C, S-182 33 Danderyd<br>Sweden Tel: +46 8 622 63 50 Fax: +46 8 622 63 51 CServer: 100341,404                                                                                                           |
| Swiss APL User Group      |                          | Swiss APL User Group, CH-3001, Bern 1, Switzerland<br>Email: sl@fi.unizh.ch                                                                                                                                                                      |
| Sykes Systems Inc         | Roy Sykes Jr             | 4649 Willens Ave., Woodland Hills, CA 91364, USA<br>Tel: +1 (818) 222-2759 Fax: +1 (818) 222-9250                                                                                                                                                |
| Toronto SIG               | Richard Procter          | PO Box 55, Adelaide St. Post Office, Toronto Ontario M5C 2H8, Canada<br>Email: info@torontoapl.org                                                                                                                                               |
| Stephen Wynn              |                          | 8 Clarence Gardens, Brighton, Sussex BN1 2EG<br>Tel: 01273-327238 Email: centre@mlstrail.co.uk                                                                                                                                                   |
| Zark Incorporated         | Gary A. Bergquist        | 23 Ketchbrook Lane, Ellington CT 06029, USA. Tel: +1 (860) 872-7806                                                                                                                                                                              |

**WORLD WIDE WEB SITES**

| <b>ORGANISATION</b>      | <b>URL</b>                                                                                                                 |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------|
| AFAPL                    | <a href="http://www.ensmp.fr/~scherer/langle/">www.ensmp.fr/~scherer/langle/</a> (Journal available on line)               |
| APL2000                  | <a href="http://www.APL2000.com">www.APL2000.com</a>                                                                       |
| APL-385                  | <a href="http://www.demon.co.uk/apl385">www.demon.co.uk/apl385</a>                                                         |
| The APL Group Inc        | <a href="http://www.aplgroup.com">www.aplgroup.com</a>                                                                     |
| AUSCAN                   | <a href="http://www.interlog.com/~rjp/auscan">www.interlog.com/~rjp/auscan</a>                                             |
| Capital PC User Group    | <a href="http://cpcug.org">http://cpcug.org</a>                                                                            |
| Causeway                 | <a href="http://www.causeway.co.uk">www.causeway.co.uk</a>                                                                 |
| CODEWORK                 | <a href="http://www.inrete.it/cdwk/eng/home.html">www.inrete.it/cdwk/eng/home.html</a>                                     |
| COSY (Bob Armstrong)     | <a href="http://www.cosy.com">www.cosy.com</a>                                                                             |
| Dinosoft Oy              | <a href="http://lyritys.kolumbus.fi/dinosoft">http://lyritys.kolumbus.fi/dinosoft</a>                                      |
| Dyadic Systems Ltd       | <a href="http://www.dyadic.com">www.dyadic.com</a>                                                                         |
| FinnAPL                  | <a href="http://www.pyr.fi/apl">www.pyr.fi/apl</a>                                                                         |
| Godin London Inc         | <a href="http://www.godin.com">www.godin.com</a>                                                                           |
| Hoekstra Systems         | <a href="http://www.khamsin.demon.co.uk/about/hsl/noframe.html">www.khamsin.demon.co.uk/about/hsl/noframe.html</a>         |
| IBM APL2                 | <a href="http://www.torolab.ibm.com/ap/apl/apl2.html">www.torolab.ibm.com/ap/apl/apl2.html</a>                             |
| Infostroy                | <a href="http://www.insight.dk/infostroy">www.insight.dk/infostroy</a>                                                     |
| Insight Systems ApS      | <a href="http://www.insight.dk">www.insight.dk</a>                                                                         |
| Iverson Software Inc     | <a href="http://www.jsoftware.com">www.jsoftware.com</a>                                                                   |
| Japan APL Association    | <a href="http://www.naska.co.jp/JAPLA/">www.naska.co.jp/JAPLA/</a>                                                         |
| Kestrel Consulting       | <a href="http://www.kestrel-consulting.com">www.kestrel-consulting.com</a>                                                 |
| Lescasse Consulting      | <a href="http://www.lescasse.com">www.lescasse.com</a>                                                                     |
| Lingo Allegro USA Inc    | <a href="http://www.lingo.com">www.lingo.com</a>                                                                           |
| Mackay Kinloch Ltd       | <a href="http://ourworld.compuserve.com/homepages/Alastair_Kinloch">ourworld.compuserve.com/homepages/Alastair_Kinloch</a> |
| MicroAPL Ltd             | <a href="http://www.microapl.co.uk">www.microapl.co.uk</a>                                                                 |
| RE Time Tracker Oy       | <a href="http://www.rett.fi">www.rett.fi</a>                                                                               |
| The Rochester Group Inc. | <a href="http://www.rochgrp.com">www.rochgrp.com</a>                                                                       |
| Shepp & Associates       | <a href="http://www.digitravel.com">www.digitravel.com</a>                                                                 |
| SigAPL                   | <a href="http://www.acm.org/sigapl">www.acm.org/sigapl</a>                                                                 |
| Rex Swain                | <a href="http://www.pcnnet.com/~rhswain">www.pcnnet.com/~rhswain</a>                                                       |
| The APL Group Inc        | <a href="http://www.aplgroup.com">www.aplgroup.com</a>                                                                     |
| Toronto SIG              | <a href="http://www.torontoapl.org">www.torontoapl.org</a>                                                                 |
| Jim Weigang              | <a href="http://www.chilton.com/~jimw">www.chilton.com/~jimw</a>                                                           |

**FTP SITES**

| <b>ORGANISATION</b> | <b>DOMAIN NAME</b>                                                                                                                 |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------|
| IBM APL2            | <a href="http://ps.boulder.ibm.com/ps/products/apl2/">ps.boulder.ibm.com/ps/products/apl2/</a>                                     |
| Toronto toolkit     | see Toronto SIG home page                                                                                                          |
| Waterloo Archive    | <a href="http://archive.uwaterloo.ca/ftparch/languages/apl">archive.uwaterloo.ca/ftparch/languages/apl</a>                         |
| APL-to-ASCII        | <a href="http://archive.uwaterloo.ca/languages/apl/workspaces/aplascii">archive.uwaterloo.ca/languages/apl/workspaces/aplascii</a> |

# K-lite Review

*First Impressions by Jim Lucas (jel@danbbs.dk)*

This is the first instalment of a review of *K-lite*, a time-limited, reduced version of *K*, a language and programming environment with some of its roots in APL. A few days ago I downloaded my own copy of *K-lite* and its associated documentation — the *K User Manual*, *K Reference Manual*, and *K Reference Card* — from KX Systems' website at [www.kx.com](http://www.kx.com). I've now read through both manuals and begun experimenting with *K-lite*.

Perhaps this instalment should really be called "Second Impressions", since several of my first impressions of features in *K* have already changed in just those few days. But for that very reason I think it's important that I publicize my thoughts at this stage, so that you the reader might learn something from following the progress of my own developing understanding. My next instalment — a more comprehensive and detailed review — is scheduled for the next issue of *Vector*.

## What is K?

The first thing I should say is that *K* is *not* APL. *J* may be a dialect or variant form of APL, or it might be called "son of APL", but the relationship is no more distant than that. *K*, on the other hand, is more like a second cousin. Or maybe I should say a great grandchild, since APL is clearly one of *K*'s ancestors, but a majority of *K*'s characteristics just as clearly display other ancestry. *K*'s principal architect is Arthur Whitney, known for his previous creations, the language *A*, and its successor *A+*. When I was working with *A* I tended to characterize it as being composed of "one third APL, one third C, and one third Arthur." I'm inclined to describe *K* in a similar fashion as "20% APL, 20% LISP, and 60% Arthur."<sup>1</sup>

I will say that *K*'s design, with several similarities to APL and numerous differences, seems to be at least as well thought out and self consistent. Some of its features might be profitably borrowed by APL; others represent a different but equally valid approach to problem solution, and there are some I'm not yet sure about.

---

<sup>1</sup> My knowledge of LISP is quite limited, and I know almost nothing about JAVA, DELPHI, and many other popular — and less popular — languages. For this reason, if any readers note similarities that I have missed between *K* and other languages, please let me know via e-mail ([jel@danbbs.dk](mailto:jel@danbbs.dk)) or a letter to *Vector*, and I'll try to include that information in my next instalment.

## Admiration and Frustration!

After reading through the User Manual and the Reference Manual, I was quite impressed by some of the features. I attempted some simple expressions and my admiration grew. While there are a few details I might question, on the whole I'd say that this is one very well designed language!

Then the frustration began as I tried to build the sample application as described in Chapter 3 of the User Manual. The frustration was twofold: first, the interactive "console" lacked the simplest editing and scrolling. Secondly, the features which are not available in *K-lite* include some which are needed by the sample application, so it can't be made to work.

## Editing, the Session, and the Console

In my console, the cursor keys don't work, nor does the mouse interact with the console, so insertion isn't possible, and deletion is only through backspace. Lines that scroll off the screen are lost; they can no longer be viewed, copied or edited. Luckily, I had also enrolled in the *K* listserv, so I asked there for help and information, and I got some.

I discovered that the default console under Windows 95 is basically a DOS session with reduced editing capabilities. I am told that the Windows NT console does have some of the standard editing and scrolling capabilities which I found so frustratingly absent in Windows 95. In either case, a semi-interactive double-window approach seems a bearable substitute: In this approach, all user input is through an editor in one window, while execution is done in a separate window, by either copying or *loading*.

An alternative approach — and the one apparently used by most current *K* users — is to use *emacs* for both editing and execution. Members of the *K* listserv told me where I could get a free copy of *emacs* from the net, which I have now downloaded. I know it's quite powerful, but now I have to learn to use it.

## Lost connections

The documentation includes the following disclaimer with regard to *K-lite*: "K-Lite is a time-limited, reduced version of *K* which enables interested developers to learn the language and develop small applications. K-Lite consists of the *K* language and interpreter, GUI software, and ASCII file read/write capability. It does not include connections, file mapping, interprocess communications or runtime capabilities."

Unfortunately, the features not included are not only among those emphasized in KX Systems' promotional literature, they also include some needed to implement the one significant programming example given in the documentation. This reduces *K-lite's* value in introducing *K* to newcomers.

It's not that *K* without these features is useless, but it would have been nice to have some non-trivial exercises that work, examples that demonstrate *K* techniques by doing real work without needing the missing functionality. I'm trying to design some small applications myself — as a means of learning the strengths and limitations of *K* — but I can't be sure that I'm doing things in a reasonable way. How do I know I'm not like a Fortran programmer happily writing loops in APL because he doesn't know any better?

And an introductory tutorial with examples that work in the *K-lite* environment becomes all the more important when one basically has only one chance to try it. *K-lite* comes with a 60-day time limit and no offer for extending that limit, yet the rumoured price tag<sup>2</sup> for full-featured *K* would put it completely out of the reach of any ordinary person.

### Still, don't be put off

*K* embodies some powerful new concepts. Using *K-lite* to learn about them should provide new perspectives in problem solving. Sixty days should be long enough to get you started with that. In addition, you might find that some features could be compatibly added to existing APLs, and you might want to lobby for one or more of them to be added to your favourite implementation. In order to do that you'll first have to discover them. *K-lite* provides a way.

Let's look at some of those concepts....

### Functions, Operators, and Syntax: Pay Attention, Now

Starting out simply, *K* can look very much like APL, e.g.

```
- 2 3.1 8 = (5-3),+\5 3
-1 0 -1
```

The same expression in APL gives the same result. The five functions (verbs) and one operator (adverb) seem to have the same meanings; the syntax appears to be the same, and the "atomic" functions seem to apply element-by-element to entire

<sup>2</sup> I haven't yet confirmed the rumour with KX Systems, so I won't quote it here. I *will* try to get firm price information for the next instalment of this review.

arrays (Vectors, at least. Actually, in *K* they're lists, but I'll get to that in a minute.), like their scalar-function APL counterparts.

There are other similarities. All *K* primitive functions are single characters, though the characters are all straight ASCII. There are also system functions and variables, with names starting with a special character (underbar in *K*, quad in APL). But some functions which are primitives in APL are system functions in *K*, many familiar APL symbols represent different functions in *K*, and *K* even has some new functions and operators. One of these is "Where" (monadic  $\epsilon$ ). With a boolean argument  $\epsilon b$  gives the same result as  $b/\iota \rho b$  in APL, but without generating the full  $\iota \rho b$  as an intermediate step.

### Enter brackets, stage right

These are small differences, of the sort one might expect with a minor variant of APL. But what about the following?

```
- = [2 3.1 8; (5-3), (+\)[5 3]]
-1 0 -1
```

This is the same computation as above, but in two places it uses an alternative bracket-based syntax for specifying function arguments.

Actually, I think list (bracket-semicolon) form is the primary syntax, while infix (as in 3+4) is the alternate. Infix notation is convenient for dyadic functions, but the list notation can be extended to any number of arguments... three, seven, or even none. In fact, this provides a way of distinguishing between display and execution of a niladic function:

```
foo: {[ ] 3+4}
foo
{[ ] 3+4}
foo[ ]
7
```

The first line defines the niladic function `foo`.<sup>3</sup> On the second line, using the function name without any arguments passes on its definition for assignment, or (in this case) display. The third input (indented) line has an argument list of length zero, so it executes, returning the result of seven.

---

<sup>3</sup> I won't try to explain the details of that syntax just yet, except to say that the colon following a name is *K*'s equivalent of the assignment arrow. The same symbol is used for assignment of both variables and functions.

## Data: Remove Your Blinders!

Data. This is where *K* is really different from APL. Our old concepts don't apply. Some details seem at first to be steps backward: integers and floats are not identical, though they can be compared. And *K* doesn't have multi-dimensional arrays, but only lists. Well, don't let your prejudices fool you!

### Lists

I haven't yet made up my mind about the integer/float distinction, but *K*'s lists seem every bit as powerful as APL's arrays. Lists can be homogeneous or heterogeneous, and they can be arbitrarily nested. In fact, the way that nesting is handled by *K*'s primitive functions makes regular nested lists equivalent to arrays. Not just similar; not just a simulation; but functionally identical, e.g. indexing<sup>4</sup> a nested list:

```
a: (1 2;"ab";5 6)
  a[0 2;1]
2 6
  a[1 2;]
("ab"
 5 6)
```

Not only is this the standard syntax for indexing a matrix, but the "rectangular" second selection is displayed two-dimensionally. There are also primitive or system functions for operations such as transpose and least squares (matrix divide). I think this point is significant enough that I'll repeat it: regular nested lists *are* higher-dimensional arrays. It's not that they can be interconverted, rather that they're two different interpretations of the same structure.

## Namespaces, Directories, and More

In *K* all namespaces form a single tree. Objects can be referenced and even created<sup>5</sup> using the standard syntax with dots as separators:

```
one.one.a: 5 7 9 11
one.two.a: 2*!4 / 2*!46
one.two.plus: +
```

<sup>4</sup> *K*'s index origin is always zero.

<sup>5</sup> By now you should have noticed that *K* uses the colon like an assignment arrow.

<sup>6</sup> Here are a few symbols which are different from their APL equivalents: star for multiplication, exclamation point for index generation, and slash for comment.

```

one.two.a[1 3]
2 6
one[`one `two;`a;1 3]
(7 11
2 6)

```

The first two input lines above create variables in subdirectories. The third creates a function. The fourth is simple indexing. But what's that fifth input line doing? In *K* one can select named objects by using their names as indices just as one can select ordered objects by using their relative locations. That line selects elements 1 and 3 from each object *a* in each of the two namespaces *one* and *two* in the namespace *one*, and the result is a two-element list, each element of which contains two elements, i.e., a matrix. Indexed assignment can be done in the same way. This is powerful!

You must have noticed something else there; the named objects are indexed using a new data type, the symbol. These are names in a form that can be thought of as pointers or handles, but they are distinct from character strings. Symbols can also be created independently of named objects, and in fact symbols can be created which don't represent valid names, but you can see that just their use as name referents is a significant concept. I think it's one APL would do well to borrow.

```

one[`two;`plus `a]
(+
0 2 4 6)
b: one[`two;`plus `a]
b[0]
+
b[1]
0 2 4 6

```

And there you see another important concept. Functions are values, and those values can be elements of lists in the same way as numbers, characters, symbols or other lists. These values can be assigned, indexed, displayed, or even applied.

```

b[0] . 2 5
7

```

In this context the dot is the function "apply", which would seem to blur the distinction between functions and operators just as allowing functions as items of lists blurs the distinction between data and functions.

We can create function lists and apply them to lists of data using the each operator (single quote); we can even combine this with the pervasive nature of atomic functions:

```
(+;-) .' ((2 5;7 17);(5 2;5 7))
(9 22
 0 -5)
```

## Attributes!

Attributes are nothing new, but K has significantly extended the concept. Every object has a syntactic class; all lists have length and depth; APL arrays have rank and shape; every variable has a type and a value; functions have valence. Setting and reading these attributes is generally done by special functions or syntax even in K, but K has generalized the concept of attributes, with a general syntax for setting and accessing their values. In addition, some of them are active, not passive; they cause things to happen whenever their associated objects are either modified or referenced.

The simple syntax for attributes uses double dots, e.g., `a..d` is the display attribute for the object `a`. A number of single-letter attribute-names have reserved interpretations, but in general the K user can assign arbitrary values to arbitrarily-named attributes. These can then be employed however the user wishes.

But user-defined attributes are all passive; their contents must be explicitly retrieved to be useful. On the other hand, reserved attributes can be active, and three appear particularly important. These are dependency (`..d`), trigger (`..t`), and class (`..c`).

Dependency allows the user to provide an expression that will dynamically update the value of its associated variable whenever that variable is accessed and one of the named objects in the dependency has been updated. In a sense, this is like a formula in a spreadsheet cell. Circular dependencies are not allowed; they are detected and reported as loop errors.

A trigger is an expression that is executed immediately after the value of its associated object changes. A dependency makes internal changes based on external events, but a trigger performs external actions based on internal events. Thus, the following two expressions would seem to be equivalent, though there may be subtle differences I'm not yet aware of:

```
y..d: "a*2"
a..t: "y: a*2"
```

And the latter may be rewritten using the system variable for self reference:

```
a..t: "y: _f*2"
```

The class attribute specifies the GUI display class for its associated object. Possibilities include ``form`, ``data`, ``chart`, ``plot`, ``button`, ``check`, and ``radio`. This attribute determines the how the object will appear when displayed as a GUI object. Which brings us to the next major concept, *K*'s GUI. It's like nothing else I've seen.

### Graphical User Interfacelets?

The standard GUI concept is of a single, master Graphical User Interface to which you send requests about what sorts of display areas — windows, buttons, graphs — you want and which data you want pasted into each one. *K* turns this inside out. Each object carries its own personal display information in the form of attributes. This includes not only the display class mentioned above, but such properties as size, colour, font, label, and action on mouse click. It should even be possible to convert the display of a numeric list from a column to a graph just by resetting its `..k` attribute<sup>7</sup>.

If the "editable" attribute (`..e`) on an object is set, then the display can be edited, and any changes will be reflected in the value of the actual object in the workspace, and in any objects affected by associated triggers or dependencies.

The only property that isn't handled in this way is visibility, i.e., whether to show or hide the display defined by the object's attributes and value. Instead, this is manipulated by two special left arguments to the format primitive (`$`), ``show` and ``hide`. Perhaps I'm missing a subtle reason for this, but at the moment it seems to me that this property could and should also be handled as an attribute.

### More Notation: Semicolons, Brackets, and ...

Some have maintained that bracket-semicolon notation in APL is anomalous and should be removed from the language. A few have suggested the opposite, that the notation is a good one and should in fact be extended. Until now I've been a member of the former group, and the proposals for extensions have seemed to offer no improvement. Has *K* finally done it right?

---

<sup>7</sup> Actually, I've tried this and failed, but maybe I've just misunderstood something simple. Or is graphing one of the features unavailable in *K-lite*?

A major problem with bracket-semicolon notation in APL is that a result can be produced which can't be assigned or re-used.  $[a;b;c]$  is not assignable; nor is  $a;b;c$ , and it's not possible to create a variable  $x$  which can be used in the expression  $a[x]$  where  $a$  has any dimension other than one. With this notation, rank has to be hard-coded into the program.<sup>8</sup>

J's solution to this dilemma eliminated the brackets and upgraded semicolons from syntactic separators to functions. Then the special case of elided indices (e.g.,  $a[x;;y]$ ) was simulated by another convention, involving an additional level of nesting.

In  $K$ , on the other hand, brackets ("square" brackets), braces ("curly" brackets) and parentheses are all intimately connected with semicolons in list notation. Brackets and braces signify special interpretation of their contents (indexing and function application for brackets; function definition for braces), but any result created within this notation can be assigned, modified, and re-used. And an elided list element has a "value" of  $\_n$ , or "not a value", e.g.,  $(;;)$  is identical to  $\_n, \_n, \_n$ .

Actually, like infix notation, brackets seem to be handy but redundant. If one defines

```
g: {[x;list] x[list]}
```

then  $x\ g\ (list)$  would be the same as  $x[list]$ . In fact, this  $g$  is the primitive `dot(.)`:

```
x . list.
```

Redundancy aside,  $K$  appears to provide a consistent, functional, powerful and complete notation using semicolons and the various kinds of brackets.

### Conditionals and control structures

I've never felt that traditional control structures fit well into APL. After all, APL already has constructs that implement particular iterations in specific contexts. Aside from itemwise (or pervasive) application of functions, these are certain operators: inner and outer product, reduce, scan, each, and (in Sharp APL) rank. I always felt that it would be more appropriate if APL also implemented IF, DO,

---

<sup>8</sup> Or *constructed* as text and executed. Ugh!

WHILE, etc. as operators. The only trouble is that APL doesn't have a syntax for expressions as arguments or operands. Well, *K* does.

Expressions and functions in *K* are valid list elements, just like numbers, characters, symbols and other lists. Thus it's only natural for *K* to implement controlled blocks as argument lists. `if[condition; expression]` doesn't need an `endif`; in fact, it would be silly. And expression can be more than one expression; alternatively it could be a list of expressions.

### Semicolons and newlines

An independent feature of semicolons in *K* is that outside quotes and in incomplete expressions, they are identical to new lines, e.g.

```
foo: {[a;b] t: a+b; t*t}
foo[7;3]
100
goo: {[a;b] t:a+b
> t*t}
goo[7;3]
100
```

This feature may not seem too important in function definition, though it can be used to improve the readability of code. More significant to my mind is that the same equivalence holds true for data input. This means that a matrix can be entered in the same intuitive multi-row format as its display:

```
a: (2 3;4 5)
a
(2 3
4 5)
b: (2 3
> 4 5)
a ~ b / a identical b?
1
```

## Miscellaneous

*K* has many other features, including error trapping and other debugging tools, various kinds of file I/O, interfaces to other processes, and much more. I don't have the time or space to report everything I know here, and there's still a lot I don't know. The interfaces will be hard to learn, since most are not included in *K-lite*.

*K* is not nearly as permissive as APL in its treatment of white space. White space can have syntactic meaning, i.e., the presence or absence of white space can completely change the meaning — and therefore the result — of an expression. (E.g., `+ [2;5]` gives the result 7, but `+ [2;5]` gives a parse error.) At the moment I'm finding this a bit difficult, though I hope I'll learn the rules fairly quickly.

There are several different ways to do both indexing and function application in *K*. Perhaps with practice I'll come to see the unique advantage of each function and syntax, but at the moment it just seems redundant and confusing.

In my experimentation so far I've run across some details that don't work as I expected. Are they inconsistencies, or just misinterpretation on my part? Hopefully, I can tell you in the next issue.

## Next issue:

I look forward to presenting a more detailed and experienced view of *K* — or at least *K-lite* — in the next issue of *Vector*. In the meantime you might want to download your own trial copy from KX Systems' web site.

---

# Computer Architecture Concepts and Evolution

It's APL, but not as we know it

*book review by Ajay Askoolum*

The authors Gerrit A. Blaauw & Federick P. Brooks use APL, which they describe as a "conversationally executable" high level language, in their book *Computer Architecture Concepts and Evolution* (Addison-Wesley 1997).

"The prime purpose of our APL descriptions is to describe, although the descriptions are executable and hence can be used to simulate and demonstrate. Clarity is favoured over efficiency. We do not shrink from computing a value twice if this repetition makes the meaning of the description clearer ... Although the APL descriptions are intended to be read by many people, they should also be convenient to write and use. A description should be a design tool. As soon as one must make a separate effort to derive a description from a design, the description almost inevitably becomes out of date as the design changes. ... The desiderata for reading and writing sometimes conflict. Unabridged names are easier to comprehend than abbreviations, but are also more tedious to write correctly. We have struck a balance....We assume that the reader is familiar with APL."

Although the book does not contain an APL system (user interface, input/output, error validation etc.), it contains some 650 APL descriptions. These are radically different from the APL I have seen or write.

The APL code is :

1. written in an APL which does not have control structures. However, the code uses the authors' own (APL) implementations of if ... then ... else, while ... endwhile, repeat ... until, case. "... to be read by many people, they should also be convenient to write and use."
2. listed/presented without line numbers.

An example (from page 258):

"The Grau form selects where possible a zero exponent by taking the exponent's minimal absolute value. This choice allows the integer coefficient value to be the same as the fixed-point representation of the given number....

```

grau number
  a grau normal form
  →IF 0=number
  THEN: a exponent for zero
        exponent←zero
        →ENDIF
  ELSE: a Minimal exponent
        exponent←0⌊(1+base⊙|number)+point-(p1+Coef)×base⊙radix
  ENDIF: coefficient←number÷baseexponent-point
  a exponent with minimal magnitude
  →WHILE (exponent<0)∧0=base|coefficient
        exponent←exponent+1
        coefficient←coefficient÷base
  →A
  ENDWHILE:      "

```

As the APL code in this book is sequential, this example presumes global variables exist.

Doesn't the APL look different?

Obviously, the APL the authors have used does not support control structures, and they have implemented their own in order to improve coding and readability. I offer the following illustrations :

### Repeat ... Until

Example: Given a vector, display it dropping leading element until the vector is null. The block of code is executed at least once.

```
RepeatUntil :5
```

```
1 2 3 4 5
```

```
2 3 4 5
```

```
3 4 5
```

```
4 5
```

```
5
```

```
Continue DownHere
```

```
▽ RepeatUntil R
```

```
[1] Repeat:
```

```
[2] R
```

```
[3] →until 0=pR←1÷R
```

```
[4] 'Continue DownHere'
```

```
▽
```

```
▽ Z←until R
```

```
[1] Z←R÷Repeat
```

```
▽
```

**While ... Endwhile**

Example : Display the initial element of a vector, dropping the initial element with each cycle until the vector is null. The block of code is executed only if the condition is true.

```
WhileEndWhile 15
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
Continue Here
```

```
▽ WhileEndWhile R
```

```
[1]  Δ:→while 0≠pR
```

```
[2]  0pR
```

```
[3]  R+1+R
```

```
[4]  →Δ
```

```
[5]  endwhile:
```

```
[6]  'Continue Here'
```

```
▽
```

```
▽ Z←while R
```

```
[1]  Z←R+endwhile
```

```
▽
```

**If ... Then ... Endif or If ... Then ... Else ... Endif**

Example: Determine if a number is odd or even.

```
IfThenElse 4
```

```
Even
```

```
Continue DownHere
```

```
▽ IfThenElse R
```

```
[1]
```

```
[2]  →if 2|R
```

```
[3]      then:
```

```
[4]          'Odd'
```

```
[5]          →endif
```

```
[6]      else:
```

```
[7]          'Even'
```

```
[8]  endif:
```

```
[9]  'Continue DownHere'
```

```
▽
```

```
▽ Z←if R
```

```
[1]  Δ(0=nc
```

```
'else')/'else+endif'
```

```
[2]  Z←0+else,R/then
```

```
▽
```

**Case ... Endcase**

Example: Select a block of code depending on argument.

```
CaseEndCase 'Coffee'
```

```
Coffee
```

```
Continue DownHere
```

```
CaseEndCase 'Orange'
```

```
Unknown
```

```
Continue DownHere
```

```
v CaseEndCase R
```

```
[1]
```

```
[2]   +case R
```

```
[3]
```

```
[4]   'Unknown'
```

```
[5]   +endcase
```

```
[6]   Tea:
```

```
[7]   'Tea'
```

```
[8]   +endcase
```

```
[9]   Coffee:
```

```
[10]  'Coffee'
```

```
[11]  +endcase
```

```
[12]  endcase:
```

```
[13]  'Continue DownHere'
```

```
v
```

```
v Z+case R
```

```
[1]   R+(0+ρ[NL 1])†R
```

```
[2]   Z+, (R^.=ρ[NL 1])†[NL 1]
```

```
[3]   +if 0=ρZ
```

```
[4]   then:
```

```
[5]       Z+θ
```

```
[6]   +endif
```

```
[7]   else:
```

```
[8]       Z+Z
```

```
[9]   endif:
```

```
v
```

Home-built implementations of the control structures, such as the ones above have one serious limitation: the structure can be used only once in any given function. This is because they work by referring to the line numbers on which the keywords appear. If the same labels/keywords appear more than once in a function, they will have the last line value thereby causing incorrect branching. In *Computer Architecture Concepts*, the authors have overcome this as follows: when the If ... Then ... Else structure is required more than once in the same function, they use If ... Then ... Else , If1... Then1... Else1, If2... Then2... Else2 etc. This becomes messy, although it does have the advantage of being able to verify the blocks of code visually.

The interpreter I use supports control structures: I agree that APL code is convenient to write and use with control structures, even with the own implementation variety.

**Conclusion**

*Computer Architecture Concepts and Evolution* is a book on computer Concepts and Evolution, not one which teaches APL.

# Zark Newsletter Extracts

*introduced by Jon Sandles*

This section continues our series of extracts from the Zark APL tutor news. Sadly, I can report that we received no solutions to last issue's crossword nor the limbering up exercise. It is food for thought that Zark had no shortage of entries to the same exercise eight years ago.

## Utility Corner:

### Converting Character Matrices to Numbers

(The purpose of this column is to make you more productive by introducing you to utility functions. Think of utility functions as APL functions that have names instead of symbols. By expanding your function vocabulary, you'll be able to write APL code that's more concise, more efficient, and more readable.)

In the last issue's LIMBERING UP column, we asked you to write a utility function named  $\Delta VM$  ( $\square VI$  on a Matrix). In this column, we'll show the winning solution and explain its logic.

The  $\Delta VM$  function verifies that each row of its character matrix argument can be converted to a single number. Its result is a vector with one element per row of the matrix; a 1 means the row can be converted; a 0 means it can't; a -1 means the row is all-blank.

In addition to this explicit result, the  $\Delta VM$  function returns the global variable  $\Delta FM$  ( $\square FI$  on a Matrix), which is a numeric vector of the converted numbers, one number per row of the character matrix argument. For rows that cannot be converted to a single number, the result is 0.

For example:

```
CMAT
```

```
36.5
10
```

```
STOP
```

```
-28
3 4 5
```

```
ρCMAT
6 5
```

```

      ΔVM CMAT
1 1 -1 0 1 0
      ΔFM
36.5 10 0 0 -28 0

```

The  $\Delta VM$  function can be handy when reading several fields at once from the screen in a full-screen input application. Here's an illustration of  $\Delta VM$  in an APL\*PLUS/PC application:

```

[7]  A Read 15 fields (10 columns) from screen:
[8]  FLDS←0 0 15 10 □WGET 1
[9]  A Branch for err msg unless all
[10] A convertible of blank:
[11] →(∧/|V←ΔVM FLDS)←ERRMSG A Global: ΔFM
[12] A Insert default values where blank
[13] A (DEFAULTS is a 15 element vector of the
[14] A default values for the 15 fields):
[15] BLANK←(V=1)/10V
[16] ΔFM[BLANK]←DEFAULTS[BLANK]
[17] A File the values:
[18] ΔFM □FREPLACE 999 5

```

So how do we write  $\Delta VM$ ? Let's work it out one step at a time.

First, we know that  $\square VI$  and  $\square FI$  work on vectors, not matrices. We'll need to ravel the matrix before applying either function (suppose the character matrix argument is called  $M$ ):

$VI \leftarrow \square VI \ , M$

However, we can't JUST ravel the matrix. Any digit characters in the last column will become adjacent to digit characters in the first column of the next row. For example:

|             |     |     |
|-------------|-----|-----|
| -3625 16810 | ← , | -36 |
|             |     | 25  |
|             |     | 168 |
|             |     | 10  |

To avoid this problem, we can catenate a column of blanks before ravelling the matrix:

|               |     |     |
|---------------|-----|-----|
| -36 25 168 10 | ← , | -36 |
|               |     | 25  |
|               |     | 168 |
|               |     | 10  |

When concatenating to a matrix, the array concatenated can be a matrix, a vector or a scalar:

$M \leftarrow M, [2]((1 \uparrow \rho M), 1) \rho ' '$       a 1 col mat of blanks

$M \leftarrow M, [2](1 \uparrow \rho M) \rho ' '$       a vector of blanks

$M \leftarrow M, [2] ' '$       a scalar blank

The last expression is simplest so we'll go with that. Further, since column-wise (vs. row-wise) catenation is the default, we can omit the axis operator  $[2]$ :

$M \leftarrow M, ' '$

Now let's address the possibility of all-blank rows. We squeeze out the all-blank rows before ravelling and applying  $\square VI$  or  $\square FI$ :

$GOOD \leftarrow \vee / M \neq ' '$       or       $GOOD \leftarrow M \vee . \neq ' '$

$VI \leftarrow \square VI, GOOD \neq M$

While we're at it, let's squeeze out any nasty rows that have more than one number or, more formally, those rows with more than one group of contiguous nonblanks. But how do we determine these rows?

By using a shifting technique, we can identify and then count the nonblank characters that are followed by a blank. Since each group of contiguous nonblanks ends in such a character, we will be counting the number of groups per row. For example:

|                |      |
|----------------|------|
| $M \leftarrow$ | 36   |
|                | 2 25 |
|                | 68 4 |

| NUM | ← +/ |           |   | (M≠' ')   | ^ | (1ϕM=' ') |
|-----|------|-----------|---|-----------|---|-----------|
| 1   |      | 0 0 1 0 0 |   | 0 1 1 0 0 |   | 0 0 1 1 1 |
| 2   | ← +/ | 1 0 0 1 0 | + | 1 0 1 1 0 | ^ | 1 0 0 1 0 |
| 0   |      | 0 0 0 0 0 |   | 0 0 0 0 0 |   | 1 1 1 1 1 |
| 2   |      | 0 1 0 1 0 |   | 1 1 0 1 0 |   | 0 1 0 1 0 |

By assigning the result of  $M \neq ' '$  to a variable we can perform just one relational function ( $\neq$ ) instead of two ( $\neq$  and  $=$ ):

```
NB←M≠' '
NUM←+/NB^~1ϕNB
```

The symbols  $\wedge$  ("and not") can be replaced by the symbol  $>$  since  $A > B$  is the same as  $A \wedge \sim B$  for Boolean arrays  $A$  and  $B$ . (When you see the  $>$  function used on Boolean arrays, read it as "and not." Likewise, read the  $\geq$  function as "or not," since  $A \geq B$  is the same as  $A \vee \sim B$  for Boolean arrays.)

```
NB←M≠' '
NUM←+/NB>1ϕNB
```

These statements are terse and efficient and seem hard to improve. However, Don Erickson's solution showed us that he stays abreast of enhancements in APL. He used the find ( $\in$ ) function:

```
NB←M≠' '
NUM←+/1 0∈NB
```

(The  $\in$  function returns a Boolean array having the same shape as its right argument. Each 1 in the result flags the starting location within the right argument where the left argument is found.)

In fact, since  $NB$  is used only once, we can eliminate its assignment:

```
NUM←+/1 0∈ M≠' '
```

So, consider just the rows of  $M$  having one group of contiguous nonblanks:

```
GOOD←NUM=1
M←GOOD∇M
```

The result of the  $\Delta VM$  function contains  $\bar{1}$ s for all-blank rows (where  $NUM=0$ ), 0s for rows with more than one group (where  $NUM=\bar{1}$ ), and 0s and 1s for rows with one group, depending on the result of  $\square VI$ :

```
VI←-NUM=0
VI←VI+GOOD\square VI ,M
```

The global result,  $\Delta FM$ , contains 0s for rows without exactly one group (where  $NUM\neq 1$ ) and the result of  $\square FI$  for rows with one group:

```
 $\Delta FM \leftarrow GOOD \setminus \square FI ,M$ 
```

After doing a bit more streamlining, we arrive at the winning solution:

```
v R+ $\Delta VM$  M
[1] a Performs  $\square VI$  on M, its char mat right
[2] a argument. Returns one elt per row of M;
[3] a 1 if a single valid number in that row;
[4] a  $\bar{1}$  if all-blank; 0 otherwise. Returns
[5] a numeric vector  $\Delta FM$  ( $\square FI$  on M) globally,
[6] a which has one number per row of M, 0 for
[7] a rows that can't be converted to a single
[8] a number.
[9] a
[10] M←M,' ' a Make sure last col is blank
[11] a How many groups of contiguous nonblanks:
[12] R←+/1 0≤M≠' '
[13]  $\Delta FM \leftarrow R=1$  a Which rows have just one group?
[14] M←, $\Delta FM \neq M$  a Keep these rows; ravel for  $\square VI$ 
[15] a Result is  $\bar{1}$  for all-blank rows and is
[16] a where convertible:
[17] R←(-R=0)+ $\Delta FM \setminus \square VI$  M
[18]  $\Delta FM \leftarrow \Delta FM \setminus \square FI$  M a Replace 1s by converted no.s
v
```

This function is not only the fastest of those submitted, it's the smallest. According to the rules of the contest, it scores 300 points. Here are the authors' names and scores for the five best solutions:

|     |                  |
|-----|------------------|
| 279 | Henry Brudzewsky |
| 258 | Jim Weigang      |
| 209 | John Gaboury     |
| 198 | Greg Mateja      |
| 182 | Don Erickson     |

Many thanks! The final solution was constructed by stealing the best from each submission.

Two similar submissions, from Eric Baelen and Gary Logan, deserve special mention. Eric says he based his solution on a technique used in an article written by Gerald Bamberger in the ACM publication *APL Quote Quad*. The technique is to convert each row into a single number based on the pattern of digits, decimal points, negative signs and spaces. For example, the characters ' -6381.25 ' are translated to '03111121100' (0 for blanks, 3 for negative signs, 1 for digits, 2 for periods) and to '031210' (by deleting repeating 1s and 0s) and then to the number 31210.

Using such a translation scheme, there is a limited set of valid resulting numbers. If not in this set, the corresponding row is invalid.

Here is a complete function using the technique. See if you can follow the logic:

```

V R+ΔVM2 M;P
[1]  A Performs  $\square$ VI on M, its char mat right
[2]  A argument. Returns one elt per row of M;
[3]  A 1 if a single valid number in that row;
[4]  A -1 if all-blank; 0 otherwise. Returns
[5]  A numeric vector ΔFM ( $\square$ FI on M) globally,
[6]  A which has one number per row of M, 0 for
[7]  A rows that can't be converted to a single
[8]  A number.
[9]  A
[10] A Convert each row to a numeric pattern P
[11] A (blank≠0 eliminates leading blanks later):
[12] P+'111111111112304'['0123456789.' '\M]
[13] P+,P,' ' A Work with patterns as a vector
[14] A Eliminate redundant digits and blanks:
[15] P+((P≠1⊕P)≥Pε'01')/P
[16] P+2⊃ $\square$ VFI P A Convert patterns to numbers
[17] A Valid no.s may have 0 (blank) at end;
[18] R+Pε321 21 1 31 12 121 312 3121∘.×1 10
[19] ΔFM+R\ $\square$ FI,' ',R÷M
[20] R+R-P=0 A Result is -1 for all-blank rows
V

```

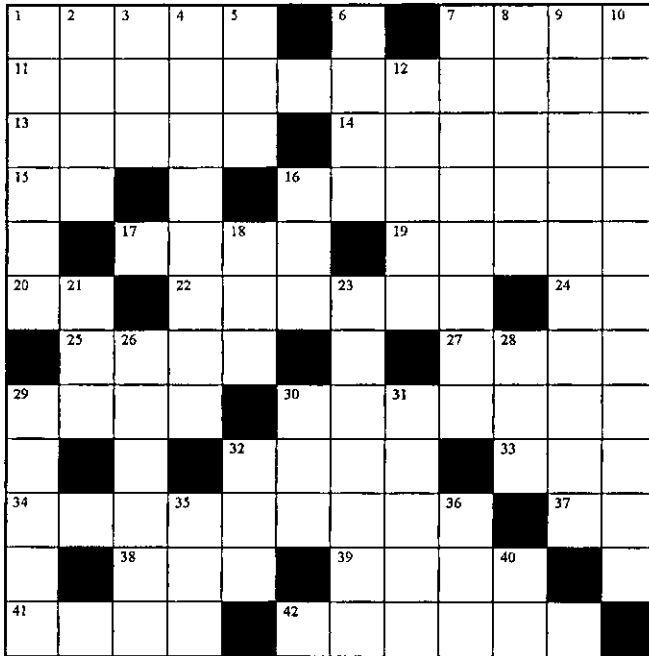
---

Reprinted with kind permission from Zark APL Tutor News, a quarterly publication of Zark Incorporated, 23 Ketchbrook Lane, Ellington, CT06029, USA

## Solution to Crossword from Vector 14.4

|               |                 |               |                 |                 |                 |                 |                 |                 |                |                 |                 |                 |                 |                 |                 |                 |  |
|---------------|-----------------|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--|
|               | <sup>1</sup> ^  | /             | <sup>2</sup> v  | /               | C               | M               | <sup>3</sup> €  | †               | <sup>a</sup> A | <sup>b</sup> E  | I               | <sup>4</sup> O  | U               | <sup>5</sup> †  |                 |                 |  |
| <sup>7</sup>  | L               | /             |                 | /               |                 |                 | [               |                 | <sup>8</sup> 8 | 1               |                 | V               |                 | <sup>9</sup> ρ  | †               | <sup>c</sup> †  |  |
| <sup>10</sup> | M               | ×             | <sup>11</sup> Z |                 | <sup>12</sup> Y | <sup>13</sup> ⊗ | 1               | ,               | [              | 1               | .               | 1               | ]               | ι               | ρ               | Y               |  |
|               | <sup>14</sup> ρ | ρ             | <sup>15</sup> 1 |                 | <sup>16</sup> D | ,               |                 | <sup>17</sup> C | [              | N               | ι               | [               | /               | N               | ]               |                 |  |
| <sup>18</sup> | ρ               | M             | T               | +               | <sup>f</sup> V  |                 | <sup>19</sup> 1 | φ               | ρ              | A               |                 | <sup>21</sup> I | ⊥               | U               |                 |                 |  |
|               | φ               |               |                 | <sup>22</sup> C | [               | <sup>23</sup> P | †               | ψ               | N              | ]               |                 | <sup>24</sup> V | !               |                 | <sup>25</sup> ? |                 |  |
| <sup>26</sup> | N               | [             | <sup>27</sup> 1 | ,               | 1               | †               | ρ               | N               | ;              | 1               | ,               | 1               | +               | <sup>29</sup> ρ | N               | ]               |  |
|               |                 | <sup>30</sup> | N               | +               | A               | M               |                 |                 |                |                 | S               |                 | <sup>31</sup> I | ρ               | V               |                 |  |
| <sup>32</sup> | ^               | /             | C               | =               | C               | [               | ;               | <sup>g</sup> (  | <sup>h</sup> 1 | <sup>i</sup> +  | ρ               | <sup>j</sup> C  | )               | ρ               | 1               | ]               |  |
|               | /               |               |                 | <sup>33</sup> 1 | +               | L               | 1               | O               | ⊗              | N               | [               | 1               |                 | Y               |                 |                 |  |
| <sup>34</sup> | N               | †             | <sup>35</sup> V |                 |                 | <sup>36</sup> M | ,               | 9               | 9              |                 | <sup>37</sup> V | 2               | <sup>38</sup> φ | B               | <sup>39</sup> A | <sup>40</sup> L |  |
|               | V               |               | <sup>41</sup> / | <sup>k</sup> L  |                 | <sup>l</sup> ]  | 1               |                 |                |                 |                 |                 | N               |                 | <sup>42</sup> ^ | /               |  |
| <sup>43</sup> | =               | /             | +               | /               | <sup>m</sup> †  | †               | +               | <sup>n</sup> †  | <sup>o</sup> o | <sup>p</sup> .  | <sup>q</sup> =  | <sup>44</sup> J |                 | <sup>45</sup> I | ~               | J               |  |
|               | [               |               |                 |                 | <sup>46</sup> B | ←               | N               | ρ               | O              | ◊               | B               | [               | I               | ]               | ←               | 1               |  |
| <sup>47</sup> | N               | ←             | <sup>48</sup> M | -               | (               | ρ               | M               | )               | †              | O               | ,               | M               |                 | <sup>49</sup> ι | ρ               | <sup>50</sup> V |  |
|               | V               |               | <sup>51</sup> T | ι               | '               | V               | ]               | †               |                | <sup>52</sup> φ | M               |                 | <sup>53</sup> ) | F               | N               | S               |  |

## Crossword



## Across

1. The shape of a square matrix with as many rows as the array  $A$  has elements
7. Given the two-column matrix  $NR$  and  $S \times R[; 1]$ , reverse the elements in all rows except those beginning with 0
11. Add the vector  $VEE$  to each column of matrix  $B$
13. Do any of the 1s in the bit vector  $X$  correspond to 1s in the same-length bit vector  $Y$ ?
14. Replace the 1s in the bit vector  $SEL$  by the number  $P$ , negated
15. Toggle the 1s in  $M$  to 0s, and the 0s to 1s
16. Jump to the line labelled  $KA$  if the vector  $V$ , whose shape is  $L$ , is not empty
17. Flag the elements of  $V$  that are not multiples of  $M$
19. Put a ceiling of  $DNI$  on each number in  $V$
20. A common variable name for a Boolean vector
22. If a suspended function can't be resumed, \_\_\_\_\_
24.  $1 \div \phi 1 \phi 'ACE'$
25. Select the rows of  $M$  flagged by the bit vector  $B$  and reverse the order of rows
27. The reciprocal of the biggest number in the vector  $A$

29.  $(\sim M) \wedge (\sim N)$
30. The index vector that reorders *VEC* into three subsets: the values less than *T*, equal to *T*, greater than *T*
32. The index vector that reorders *NS*, a character matrix of names, into alphabetic order, where  $A \leftarrow 'ABCD...XYZ'$
33. Printer setting: 12 pitch = 12 \_\_\_\_
34. Flag the last element in each run of like values for the sorted vector *VR*, which contains no 0s
37. Those who can't \_\_\_, teach. Those who can't teach, teach APL
38. Flag the elements of the matrix *M* that can be found in the list *L*
39. For any bit vector *P*:  $+ / P$
41. If  $NV \leftarrow (\phi \times \backslash \phi 1 \div Z, 1) + . \times NM$ , then  $NM \leftarrow$  \_\_\_\_.
42. The error message that reflects shortsighted design

### Down

1. That is the question
2. The number of rows in the numeric matrix *M*
3. Another way to do:  $\phi \rho X$
4. The list of numbers in the matrix *N* from rows that are all-negative (where  $M+N < 0$ )
5. Shift the elements of the vector *Y* to the left *A* positions
6. Return 1 if  $B > K$ , 0 if  $B = K$ , -1 if  $B < K$
7. *S* copies of the value *T* rounded down to the nearest multiple of *E*
8. Another expression for  $VL \div \phi D$
9. Given the boolean vector *NE*, perform  $- \times NE \times PD$  without multiplying
10. Compression of a specific application of this general function
12. What to type when your dog's legs get tangled in the power cord of your computer
16. Resume execution at the line labelled *VE*
18. Eliminate the signs from the numbers in *RM*
21. Shift the values in  $M[; I]$  up  $V[I]$  places, for all *I*
23. Multiply the vector *S* times each row of the matrix *SR*, whose shape is *NR*
26. Flag the rows of *B* that differ from the vector *MN*
28. Round up the numbers in *CC* to the nearest integer
29. Flag the numbers in *NV* that are not in the list *Z*
30. Rank the vector of golf scores *V* (1 for lowest, 2 for next,...)
31. Add a column to the left of the matrix *PF*, containing the number *VS* in every row
32. All but the first *A* elements of the vector *L*
35.  $\vee / , V$
36. For any matrix *U*:  $((1 + \rho U), 0) \rho 0$
40. What APL has in common with PL/1

## J-ottings 17 or: Auto-stereograms and the Data Craftsman

by Norman Thomson

The essence of all computing is *data transformation*. Array languages allow the user to economise on the number of data transformations by perceptive pre-structuring of data. Of course such pre-structuring is itself data transformation, and an assessment of what J provides should make a distinction between *data structuring* transformations as opposed to *operational* transformations. It is convenient to think of the first of these as being about *data craftsmanship*. I would not of course dispute that data-base specialists are data craftsmen in their own right, rather I argue that in array languages, and J in particular, the art of data craftsmanship comes closer to the user than elsewhere in computing.

The J data craftsman has three primary adhesion tools, namely a joiner, a stitcher and a laminator, the flexibility of each of which can be increased by using the rank conjunction attachment. In addition, the joiner and the laminator possess an automatic squaring-off component which ensures that their output is always perfectly rectangular. The versatility of all of these is increased by the addition of two further tools, the boxer and the opener, which allow the craftsman to work on objects either as wholes or as sums of parts.

Like all good craftsmen, the good *data craftsman* is the one who can instinctively assesses the right tool to apply in each workshop contingency. I make no claims to be an expert data craftsman, nevertheless I would like to introduce you to my available tools, my hammers, saws and chisels as it were, and by showing you some of my rough-hewn efforts, invite you, the reader, to contemplate how you yourself might employ them with greater dexterity. Unlike items in a physical workshop, the end result should be the same whether produced by the good or the bad workman, the skill lies in the grace and beauty with which the tools are wielded, and the extent to which this leads to greater versatility in their future use.

Part of the difficulty of comprehending and appreciating the tools is the variety of possible ways in which the same end effect can be achieved once adjuncts such as the rank conjunction and box/open are taken into account. In order to simplify comprehension problems, box/open are not considered in this elementary review.

The tools in the APL data craftsman's kit differ from those of the J craftsman (see J-ottings 5, Vector, vol. 11 no. 4 for details). When the APL craftsman sees

```
abc
def
```

he sees a 2 by 3 character matrix. By contrast the J data craftsman sees a sequence (or list) of two 3 character sequences. To pursue the workshop analogy, the former sees a rectangular block of wood, the latter some sliced contiguous planks. Thus when such an object is constructed using  $\rho$  or  $\$,$  the APL craftsman says to himself "a 2 by 3 matrix using as data ...", whereas the J craftsman says "a list of two 3-item lists using as data ...".

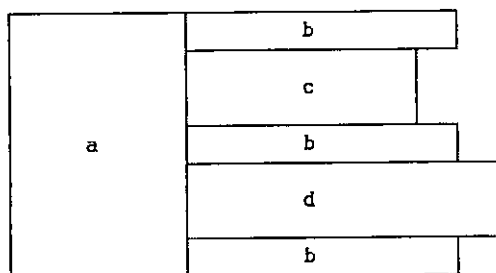
Allow me then to take you on a tour of my carpentry shop. I use the joiner when I already have a pile of planks and I want to stack a second pile on top of it. I do not need to worry about imbalance due to non-matching widths of planks since my joiner automatically squares things up, and ensures that the resultant pile has a nice rectangular cross-section. Next my stitcher ( $. .$ ). I use this when I stack two piles side by side, and then join plank to plank in matching pairs. Naturally the stitcher only works when the numbers of planks in the two piles are equal. Finally the laminator (symbol  $. :$ ) starts a new pile from two existing piles. The shape verb  $\$$  starts its work as a laminator, and continues as a joiner. In this sense the laminator is one of the lowest level (most primitive) operators in the overall J tool kit, and thus one of the most pervasive verbs even although its *explicit* usage is relatively small.

```
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901235678901235678901234567890
1234567890123456789012345678901235678901235678901234567890
1234567890123456789012345678901235678901235678901234567890
1234567890123456789012345678901235678901235678901234567890
1234567890123456789012345678901235678901235678901234567890
1234567890123456789012345678901235678901235678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123X4567890123X45678901234567890
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890
```

Now I want to take you through a real job. I count myself amongst the enthusiasts for stereograms, an invention which Eddie Clough described aptly as

one of life's minor miracles (see again Vector, vol.11 no. 4 for details). In a recent and fascinating book entitled "How the Mind Works" by the American psychologist Steven Pinker, the above pattern is given as a diagram used to explain how autostereograms are constructed (it is in itself a convincing autostereogram — try it!).

The basis of the diagram is that in the short lines two 4s have been removed; in the long lines two Xs have been inserted between 3 and 4. Look at these two areas and use stereogram viewing techniques to observe a lifted and a recessed rectangular block respectively. Viewed as a problem for the data craftsman, here is a possible working plan for joining and stitching:



and here are some possible intermediate stages in its construction:

|                       |                                    |
|-----------------------|------------------------------------|
| t=.'123X4567890'      |                                    |
| u=.t-.'X'             | NB. -. means "less" ..             |
| v=.u-.'4'             | NB. .. so v is '123567890'         |
| a=.21 30\$u           | NB. leftmost 30 columns            |
| b=.3 40\$u            | NB. to stitch onto 1st 3 rows of a |
| c=.(6 18\$v),.6 20\$u | NB. stitch short line blocks       |
| d=.(6 22\$t),.6 20\$u | NB. stitch long line blocks        |

(-. is one of my minor tools, a gouge!)

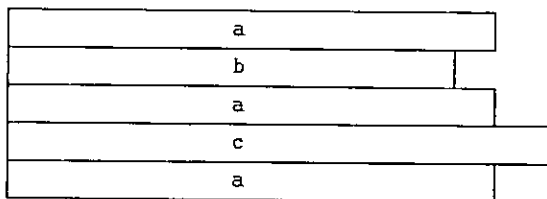
The diagram is then given using the combination of the joiner and stitcher as:

a,.b,c,b,d,b

I find the temptation to abridge the component definitions using each=.<g> and box(<) is irresistible:

```
'tuv'=. (<'123X4567890')-.each ' '; 'X'; 'X4'
'abw'=. (21 30;3 40;6 20)$each <u
'cd'=. ((6 18$v);6 22$t),.each <w
```

An alternative way of stitching and joining has the following plan:



and is obtained by replacing the last two J lines above with:

```
'awx'=. (3 70;6 30;6 20)$each <u
'bc'=. (<w)..each((6 18$v);6 22$t)..each <x    NB. stitch here
```

The Pinker diagram is now given as

a,b,a,c,a

Of course what I have shown is the two dimensional side of my workshop. In reality it is a *multi-dimensional* carpentry shop in which the upward transitions to higher dimensions are so smooth as to require no further explanation once the two-dimensional ones are understood.

Next, a peep at rank conjunction. Every woodworker needs an accountant, and it is no coincidence that mine is also a compulsive joiner and stitcher. When he sees something like this:

```
1 7 4
5 2 0
```

he immediately and instinctively recognises it as a *table* (shadows of the APL data-craftsman!) Not so a J-educated person, to whom it is a list of lists as described above. The column sums are attached by a join hooked with addition:

```
csum=. ,+/
```

and the row sums by a stitch hooked with addition at rank 1 (rows):

```
rsum=. ,.+/"1
```

To attach both column and row sums, sit one atop the other (either way round will do):

```
tots=. csum @ rsum    or    tots=. rsum @ csum
```

To transform each row into proportions of its row total construct another hook:

```
rpro=. %+"/"1
```

following which you might suppose that column proportions are given by `cpro=. %+/` . This formula is dangerously wrong. If you are lucky, you might get a length error to stop you in your tracks, but if the argument happens to be square, the result (transliterated into fractions in order to demonstrate where the figures come from) is:

```
cpro i. 3 3          NB. column totals are 9 12 15
0      1/9      2/9
3/12   4/12   5/12
6/15   7/15   8/15
```

The correct `cpro` is

```
cpro=.%"1 +/
cpro i. 3 3
0      1/12     2/15
3/9    4/12     5/15
6/9    7/12     8/15
```

A final word — much of the above consists of analogies; as soon as these have served their purpose for *you*, I urge you to destroy the images, and revert to the proper J vocabulary, without which exact communication can be greatly impeded.

---

# APL98 AT ROMA

## Conference Reports

### Introduction

*by Adrian Smith*

This was not just a marvellous gastronomic event in superb surroundings. It was a *very* good conference, well run and well attended. The mood was positive, there was genuine excitement at the new facilities in Dyalog 8.2 (the best kept APL secret of recent times), and — yes — it was the best banquet since APL88. The organisers were well and truly cheered at the closing plenary, and they deserved every bit of it.

### Acknowledgements

The Vector team would like to thank everyone who helped to put together this very thorough conference report. The photos are by Ray Cannon, and the reports are by David Crossley, Ray Cannon and Bob Hoekstra.

---

## APL98: A Personal View

*by David Crossley, St Didier, France*

### The Venue...

Rome. Whether you love it or hate it, it is not a city to see and forget. It lives, it would seem, 24 hours a day. It is full of happy, friendly people. The works-of-art are not just limited to the sculptured kind. It has its famous monuments, but they seem to be a very part of the fabric of the city. You can touch them, walk through them, even drive through them. Life is lived in the open air, though with temperatures in the high 90's (35-40°C), this is not too surprising. It is a city of cafés and restaurants that seem to be permanently open (queues for tables at midnight!), and of ice-cream...everybody eats gelato.



**David & Annette Crossley**

Then there is the driving. Creative, suicidal, artistic, macho. These are a few words that come to mind. A dent in one's car appears to be obligatory. But then they seem strangely tolerant of drivers who are foreign to their ways, who stop abruptly to pore over a street map on realising that this is yet another one-way Via going in exactly the opposite way to the intended direction. Hooting vigorously does not, in the main, seem to be part of the arsenal.

So this was the back-drop to the conference. Well...almost. The conference site was at the University of Rome "Tor Vergata" on the outskirts of the city, not an area that attracts

many tourists. ("Tor" is an old word for a tower although I never spotted this particular one.) The conference hotels were scattered around various parts of Rome, mostly around the central area.

Your correspondent, along with quite a sizeable number of delegates, was located in a very attractive quarter called Parioli, not too convenient for the city centre though it did have the advantage of being on top of one of the highest Roman

hills (the advantage, apart from the view, being that the temperature was perhaps a degree lower!). Transport to the conference site was by coach, three in total, that each collected delegates (and accompanying persons, if they wished to spend the day at the conference site!) from a scheduled group of hotels. The downside of the Parioli hotel was that it was first for morning collection (7:50am) and last for evening drop-off (arrival time in the lap of the Gods). Thus, we had two trips lasting up to 1½ hours per day. Never mind. We had a magnificent sight-seeing tour of Rome, absolutely wonderful on the first day. Then I realised the cunning of the conference organisers. These were coach forums (or is it *fora*). I would bet that whole journeys were spent by some people deep in APL discussion without even noticing the magnificence through which we were passing.



**The Front Desk, Hard at Work**

Then there was the conference itself. But first, a few words about the organisers, and the organisation. There is something fascinating about Italian organisation. Hands up all those who thought, at some stage or another, that we were heading for, well, perhaps a fiasco? What about those Web pages that were still Under Construction (and the long wait to download a pretty graphic to bring this to our attention!) just weeks before the conference. Did the accompanying persons wonder about their programme of guided tours? My wife's cousin is something of an old hand at organising conferences in Rome (two so far), but from afar with local support. He finds it a frustrating, hair-tearing experience (though he is excitable)...but then everything works out on the day.

That's what happened at APL98. Everything happened when it should, although one suspected a certain amount of nervous tension in the back-room. Sessions started on time. Equipment was provided for the speakers and presenters. A well-equipped workshop was provided. And the food was wonderful...and this in a University canteen! It certainly comes high on my list of APL conferences for organisation (realised) and facilities, not to mention the conference program itself.

Evening social events started with the Welcome drink, the Banquet, a Music evening, a Rome walk-about, and a Survivors' Bruschetta & Wine evening (*bruschetta* - pronounced brew-sketta) is a simple but delicious recipe of tomatoes and herbs on toast impregnated with olive oil and garlic). The Banquet on the second evening was unusual, but not a bad idea in that people had an earlier opportunity to get to know one another. Of course, the meal was excellent ...was it 8 or 9 courses? Italy is now thought of as the gastronomic country in Europe, and although I live in France, I would concur. We were not disappointed at the conference.

For all of this, we have much to thank Paolo Di Chio, his organising committee, and coordinators. But especially Paolo Di Chio. I do not know how he managed it, but he seemed to have time to attend to even the smallest detail as though one were the only person he had to deal with, both before and during the conference. I have heard this comment from numerous people. In my case, in pre-conference e-mails, he managed to track me down from a FAX with an erroneous e-mail address. Enough said.

### ...and the Conference

Having formerly been a regular attendee at APL conferences, this was my first since APL92 in St Petersburg. I liked the trend that has evolved during the intervening period towards tutorials and, particularly, the workshops with a computer per person. As always, the parallel sessions sometimes presented a difficult choice as to which to attend, the losers often being the refereed papers.

I am glad to report that presentational standards have improved, with the now ubiquitous, modern, graphical tools largely replacing the illegible, scrawled, hand-written foil, often shown incrementally with a covering piece of paper that always slips off. Oops! What a give-away! Also, I know all about stage-fright (hence the need for thorough rehearsal), but inaudible muttering (and this is not a language issue) just out of range of the microphone is not a big audience stimulator. We even had an incoherent speaker (not too subtle clue there) which had something to do with prior evening revelling. But these are specific rather than general criticisms.

I shall say nothing about the refereed sessions simply because they are written up in the Proceedings. I do, however, have a theme. And that is...

### **Yes, APL Does Have a Future**

By Wednesday lunch-time, at the half-way point of the conference, the climate was upbeat. We had heard two plenary sessions whose subjects concerned successful products written in APL. Two new features under late beta-test had been announced at the Dyadic Vendum Forum (sic) that potentially open up new marketing opportunities for APL. We had participated in tutorials and workshops explaining techniques that improve the efficacy of APL (and J) within and outside its domain.

Then we had the Panel Session and a grey cloud descended. The subject was the motto for the conference: "New Gems From Old Roots". Incidentally, there seems to be a new meaning to APL: The Array Processing Language, though I am relieved to say that A Programming Language is still alive, well and dominant. The panel discussion seemed to be saying that APL in its present form is doomed and that a new language, possibly salvaging its array handling capabilities, is our salvation. The panel were most certainly not of accord on this subject, but the negativists outnumbered the positivists by 3:2 with the sixth somewhere between. Early audience statements (hardly questions) also leaned strongly towards the negative view. So what is the matter with the APL community? Do we have a death wish?

It seems that APL is doomed because:

- its notation is symbolic (funny symbols);
- it is an interpretive language (too slow; we should be developing a compiled product);
- it lacks modern programming facilities, largely due to lack of interest/initiative by the vendors;
- it is not portable;
- it is not freely available to educational institutions which prevents students from not only gaining exposure to the language, but also of writing and selling applications without recourse (somebody really suggested this idea!).

We did hear some positive aspects. The advantages of APL are:

- its notation (concise, elegant code);

- it is an interpretive language (fast development on ever faster hardware platforms; à la mode like Java);
- the vendors are constantly innovating and incorporating modern programming facilities;
- it is relatively portable compared with other languages;
- vendors offer very low cost options to educational institutions, often with unlimited availability, but with reasonable prohibition on unlicensed commercial distribution of derived products.

These arguments are, of course, very familiar, repeated endlessly at conferences, in editorials and articles. Unfortunately, it seems that the negative arguments tend to prevail. This is not a good thing for the younger delegates at conferences. Moreover, I am convinced that they do not reflect reality. There are many APLers who obtain a good living from APL. There are many more APLers about than one suspects from conference attendances and association memberships. There are many new applications being developed, new areas being entered. As Mark Twain cabled: "The report of my death was an exaggeration."

### **Eric Baelen: I Can Succeed with APL - and So Can You**

Probably the best, and certainly the most witty, session of the conference was given by Eric Baelen, President of LEX2000 and also APL2000, who completely reversed the negativity of the day before. He commenced by examining 16 or so companies that are, one way or another, APL success stories. Some develop products written in APL. Others have important control systems written in APL. At least one company ran everything using coordinated systems written in APL, including accounts and payroll packages. Many of these companies are large, publicly-quoted multi-million dollar businesses. Often, the users are unaware that the language in use is APL; and, indeed, why should they care. In one case, after a company take-over, the APL system was replaced for the usual pat reasons, only to be re-instated when its full value was realised.

He then went on to review the perceived shortcomings of APL...and demonstrated that these were often advantages. Does it matter that an APL function that runs in 0.5 seconds on the current range of fast machines (mandatory anyway if you wish to run Excel or MS Word in finite time) will run 100 times faster in C? For most business applications, the answer is No. What about the enormous productivity gain in development time of that same piece of code when written in APL?

Eric concluded with a series of tips on how to market APL. The message was clear

and positive. If we believe in the product, spread the gospel, search out market opportunities (and there are many), and exploit them.

[As this was a Plenary Session, it does not appear as a paper in the Proceedings. I do hope that Eric will publish the talk as an object lesson to us all. Exhilarating stuff.]

### **Morten Kromberg: The Adaytum Planning System Architecture**

The opening Plenary Session introduced us to one of the APL success stories. The Adaytum product is a highly successful, business planning product based around a multi-dimensional spreadsheet.



**The Adaytum Delegates Relaxing in the Italian Sun**

We were taken through the historical development of the product. It was the brainchild of George Kunzle whilst working with IBM. When IBM declined to market his product, he left to set up what became Adaytum KPS Software based on this product which he somehow managed to take with him. In 1995, Adaytum acquired the Danish software group, Insight Systems. Their team of APL experts

along with SQL interface know-how provided the basis for rapid development and growth of the already successful KPS product, now known as the Adaytum Planning System. The company has continued to grow rapidly, with current expansion into the US market and ambitious plans for the future.

Morten has no doubts about the reasons for this success. Firstly, they have an excellent product that fills a market niche that was incredibly unfilled. Many large companies get by with a horrendous mish-mash of Excel spreadsheets. Secondly, Adaytum fully understands the importance of excellent marketing. They have a versatile product that is usually sold with a consultancy part to tailor the product to the specific needs of the company.

An on-line demonstration gave us an admittedly limited flavour of this highly functional product. With a fully up-to-date Windows interface, it is no slouch compared with state-of-the-art mass packages. Yet this has been achieved with a handful of APL programmers and a couple of C programmers (to optimise critical coding parts and external interfaces, as I understand it).

In an amusing look at the architectural aspects of the system, and the inevitable pros and cons of using APL, I think that we were all convinced that APL was the right choice.

### **Alexei Mirochnikov / Carlo Alberto Spinicci: The GAMA/SOFIA Portfolio Management System**

Another Plenary Session describing a successful product developed in APL.

In a very amusing opening speech, Carlo Alberto Spinicci of APL Italiana explained how the original SOFIA product came to market as a DOS product in the 80's written in APL\*PLUS for the PC. It is a highly technical product designed for the management of portfolios, latterly specialising in the insurance sector. It has become the de facto package for the business in the Italian market, having obtained an 85% penetration.

[Quote: "As you all know, I speak English very well. However, as a protest against having English as the language for the conference, I shall now speak English very badly."]

The main talk was given by Alexei Mirochnikov of Infostroy, a software house in St Petersburg. He described the evolution of GAMA, a kind of son of SOFIA, to take the DOS version of SOFIA and re-package it as a Windows product using Dyalog APL. His company already had close links with APL Italiana and Insight Systems of Denmark, co-operating to develop software less expensively in Russia.

In a marketing tie-up with a Norwegian company, Infostroy were given the go-ahead to develop the Windows product for a very specific sector in three months! For various reasons that Alexei covered, this target was not met, but the product did come to market and is now selling in various Scandinavian countries, though with different financial emphases.

Alexei then gave a short on-line demonstration of the product. As mentioned, this is a highly technical product, presenting an enormous range of densely packed analyses, not ideal for a projected screen display. It is also a package that requires some considerable training before professionals can take full advantage of its facilities. However, we were able to get the flavour of a very sound product.

Indeed, judging by the questions, there were numerous delegates in the audience who obviously have substantial experience of this highly technical, financial sphere, to the extent that Carlo Alberto felt obliged to defer such questions until later and concentrate on APL implementation matters.

An important aspect here is that extensive multi-lingual support (though maybe not Japanese!) is provided that is fundamentally designed into the product. It is integrated into products such as Excel, Internet Explorer and ODBC.

Of particular interest is the GUI screen development tool developed by Infostroy. With a large number of existing dialogs - I think 70 (different technical screens) was the number mentioned - this facility allows existing dialogs to be quickly changed, and new dialogs to be added without code modification.

## Other Sessions

There were, of course, many other sessions. As mentioned, you can read about the refereed papers in the Proceedings. If you are a developer, I would strongly recommend Bob Bernecky's paper entitled *Egregion: A Branch Coverage Tool for APL*. Although the approach is well-known to most developers, the extension of the ideas should be seriously considered.

As for the vendor workshops and product announcements, I am sure that articles will soon be appearing in Vector. Watch out for Dyadic's Threads and ActiveX Controls.

# APL 98 in Rome

*Observations by Ray Cannon*

## **Work, Rest and Play in Rome, (well 2 out of 3 is not bad)**

**Official Motto "New Gems from Old Roots"**

**Unofficial Motto "Just in Time APL"**

### **Location**

Rome is one of the great cities of the world, even with the temperature in the 90s. The only drawback in holding the conference at Rome's 2nd University was the daily shuttle bus trip from Hotel to Campus.

Even so, I found the trips interesting both for the views of daily life on Rome's roads, and for the opportunity the rides gave me in talking to other APL98 attendees. The shuttle buses were usually on time, I never had to wait more than 10 minutes, and considering the traffic in Rome, that was very good going.

Most of the "official" hotels were located near the centre of Rome, with good access to its shops, restaurants and tourist attractions. Many of the rooms had air conditioning, most of which (but not all) worked.

At my first Hotel, the lift up to the 4th floor regularly overloaded the power supply and tripped a cut-off switch. This happened 4 times of the 5 I used the lift. But power was quickly restored, and the lift restarted. (The last time I was trapped in a lift was in St Petersburg, but that's another conference.)

The Italian food (and wine, I was not driving anywhere) was wonderful and very plentiful, both at conference "do's" and out and about in the city. I developed a real liking for Italian sea-food, spaghetti and clams, fruits of the sea pizza, squid salad....

### **Conference Venue**

The conference was held in a modern building, part of the Faculty of Economics, University of Rome "Tor Vergata", about 30 minutes' drive from the city centre.

The facilities were for the most part excellent and worked well. The workshop room was set out with new computers installed with Windows 95, brought in for

the duration of the conference. As usual, the hardest part was using the Unified APL keyboard on an unfamiliar keyboard with Italian driver. Also, getting some of Windows' error messages in Italian caused some confusion, but that is inevitable at an international conference, unless you take your own keyboard (and drivers) along!

## Timetable

The conference was held over five days, with plenty of slack, and some duplication of items, allowing ample time to attend most of the papers and talks of interest. I think I missed only about 2 items during the conference that I would have liked to attend, due to time-table clashes. For about half the conference, there were up to 3 streams, normally one workshop, one tutorial and one paper (occasionally 2 papers and a workshop). In addition there were Plenary Sessions, Vendor Forums, and Meal breaks which did NOT overlap.

## Workshops

I attended 3 workshops, one on JavaScript and two from Dyadic on OCX and Threads.

The workshop "A Very Gentle Introduction to JavaScript" was given by Moshe Sniedovich, and was more of a tutorial than workshop. In addition to explaining what JavaScript was, it put JavaScript in its place on the web beside HTML, Dynamic HTML, Perl, Java etc. It explained very well when one should use JavaScript (for checking input acquired on web pages) and pointed out that its most common use was to discover which web browser (Netscape or MSIE) was being used to view the web page. He also explained why JavaScript (no relation to Java) should be placed in the Header of HTML rather than the body.

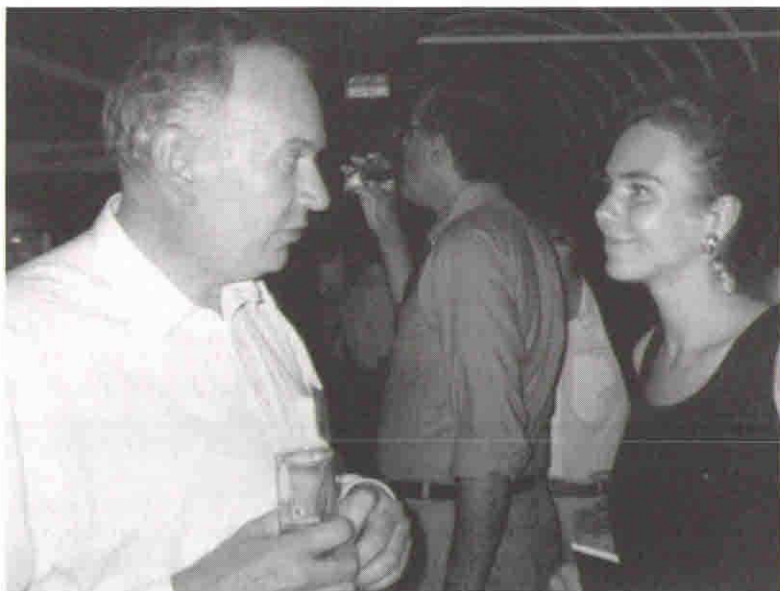
Dyadic tried very hard to keep their twin announcements on OCX and Threads which they made at the first Vendor Forum (Monday lunch) a secret, so the workshops they had planned for them were printed up in the programme as being on "APL as a Web Server" and "OLE revisited".

Peter Donnelly hosted the workshop on "Writing ActiveX Controls in Dyalog APL/W". It was accompanied by a 16-page set of notes which set out to :

- Create an ActiveX Control in Dyalog APL
- Define and Export Properties, Methods and Events
- Include an ActiveX Control in a Visual Basic application
- Run the ActiveX Control from a Web Browser

The workshop appeared to (and in my case actually) achieved all its objectives. It is always very difficult to set the correct amount of work to be undertaken in a workshop. The workshop should be more than just following verbatim a set of written instructions; doing something just once does not teach one the difference between the general requirements and the specifics required in the example. Doing an operation too many times becomes a bore and waste of time. Peter managed the almost impossible, in getting the balance just right. An excellent workshop, thank you very much Peter.

John Scholes hosted the second Dyadic workshop on "Threads in Dyalog APL". Now the version of Dyalog running threads was very much "pre Beta", and although the code was very solid, not all the syntax and constructs had been finalised. So we were warned that the final release version of threads might look quite different. This being the case, the workshop was not aimed at teaching us how to use the current version of threads, but rather the concepts behind threads, and how we might implement multi-tasking within our applications.



**John Scholes explains the finer points of Multi-threading ...**

Again a very enjoyable workshop, being well prepared and documented. It taught me a lot about threads and multi-tasking issues. I look forward to seeing how they are finally implemented in the next production release of Dyalog APL/W.

## Tutorials

I attended two and two thirds out of the six tutorials on offer, sadly missing the others, all of which I would have liked to attend.

Conrad Hoesle-Kienzlen gave the tutorial entitled "APL, TCP/IP and other Languages". I enjoyed the whole talk and found it in particular very instructive about TCP/IP and Windows 95. In re-reading my notes in order to write up this review, I found it impossible not to go off and try out some of the things I learnt. A very good introduction to TCP/IP.

Alexander Skomorokhov started off the tutorial on "Using OpenGL Graphics in Dyalog APL" which ended with Adrian Smith demonstrating a real application developed in conjunction with the Causeway product.

Bob Armstrong gave a three part tutorial, "CoSy: Yesterday, Today and Tomorrow", the title of which led to the following overheard conversion: "Well, today I am talking about 'Yesterday', the day after tomorrow I am talking about 'Today', and the day after the day after that, I am talking about 'Tomorrow'."

(Unfortunately, due to a misunderstanding on my part, I missed the first part on "Yesterday" going instead to the Monday version of a Bob Bernecky paper, which was repeated on the Wednesday without any conflicts. Hence the "two thirds" in the above.)

Bob Armstrong is always worth listening to. For those of you who have never met him (you would not forget if you ever had) Bob is very intelligent and full of life. He lives in New York and travels around Manhattan on a motorised skate-board with his laptop. He very rarely lets himself finish a sentence, but goes off at a tangent as soon as one thought is out. This chaotic style along with some very late nights in the company of (amongst others) some of the Vector working group, led to some amusing talks about CoSy its past, present, and its future.

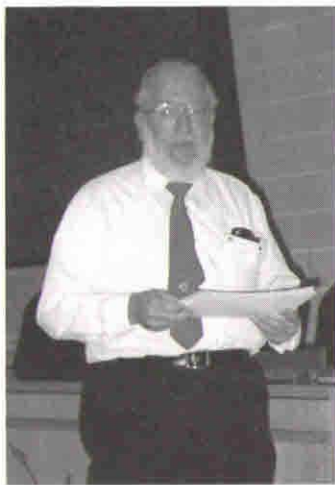
(At this point I would have liked to include a photo of the three Bobs singing "When the red red robin goes Bob, Bob Bobbing along" which I have in my possession, but this idea was vetoed by somebob in the Vector working group.)

## Papers

As a long term fan of Bob Bernecky, I was disappointed in only being able to attend one of the two papers he gave. However I did manage to attend it twice! In "Reducing Computational Complexity with Array Predicates" Bob discussed some ideas that may allow automated optimisation. The idea is that certain

information about data (Array Predicates) can be "tracked" as it is modified as it passes through functions, and that this information can be used to optimise the expression. Without this extra information, the expression has to be implemented in a far more general (and hence inefficient) manner. Now the cost of tracking this information in an interpreter would be far too high to be worth the effort, but it is well be worth the time for an APL compiler and thus has been implemented in "APEX" (Bob Bernecky's APL compiler).

The title of Enrique Alfonseca's paper "Writing a Compiler's Compiler in APL" ensured my attendance. In it, Enrique described a compiler written in APL2 that can (like LEX or YACC) automatically generate a compiler for a given language. The object of this paper TWS (Translator Writing System) takes as its input a "complete description of the language including its grammar in Backus normal form".



Jack Rudd

Another of my favourites at APL conferences is Jack Rudd "APL's fastest gun". His paper "Large Scale Space Object Tracking using APL2" (written with Richard Marsh, and Marcus Munger Ph.D.), completed his trilogy ("Roles of APL in Satellite Surveillance" from APL93 and "Real-time APL Prototype of a Wide-area Differential GPS System" from APL96). He describes how a set of 20 satellites in low orbit could keep track (via infrared imaging systems ) of ballistic missile launches and then track them against a background of stars and 7666 other artificial space objects.

In his paper, "Simplifying Array Processing Languages" Neville Holmes put forward some radical but very interesting ideas to simplify APL-like languages. One such idea was to make ALL functions strictly dyadic. This would require the introduction of a C-like "VOID" construct which Neville called "ZILCH". Zilch as the left and right argument would produce a niladic form. Zilch as a left argument would produce a monadic form. Zilch as a right argument would produce a different monadic form. Zilch as the function would be equivalent to a diamond separator.

Walter Spunde and Peter de Voil's paper "A Web-interfaced Array-based Mathematics Course" gave a fascinating insight as to where education on the web is headed. The idea of a web-based interface giving access to a J server as an integral part of a maths course, allowing a remote student to see the result of an expression plotted on his web browser, is outstanding.

### The Survivors' Party

Following the formal closing of APL98, with a call for papers and participants and organisers for APL99 in Scranton PA (USA), all the survivors of APL98 were invited to a Bruschetta and Wine party hosted by Bob Armstrong. Here our hard-working hosts got the opportunity to let their hair down and join in the fun. A great time was had by all. On behalf of all the survivors, may I take this opportunity to thank Bob.



**Assorted Debris at the Survivors' Party**  
(Bob Armstrong with Paolo di Chio)

### One Final Remark

Overheard at APL98 "... It's a lovely country, except it's full of APL'ers."

## OpenGL Workspace & Russian Student Fund

Under the inspiration and guidance of Alexander Skomorokhov, Alexei Zalivine, a Russian Information Processing student in Obninsk, has written about 150 functions to access the OpenGL API functions under Microsoft Windows using Dyalog APL/W version 8 (32 bit). Included in the workspace are a number of utility functions to facilitate the use of the APIs as well as an interactive tool to experiment with the graphic image. For a more detailed description, please see the April, 1998 issue of *Vector*, pages 128-142.

To make it possible for Alexei and some of his colleagues to attend APL99, a Russian Student Fund has been established. For \$99 you will receive a copy of the software and the *Vector* article. \$50 will be contributed to the Fund and the remainder will compensate Alexei for his work (capitalism, you know) and cover the distribution costs. For \$150, the book, *OpenGL Superbible*, by Wright and Sweet, will be included as well. Additional contributions to the Fund are, of course, most welcome.

Please send a check (preferred in US) or money order in \$US made out to Renaissance Data Systems, or credit card information to: Renaissance Data Systems; P.O. Box 421; Georgetown, CT 06829

Additional information may be posted to our web site, [www.aplbooks.cnchost.com](http://www.aplbooks.cnchost.com). Additional information on OpenGL may be found at [www.opengl.org](http://www.opengl.org).

# Writing ActiveX Controls in Dyalog APL/W

*introduction by Adrian Smith, workshop notes by Dyadic Systems*

## Introduction

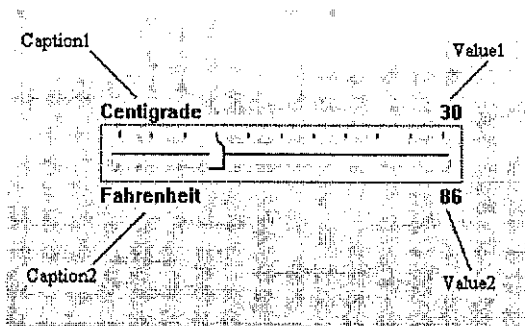
This workshop was probably the highlight of the conference - it was given twice, and was fully booked on both occasions. As Ray has already said, it was an object lesson in how to manage a room full of eager APLers faced with some unfamiliar hardware and novel concepts. Virtually everyone ended the session with the OCX control working correctly, but we all had to work at it just enough that most of the material stood a chance of sticking to our poor confused brains. Here are Dyadic's workshop notes, exactly as used.

## Workshop Notes

An *ActiveX Control* is basically a user-defined control that may be included in GUI applications and Web Browsers. In this workshop you will learn how to

- Create an ActiveX control in Dyalog APL
- Define and Export Properties, Methods and Events
- Include your ActiveX control in a Visual basic application
- Run your ActiveX control from a Web Browser.

## The Dual Control



The ActiveX control we will use for this workshop is deliberately an extremely simple one; so that the intricacies of the control itself do not get in the way of the principles involved.

There are actually very few restrictions concerning the complexity of the ActiveX control, and it is perfectly

possible to package complete multiple-window Dyalog APL applications in this way. Your ActiveX control will be called a *Dyalog Dual Control* and is based on the Dyalog APL *TrackBar* object.

The Dual control allows the user to enter a number using a slider, whilst displaying its value in two different units. For example, you could use it to enter a temperature value which is displayed in both Centigrade and Fahrenheit units. Equally, the same control could be used to enter a measurement of length which is concurrently displayed in centimetres and inches.

### Methods

None. (The Dual control provides no methods.)

### Properties

The Dual control provides the following properties:

| Property  | Description                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------|
| Caption1  | A text string that describes the primary units. This is displayed in the top left corner of the object.                 |
| Caption2  | A text string that describes the secondary or derived units. This is displayed in the bottom left corner of the object. |
| Value1    | The current value of the control measured in primary units                                                              |
| Value2    | The current value of the control measured in secondary units.                                                           |
| Intercept | Used to derive Value2 from Value1                                                                                       |
| Gradient  | Used to derive Value2 from Value1                                                                                       |
| Min       | The minimum value of Value1                                                                                             |
| Max       | The maximum value of Value1                                                                                             |

Value2 is derived from Value1 using the expression:

$$Value2 = Intercept + Gradient \times Value1$$

## Events

The Dual control generates a `ChangeValue` event whenever the user alters `Value1` using the slider.

The event message will contain a single parameter (the new value) which may be modified by the host application. In other words, every time the value in the control changes, the host application may detect this as an event and has the opportunity to override the user.

## Introducing the Dual Control

To save time, the basic APL code for the Dual control has already been written. However, you will have to turn it into an ActiveX control yourself.

1. `)LOAD DUALBASE`
2. Run the function `TEST` and observe how the 2 Dual controls behave.
3. View the function `TEST` and observe how 2 separate instances of the `Dual` namespace `F.D1` and `F.D2` have been created using `⊞OR` and `⊞NS`.
4. Using the Dyalog APL Workspace Explorer, open up the various namespaces. See how `F1.D1` and `F1.D2` are clones of `Dual`.
5. Open the function `Dual.Create` and see how the individual components of the control are defined.
6. Close the Form `F`.

## Changing Dual into an ActiveX Control

1. Change the name of the workspace to `DUAL`:  
`)WSID DUAL`
2. Make a new namespace called `F`  
`)NS F`
3. Using the Workspace Explorer, move the `Dual` namespace into `F`, so that `Dual` is a child namespace of `F`.
4. Now edit the function `F.Dual.Create` and make the following changes:
5. Remove all references to the local variable `POSITION`. This change is required because an ActiveX control has no say in its position within its parent. (Hint: use the Search/Replace dialog to remove all occurrences of `POSITION+`)

6. Remove the right argument, *SIZE* and change *Create[1]* from:

```
H W←SIZE
```

to

```
H W←WG 'Size'.
```

This change allows the control to fit itself within the space allocated by the host application.

7. Change *Create[4]* from:

```
CH←>'##'WG='TextSize' 'W'
```

to

```
CH←>WG='TextSize' 'W'
```

The original code was designed to pick up the character height from the parent Form. The ActiveXControl object does this automatically via its own *TextSize* property.

After making these changes, the *Create* function should be as follows:

```

v Create;H;W;POS;SH;CH;Y1;Y2
[1]  H W←WG'Size'
[2]  SH←40 a Default Trackbar height
[3]  POS←2+⌊0.5×0[(H-SH)]
[4]  CH←>WG='TextSize' 'W'
[5]  'Slider'WG'TrackBar'POS('Size'SH W)
[6]  'Slider'WS('Limits'(Min,Max))('AutoConf' 0)
[7]  'Slider'WS('TickSpacing' 10)('TickAlign' 'Top')
[8]  'Slider'WS('Event' 'ThumbDrag' 'ChangeValue')
[9]  'Slider'WS('Event' 'Scroll' 'ChangeValue')
[10] Y1←POS[1]-CH+1
[11] Y2←POS[1]+SH+1
[12] 'Cap1'WG'Text'Caption1(Y1,0)('AutoConf' 0)
[13] 'Cap2'WG'Text'Caption2(Y2,0)('AutoConf' 0)
[14] 'V1'WG'Text'('Value1')(Y1,W)('HAlign' 2)('AutoConf' 0)
[15] CalcValue2
[16] 'V2'WG'Text'('Value2')(Y2,W)('HAlign' 2)('AutoConf' 0)
v

```

8. Open the function *F.Dual.Build*. This function turns the *Dual*'s parent namespace into a Form (an ActiveXControl currently requires a parent Form) and turns *Dual* itself into an ActiveXControl. It then attaches functions *Create* and *Configure* as callbacks on the Create and Configure events of the ActiveXControl object itself.

```

    ▽ Build
[1]  ##.□WC'Form'('Coord' 'Pixel')('KeepOnClose' 1)
[2]  □WC'ActiveXControl'('Size' 80 200)('KeepOnClose' 1)
[3]  □WS'Event' 'Create' 'Create'
[4]  □WS'Event' 'Configure' 'Configure'
[5]  □NQ'' 'Create'

```

▽

- Run function *F.Dual.Build*. You should see a Form containing a single instance of the Dual control. Please resist any temptation to play with it at this stage; we want it to be in its default state for when we save it.
- Type the following expression; note that the ClassID, which uniquely identifies your control, is allocated when you create the ActiveXControl object.

```
'F.Dual'□WG'ClassID'
```

- Type *)SAVE*. This causes APL to make an OCX file containing your ActiveX control and to register it with Windows. To confirm, type:

```
□CMD'DIR *.OCX'
```

## Testing the Dual Control

- Close Dyalog APL.
- Start Visual Basic and click *Standard EXE*.
- Click the right mouse button in the leftmost toolbox and select *Components* from the pop-up menu.
- Locate the control named *Dyalog DUAL Objects*, set its check box on and click OK. This adds a tool for the Dual control to the Visual Basic designer toolbox.
- Click on the new tool and drag out an outline on your Form. An instance of the Dual control will appear.
- Repeat step 5 to position a second instance of the Dual control on your Visual Basic Form.
- Select *Run/Start* from the Visual Basic menubar.
- Exercise the 2 Dual controls.
- Select *Run/End* from the Visual Basic menubar.
- Click on one of the Dual controls and scroll through its Property list. Notice that all the properties listed are standard Visual Basic ones; there are no properties (or indeed methods and events) exported. We will learn how to do this next.

11. Close Visual Basic; do not save the project.

## Defining and Exporting Properties

1. Start Dyalog APL and load the DUAL workspace (Hint: use the *File* menu; it will be the most recently saved file).
2. Change space into the *F.Dual* namespace.

```
)CS F.Dual
```

3. The properties we wish to export are:

|           |                                             |
|-----------|---------------------------------------------|
| Caption1  | Description of the primary set of units     |
| Caption2  | Description of the secondary set of units   |
| Value1    | The primary value in the control            |
| Min       | Minimum for Value1                          |
| Max       | Maximum for Value1                          |
| Intercept | Used to derive the secondary value (Value2) |
| Gradient  | Used to derive the secondary value (Value2) |

Although we could export all these properties as *variables*, it is generally more useful to employ *Get* and *Put* functions. The reason for this is that there is no mechanism to detect when the host application changes a property/variable; nor is there any mechanism to prevent it assigning an inappropriate value. The *Get* and *Put* functions you need are listed below. To save you time, you can copy them in from the workspace DUALFNS.

```
)COPY DUALFNS
```

```
▽ R←GetCaption1
[1] R←Caption1
```

```
▽ SetCaption1 C
[1] 'Cap1'⌈WS'Text'(Caption1←C)
```

```

    ▽ R←GetCaption2
[1]   R←Caption2

    ▽ SetCaption2 C
[1]   'Cap2'□WS'Text'(Caption2+C)

    ▽ R←GetIntercept
[1]   R←Intercept

    ▽ SetIntercept I
[1]   Intercept+I
[2]   CalcValue2
[3]   'V2'□WS'Text'(*Value2)

    ▽ R←GetGradient
[1]   R←Gradient

    ▽ SetGradient G
[1]   Gradient+G
[2]   CalcValue2
[3]   'V2'□WS'Text'(*Value2)

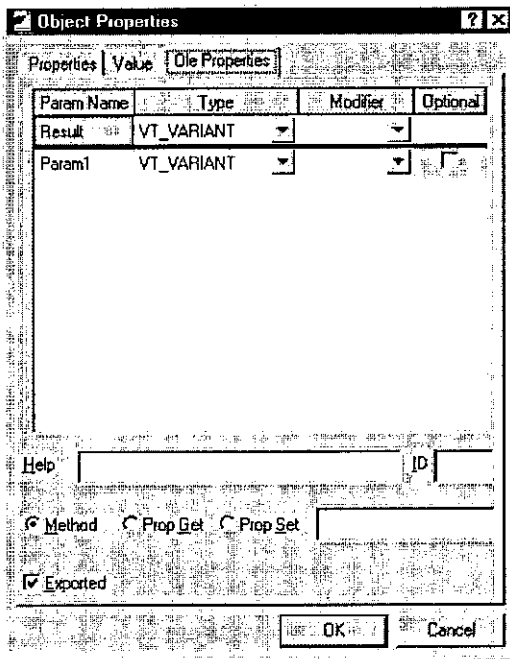
    ▽ R←GetValue1
[1]   R←Value1

    ▽ SetValue1 V
[1]   1 □NQ'' 'ChangeValue'V

```

The *Get* functions need no explanation; they simply return the value of the corresponding variable. The *Set* functions assign a new value to the corresponding variable and update the control accordingly. *SetValue1* does this by enqueueing a *ChangeValue* event which invokes the *ChangeValue* callback.

4. Display the Object Properties dialog box for the function *GetCaption1*. (Hint: use the Workspace Explorer). Select the *Ole Properties* tab. As you have not yet defined any Ole attributes, the default display is as follows:



Using the right mouse button menu, delete the *Param1* and *Param2* rows (this is a *Get* function so it requires no parameter arguments).

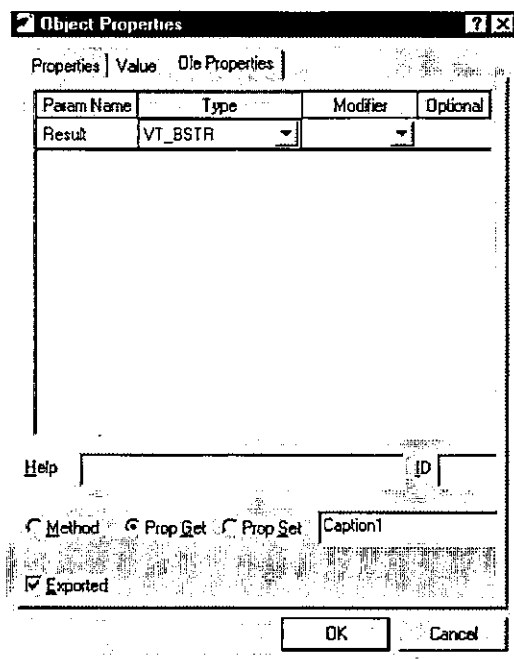
Change the data type for the *Result* to *VT\_BSTR* (a text string).

Check the *Prop Get* radio button to indicate that this is a *Property Get* function and enter the name of the property (*Caption1*) to which it applies.

Finally, make sure that the *Exported* option is checked.

Note that it is not necessary for the property name referenced by the *Get* and *Put* functions to correspond to a variable name, although in this case it does.

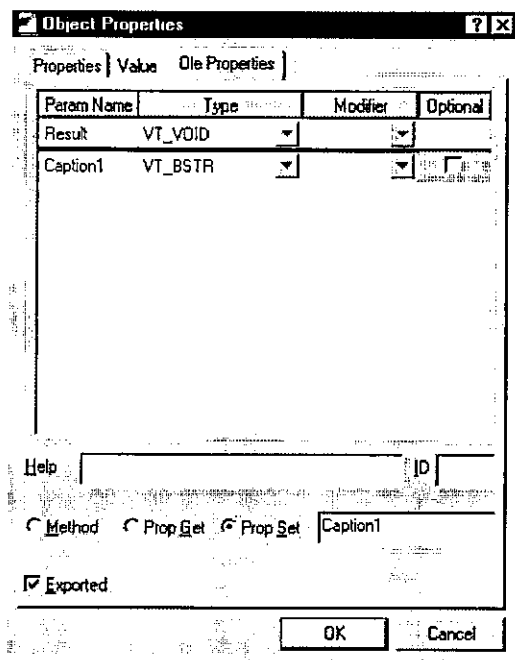
The final Ole properties dialog box for *GetCaption1* should appear as follows:



Now do the same for the *SetCaption1* function. This function takes an argument which it expects to be a character vector. It must therefore be defined as having a single parameter of data type *VT\_BSTR*; the parameter name is unimportant. However, you must ensure that the *Optional* button is unchecked.

In APL terms, the function does not return a result. However, in OLE terms the result is defined to be of type *VT\_VOID*.

The Ole properties for *SetCaption1* should appear as follows:



5. Now complete the Ole Properties for *GetCaption2* and *SetCaption2*, *GetIntercept* and *SetIntercept*, *GetGradient* and *SetGradient* and for *GetValue1* and *SetValue1* in a similar manner. Note that the data type for *Gradient* and *Intercept* properties should be *VT\_R8* and for the *Value1* property it should be *VT\_I4*. Use the following check-list.

#### OLE Properties Check-List

| Function            | Result  | Param | Get/Set | Property  | Done |
|---------------------|---------|-------|---------|-----------|------|
| <i>GetCaption1</i>  | VT_BSTR |       | Get     | Caption1  |      |
| <i>GetCaption2</i>  | VT_BSTR |       | Get     | Caption2  |      |
| <i>GetGradient</i>  | VT_R8   |       | Get     | Gradient  |      |
| <i>GetIntercept</i> | VT_R8   |       | Get     | Intercept |      |

|                     |         |         |     |           |  |
|---------------------|---------|---------|-----|-----------|--|
| <i>GetValue1</i>    | VT_I4   |         | Get | Value1    |  |
| <i>SetCaption1</i>  | VT_VOID | VT_BSTR | Set | Caption1  |  |
| <i>SetCaption2</i>  | VT_VOID | VT_BSTR | Set | Caption2  |  |
| <i>SetGradient</i>  | VT_VOID | VT_R8   | Set | Gradient  |  |
| <i>SetIntercept</i> | VT_VOID | VT_R8   | Set | Intercept |  |
| <i>SetValue1</i>    | VT_VOID | VT_I4   | Set | Value1    |  |

- When you have finished this task, open the Properties dialog box for the Dual object itself and look at the Ole Properties tag. This provides another interface to define exported functions.
- Finally, re-save the workspace. This will update your .OCX file with all the new information.

)SAVE

## Setting Properties from Visual Basic

- Close Dyalog APL
- Start Visual Basic and click *Standard EXE*.
- Click the right mouse button in the leftmost toolbox and select *Components* from the pop-up menu.
- Locate the control named *Dyalog DUAL Objects*, set its check box on and click OK. This adds a tool for the Dual control to the Visual Basic designer toolbox.
- Click on the new tool and drag out an outline on your Form. An instance of the Dual control (called Dual1) will appear.
- In the Properties dialog box, set the Dual1 properties as follows:

|           |             |
|-----------|-------------|
| Caption1  | Centimetres |
| Caption2  | Inches      |
| Gradient  | 0.3937      |
| Intercept | 0           |

7. Repeat step 5 to position a second instance of the Dual control (Dual2) on your Visual Basic Form.
8. Double-Click the left mouse button over your Form (*Form1*). This will bring up the code editor dialog box. Edit the *Form\_Load()* subroutine, entering the following program statements. This code will be run when Visual Basic starts your application and loads the Form *Form1*. It illustrates how you can change the properties of your Dyalog APL ActiveX control dynamically.

```
Private Sub Form_Load()
Dual2.Caption1 = "Inches"
Dual2.Caption2 = "Centimetres"
Dual2.Intercept = 0
Dual2.Gradient = 2.54
End Sub
```

9. Now test your application by selecting Run/Start from the menubar. When you have finished, select Run/End.
10. Exit Visual Basic; do not save the Project or the Form.

## Defining and Exporting Events

1. Start Dyalog APL and load the DUAL workspace (Hint: use the *File* menu; it will be the most recently saved file)
2. Using the Workspace Explorer, open the callback function *ChangeValue* in *F.Dual* Alter *ChangeValue[2]* from:

```
Value1←∘¯1+MSG
```

to

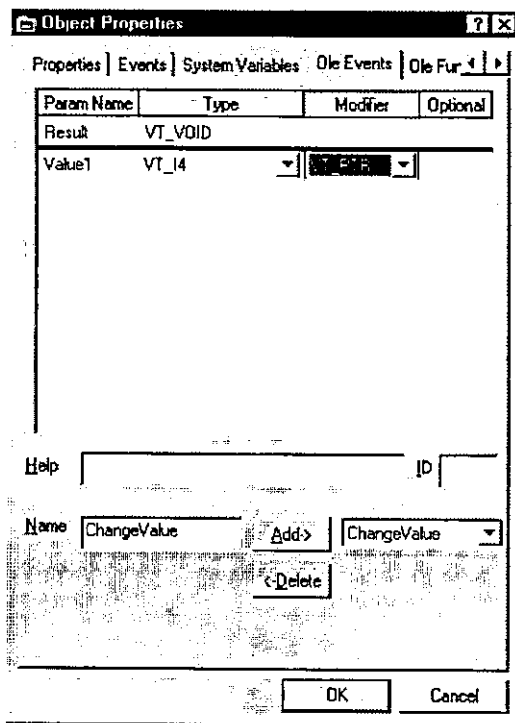
```
Value1←4 □NQ '' 'ChangeValue' (∘¯1+MSG)
```

Then close the function. Previously, the *ChangeValue* function simply accepted the new value (of the Trackbar thumb) that it received as the last element of the event message. Now it generates an external *ChangeValue* event for the host application. The host may in turn modify the new value which is returned as the result of the expression. Thus not only can Dyalog APL generate events which are detectable by the host application, it can also accept modifications.

3. Again using the Workspace Explorer, open the Properties dialog box for the Dual object itself and select the *OleEvents* tab.
  - 3.1. Enter the name of the event *ChangeValue* into the edit box labelled *Name*.
  - 3.2. Click *Add*

- 3.3. Click the right mouse button over the *Result* row and select *Insert*
- 3.4. Change the name *Param1* to *Value1*, the *Type* to *VT\_I4* and the *Modifier* to *VT\_PTR*. This defines the event to supply a pointer to an integer. The fact that it is a *pointer* means that the (integer) parameter may be modified by the host application.

The final appearance of this dialog box should be as follows:



4. Click OK, change back to the root space, and save the workspace

```
)CS #
)SAVE
```

## Using Events from Visual Basic

1. Close Dyalog APL
2. Start Visual Basic and click *Standard EXE*.
3. Click the right mouse button in the leftmost toolbox and select *Components* from the pop-up menu.
4. Locate the control named *Dyalog DUAL Objects*, set its check box on and click OK. This adds a tool for the Dual control to the Visual Basic designer toolbox.
5. Click on the new tool and drag out an outline on your Form. An instance of the Dual control (called Dual1) will appear.
6. Select the label tool and add a label object to the Form. Select its Font property and change it to 14-point bold.
7. Double-click over Dual1 to bring up the code window. Notice how Visual Basic presents you with a skeleton subroutine for the (only) event *ChangeValue* which we have just defined and exported. Notice too that Visual Basic knows that the single parameter is named *Value1* and that its data type is *Long* (VT\_I4).
8. Enter the following code, then close the code window.

```
Private Sub Dual1_ChangeValue(Value1 As Long)
Label1.Caption=Str(Value1)
End Sub
```

9. Start the application using *Run/Start*. Exercise the Dual control and observe that Visual Basic updates the Label1 control in response to the *ChangeValue* events. When you have finished, select *Run/End*.
  10. Double-click over Dual1 to bring up the code window. Alter the Dual1\_ChangeValue subroutine to the following, then close the code window.
- ```
Private Sub Dual1_ChangeValue(Value1 As Long)
Value1=2*(Value1/2)
Label1.Caption=Str(Value1)
End Sub
```
11. Start the application using *Run/Start*. Exercise the Dual control and observe that now the slider moves in increments of 2. When you have finished, select *Run/End*.
  12. Close Visual Basic; do not save anything.

## Using Dual in a Web Page

1. Start Dyalog APL and load the DUAL workspace again.

```
)LOAD DUAL
)CS F.Dual
)COPY DUALUTIL WRITE_HTML
```

2. Look at the function `WRITE_HTML`.

```

v TITLE WRITE_HTML FILE;NID;HTML;CLASS;HEIGHT;WIDTH;CRLF
[1] CLASS←1+~1+⊂WG'ClassID'
[2] HEIGHT WIDTH←⊂WG'Size'
[3] CRLF←⊂AV[4 3]
[4] HTML←'<html><head><title>',TITLE,'</title></head>',CRLF
[5] HTML,+'<body>',CRLF
[6] HTML,+'<h2 align="center"><font face="Arial">'
[7] HTML,+'</font></h2>',CRLF
[8] HTML,+'<p align="center"><object classid="clsid:'
[9] HTML,+',CLASS,'"'
[10] HTML,+' width="',WIDTH,' height="',HEIGHT,'">'
[11] HTML,+'</object></p>',CRLF,'</body></html>',CRLF
[12] :Trap 0
[13] NID←FILE ⌈NCREATE 0
[14] :Else
[15] NID←FILE ⌈NTIE 0
[16] FILE ⌈NERASE NID
[17] NID←FILE ⌈NCREATE 0
[18] :EndTrap
[19] HTML ⌈NAPPEND NID
[20] ⌈NUNTIE NID
v
```

This function writes a very simple page of HTML that references your Dyalog APL ActiveX control via its ClassID (lines [1] and [9]).

3. Now run it:

```
'Dyalog Dual Control' WRITE_HTML 'dual.htm'
```

4. Close Dyalog APL
5. Start your Web Browser and point it at the file you have just saved by typing the URL:

```
file://c:\dyalog82\dual.htm
```

6. Close your browser

## Calling Dual from VBScript

1. Start Dyalog APL and load the DUAL workspace again.

```
)LOAD DUAL
)CS F.Dual
)COPY DUALUTIL UPDATE_CLASSID
```

2. Look at the function .UPDATE\_CLASSID.

```

v UPDATE_CLASSID FILE;NID;HTML;CLASSID;I
[1]  NID←FILE ⌈NUNTIE 0
[2]  HTML←⌈NREAD NID,82,(⌈NSIZE NID),0
[3]  I←'clsid:'⌈HTML
[4]  I←I/⌈ρI
[5]  I←I+5
[6]  CLASSID←1+⌈1+⌈WG'CLASSID'
[7]  HTML[I+⌈ρCLASSID]←CLASSID
[8]  HTML ⌈NREPLACE NID 0
[9]  ⌈NUNTIE NID
v
```

This function simply updates an HTML file with the object's ClassID.

3. Now run the function to update the supplied HTML file called dualtest.htm.

```
UPDATE_CLASSID 'dualtest.htm'
```

4. Close Dyalog APL.
5. Start your Web Browser and point it at the file you have just saved by typing the URL:

```
file://c:\dyalog82\dualtest.htm
```

6. Close your browser.

## APL98: New Gems From Old Roots

by Bob Hoekstra ([bob.hoekstra@khamsin.demon.co.uk](mailto:bob.hoekstra@khamsin.demon.co.uk))



Martin Neitzel with Bob Hoekstra

Here are some extracts from my diary:

### Saturday, 25 July

Rome is a very beautiful city. Ray Cannon and I settled into our hotel and immediately "did" Rome by walking for eight hours through the summer heat. Only mad dogs and Englishmen...

### Sunday, 26 July

After Saturday, we were only fit for the beach. Ostia is very pleasant. The view is good too...

### Monday, 27 July

#### Opening Session

We were welcomed to Rome and APL98.

## The Adaytum Planning System Architecture

Gitte Christensen introduced Morten Kromberg, who proceeded to tell us about this software project, not just the architecture, but the history as well. The origin lies in an internal IBM mainframe APL project, a planning tool called FREGI. The main force behind this was George Kunzle, who (to cut a long story short) eventually combined his company with Insight Systems and this was eventually to become Adaytum Software.

Morten demonstrated the package briefly, showing some of its characteristics, describing and demonstrating D-Lists, D-Cubes and D-Links, which probably won't mean anything to those who weren't there. The apparent ease with which he could create data and meaningful views into that data was very impressive.

He then spoke of the reasons for its success, listing:

- Great Marketing,
- A CEO with technical insight and bravery,
- A great basic idea (thanking Kunzle and Iverson)
- A pragmatic approach, allowing them to mix APL and other languages where this made sense
- A great implementation platform (thanking Dyadic Systems)
- The fact that they had both developers and managers who had grown up using APL.

This was a great session, and I took a lot of interesting impressions away with me. Two comments that stay with me clearly are:

1. the fact that they were using J, not as a replacement for APL, but for C code (!) and
2. that they were encouraging growth in the company by bringing in young people and training them. This was borne out by the fact that the large Adaytum presence at the conference consisted mainly of APLers much younger than the average. Personally, I believe that Adaytum are to be congratulated on this and I hope that they continue in this way, showing that this is the way to make the APL community stronger in the future!

## Dyadic Vendor Forum

Peter Donnelly and John Scholes always seem to come up with the goods! This time they discussed new features in the soon-to-be-released version 8.2 of Dyalog APL. These included:

- Multithreading within the APL interpreter. This allows the user to create separate program strands which each get time-slices from the interpreter. As I understood it, this was not multithreading at the operating system level, and thus would not make better use of the hardware (e.g. forking processes to run on different processors) but is still a very interesting concept — one that I cannot wait to play with!
- The fact that the core of the interpreter is to be implemented as a DLL. This allows for *very* effective use of APL-created GUI objects as ActiveX controls.

Both these new features were to be the subjects of workshops delivered by Peter and John later in the week (see below).

## Dyalog APL Multithreading Workshop

I attended John Scholes's workshop and emerged quite amazed at the extra power and flexibility offered to the APL user by this feature. It would not be an exaggeration to suggest that this opens up entirely new possibilities for the use of APL. The concept of an APL web server immediately springs to mind (do we *really* want to do this with Apache being freely — in all senses of the word — available?) but there are plenty of other examples — almost anywhere where it is critical that an action is to be taken within a limited time of an event occurring.

## CoSy: Yesterday, Today and Tomorrow

Bob Armstrong presented 3 very interesting sessions discussing the past, present and (planned) future of CoSy, his note-taking productivity software package for laptops. Bob managed to amuse everyone at the first (Monday) session by stating '...today is yesterday, Wednesday is today and Friday is tomorrow...'. He was referring (of course) to the subjects covered in his three talks.

Bob demonstrated the CoSy environment and discussed the historical development in today's talk. It certainly seems to be a valid replacement for paper as a note-taking device. The ability to edit notes and have executable (arithmetic) content within the document is very powerful and it is easy to imagine that this would have been easier to write in APL than any other language. His comment that "...paper notes are never there when you need them..." has a lot of truth to it.

## Turtlegraphics — Hidden Features in APL2

Hendrik Rama demonstrated some APL code which he had developed/used in the study of tessellations leading up to the fascinating world of Islamic tiling patterns. This code uses complex number arithmetic to produce patterns which were realised using GRAPHPAK. He showed the listeners many overhead transparencies of the work, building up to some quite beautiful and intricate patterns.

Most of his talk is in the paper in the Proceedings. Unfortunately some of the best graphics we saw at the conference are not reproducible as there is no colour. Nevertheless, this paper will reward the careful reader with a fascinating insight of the subject.

## Tuesday, 28 July

### News about J

Eric Iverson told us about some of the new developments in the forthcoming release 4.02 of J. To me the most striking new features were:

- *Memory mapped files*: These allow direct access of large files, regardless of their actual physical location (hard disk, CD, flashcard, network drive) as though they had been mapped into memory. There was talk of using this as a device to allow two J processes to share data, even when on two separate machines or even operating systems.
- *LAPACK*: Providing this as a compiled software add-on allows very rapid solutions to a large number of problems. Examples discussed included simultaneous linear equations and their least square estimates and eigenvalue problems. This is to be available on all platforms: Windows (3.x, 9x, NT & CE), Mac and UNIX.
- *General C/Fortran package interface*: The interface has been simplified making it easier to call compiled languages from a J process. This was discussed in context of LAPACK as well as in more general terms. Certain problems are so computer-intensive that they are unsuitable for anything but an optimised compiled solution. Eric gave an SVD example where calculation times reduced from a matter of days to sub-second!
- *Windows CE*: This came as a big surprise to me! Iverson Software have ported the interpreter to the hand-held "toys" running CE. This is a complete implementation, including graphics and (as noted above) the ability to use compiled routines like the LAPACK product. Eric described CE as the cleanest Windows implementation and mentioned that he had had many pleased responses when demonstrating this product. One clear market opportunity exists to provide a simple graphing interface to Pocket Excel, which lacks graphics.

### Using APL to Build Science Tutors for the High School Level

Manuel Alfonseca introduced the program apparently used by many schools in Spain to assist in the teaching of Mathematics and Science. The Maths version is apparently much simpler, but the thought that remains with me was just how simple it all appeared to be! This was an excellent example of using the strengths of APL — any other language would have made this project much more complex.

## Teaching Classical Calculation Methods: APL Challenge

Andrei Buzin spoke about the experience of teaching University-level mathematical calculation methods in the University of Friendship, Moscow.

The talk was full of examples (as is his paper in the Proceedings) of the problems addressed. I was impressed by the breadth of the problem and to what extent APL had been able to help him. Many examples have been provided and graphical analysis is done by their own package, displaying graphs of functions on the screen. I would have liked more time to investigate this as a learning method for myself — it is too long since I was concerned with the problems of Cauchy, Euler and friends and my memory is a little hazy.

All in all, Andrei's methods look as though they might be very effective. I am sure that, had I had this sort of introduction to these subjects, I would remember more than I do now!

## Banquet

The evening brought an almost endless supply of great food, wine, good company and entertainment. Those who weren't there really missed something.

# Wednesday, 29 July

## Writing ActiveX Controls in Dyalog APL/W version 8.2

Peter Donnelly led us gently into the art of creating ActiveX controls and calling them from Visual Basic. I cannot imagine how they could have made this any easier! All that now remains is for me to upgrade my copy when version 8.2 becomes available.

Of course I am keen to know how all this works on the UNIX platforms. No doubt I'll find out before too long.

## The GAMA/SOFIA Portfolio Management System (APL Italiana/InfoStroy)

Carlo Alberto Spinicci from APL Italiana opened this discussion in a humorous vein by stating that his was actually perfect and accent-free but had decided to affect a strong Italian accent and bad grammar as a token protest against the continued use of English as the language of APL conferences. He then

(displaying a more than adequate grasp of English) went on to introduce us briefly to the GAMA/SOFIA software package, telling us of its Italian origins.

Alexei Mirochnikov (InfoStroy) then took the podium. He talked about the perceived need for a shrink-wrapped portfolio management system in Scandinavia, the meeting the various parties and how this led to the further development of this product by the Russians. InfoStroy made the product multi-lingual and converted from APL Plus II for DOS to Dyalog APL for Windows in about 3 months, no mean feat, considering there were some 4,000 functions, 300 screens, almost no documentation and most of the existing comments (apparently very sparse) were in Italian!

A demonstration of the software was followed by a brief mention of likely future directions. All in all, a very enjoyable and enlightening talk.

### **Soliton Vendor Forum**

Unfortunately I missed this, but I did hear about a remark concerning a possible APL interpreter for Linux. Those who know me will know that this is one of my pet subjects, so I spent most of the afternoon pestering the person responsible. Unfortunately I have nothing concrete to report, though there is a chance that such a product may see the light of day before long.

*If any readers are interested in such a product, please let me know by email.*

### **CoSy: Yesterday, Today and Tomorrow**

Bob Armstrong presented the second of the three sessions (see above) discussing the current state of the CoSy software.

## **Thursday, 30 July**

### **All Searches are divided into Three Parts: String Searches Using Ternary Trees**

In the title of this presentation, David E Siegel played on remarks Julius Caesar made about Gaul. However there was nothing classical about the subject matter — it was bang up to date!

His discussion of ternary tree search algorithms was very interesting. Of course, this type of algorithm is not a always optimal. Apply this method to a problem

not suited to it and you could make a very slow search, but it can be optimal in the right conditions. For those who want to try it, the Proceedings contain more detail than I can possibly relate — this paper, complete with function listings, greatly rewards the careful reader.

## **I Can Succeed with APL — and So Can You**

Eric Baelen (APL2000 Inc.) provided us with a most encouraging presentation, giving numerous examples of people who had “made it” using APL.

## **IBM Vendor Forum**

Nancy Wheeler ran through the latest news from IBM. The APL2 product line now looks quite impressive, covering everything from MVS/TSO or VM/CMS mainframe, UNIX (AIX and SunOS/Solaris) and OS/2 to (the latest introduction) a Windows version.

Of course the Windows version got a lot of attention, and was demonstrated very well. I found the high level of compatibility between the versions very encouraging. This extends to the inclusion of AP124 and AP127 as part of the distribution. Further highlights (for me) were the ease of sharing data with Excel (ws 2 DDESHARE) and the setup of windows objects with direct mapping onto workspace variables.

## **Making Nested Arrays Work For You**

Philip Benkard's workshop took up the whole afternoon (excluding the tea break) and provided many interesting insights into the use of the modern APL.

Recently I have just “used” the APL interpreter without thinking too much about what I am doing. The 3 hours spent with Phil made me think about the language in a different way, and subsequent re-reading of his papers has provided more food for thought.

For me to give even a taster of his workshop and paper would be impossible. I suggest that most people could do worse than give his paper (*The Epistemology of APL*) a very careful read.



Bob (Hoekstra), Bob (Armstrong) & Bob (Bernecky)

## Friday, 31 July

Unfortunately I had to fly home early in the morning and so missed the Friday sessions. I was (and am) very sorry to have missed several interesting sessions, in particular the last in the trilogy by Bob Armstrong. While I'm on that subject, I have heard from reliable sources that Bob blamed me for his incoherency during this session, something which I wish to refute...

Looking back, a very enjoyable conference. As I said before, Rome is beautiful and I look forward to another visit before too long.

# HTML Basics for APLers (V): Servers

by Adrian Smith ([causeway@csi.com](mailto:causeway@csi.com))

## Introduction

This is the fifth article in our open-ended series on HTML, and continues the themes developed in *Vector* 14.4. By the time you read this, Dyadic will have completed the development of their multi-threading interpreter, which raises a whole new set of design issues. For the moment I am going to assume that you are unlikely to accept any requests which take longer than a few tenths of a second to process, so multi-threading is not a requirement. The other subject which was promised in *Vector* 14.4 was graphical drill-down; this obviously works best with an interpreter which can generate GIF files directly, so this will be deferred until Dyalog 8.2 is available on general release in Q4 of this year.

I will start with a summary of the basic workspace organisation, then cover the functions which convert a simple Causeway-style form definition to its HTML equivalent, and finally run through the code you need in your Web server to make sense of the HTTP request which comes back to your application.

## Workspace Organisation

The basic structure is extremely simple, possibly too simple. I have a namespace called *sws* (for Simple Web Server) which fields HTTP requests. If the user types something in the header line which can be interpreted as a rooted, fully qualified function name, then this gets executed. For example if you run up the server (from now on treat the name 'delia' as 'your server address') and type:

```
//delia/lunch.Signon?name=Adrian
```

... into the query field of your browser, the server will attempt to run:

```
#.lunch.Signon
```

... passing the name of the current TCP/IP socket on the left (most such functions will ignore this) and on the right a two-column matrix of token-value pairs:

|      |        |
|------|--------|
| name | Adrian |
|------|--------|

```

      vhtm←(sock)Signon data;who
[1]  a Return lunch signon form to browser
[2]  a data is a 2-column matrix of token-value pairs
[3]  'Welcome to the Causeway Lunch Server'#.html.Use''
[4]  a The data may give us a clue to the user here!
[5]  who←Δwho←(data[;1]←'name')>data[;2],<'unknown'
[6]  a Uses CPro form definition (refers to Δwho and who)
[7]  #.html.Make lunch1
[8]  htm←#.html.Close

```

▽

This starts by calling *html.Use* to initialise a basic HTML response, then looks up the passed name from the argument and sets a local variable *who* to be the index into the dropdown list of names. The result from the function must always be a valid HTML stream, for example

```

      HelloWorld ''
<html><body>Hello, world<hr></body></html>

```

```

      vhtm←(sock)HelloWorld dummy
[1]  a Just to be sure it works
[2]  htm←Frame'Hello, world'
      ▽

      vhtm←Frame txt
[1]  a Construct minimal HTML from simple text
[2]  htm←'<html><body>',txt,'<hr></body></html>'
      ▽

```

This just allows you to test the socket by typing *//delia/HelloWorld* into the browser and getting a simple response from the server. Of course there are many other ways for the user to submit a request, the most obvious being a click on the 'Submit' button of a form. The server must also have a simple way of choosing the appropriate function to be run (you could have more than one button, or even a clickable image) depending on how the form was submitted. In the long term, I think I should add a 'target function' property to the standard button, but for the moment I just cheated in the usual way by overloading an existing property. Experimental code and all that ... so please don't rely on this not changing!

I now want to spend a few pages working through *html.Make*, but first a very brief summary of the structure of the form definition, as you don't actually need Causeway to do any of this stuff. All you need is an APL interpreter that lets you enter matrices – hardly a heavy requirement!

## Converting a Causeway-style Form to HTML

The basic structure of a *CausewayPro* form is a 5-column rank-2 matrix:

Col-1 is the object type

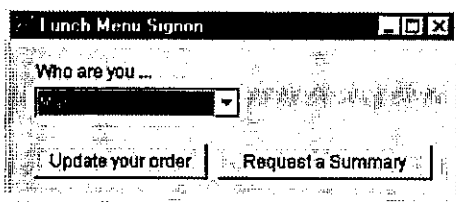
Col-2 is the caption (all objects have captions)

Cols 3 and 4 are the position and size in pixels. Negative positions are measured from the opposite edge of the parent, negative sizes give the total border remaining around the object in that dimension.

Col-5 is itself a 2-column matrix of property-value pairs. All objects have a 'Group' property, other properties vary depending on the object type. Properties known as 'Runtime' or 'User' have APL expressions as their values; these are executed when the form is made to populate the object with data. For example an edit field has a 'Text' property, a droplist has 'Items' and 'Index' which take expressions which should evaluate to a vector of text vectors and an integer respectively.

For example, the lunch-menu signup sheet (*Vettore* page 92) looks like:

```
Δwho
Jon Adrian Duncan Marc JoHo
who+4
#.CPro.Make lunch1
```



The one small enhancement I have made to the Causeway designer is an extra check-box in the preferences field to restrict the object list to those fields which can easily be made in HTML. Currently this allows buttons, edit fields, drop-down lists and check-box and radio-groups. I have started experimenting with the numeric matrix, but this needs a lot more work to be usable in an HTML form.

Otherwise, the form is completely standard, but notice the way I have stolen the 'Groups' property to name the associated function on the action buttons. It is also used to generate the HTML field name of the droplist, which is returned to the 'Update' function as one of the tokens in the token-value matrix.

The form definition is simply:

`Use.Disp lunch1`

|            |                                                          |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
|------------|----------------------------------------------------------|------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----------------|-------|---------------------------------------------------------|------------|-----|-------|----------------------------------------------------------|------------|------|
| FM         | Lunch Menu Signon                                        | 341 84     | 110 307 | <table><tr><td></td><td></td></tr></table>                                                                                                                                                                                           |        |                 |       |                                                         |            |     |       |                                                          |            |      |
|            |                                                          |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| SL         | Who are you ...                                          | 32 16      | 23 -163 | <table><tr><td>Groups</td><td>{WHO}</td></tr><tr><td>Index</td><td><table><tr><td>Expression</td><td>who</td></tr></table></td></tr><tr><td>Items</td><td><table><tr><td>Expression</td><td>Δwho</td></tr></table></td></tr></table> | Groups | {WHO}           | Index | <table><tr><td>Expression</td><td>who</td></tr></table> | Expression | who | Items | <table><tr><td>Expression</td><td>Δwho</td></tr></table> | Expression | Δwho |
| Groups     | {WHO}                                                    |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| Index      | <table><tr><td>Expression</td><td>who</td></tr></table>  | Expression | who     |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| Expression | who                                                      |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| Items      | <table><tr><td>Expression</td><td>Δwho</td></tr></table> | Expression | Δwho    |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| Expression | Δwho                                                     |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| AC         | Update your order                                        | 72 16      | 24 120  | <table><tr><td>Groups</td><td>{lunch.Update}</td></tr></table>                                                                                                                                                                       | Groups | {lunch.Update}  |       |                                                         |            |     |       |                                                          |            |      |
| Groups     | {lunch.Update}                                           |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |
| AC         | Request a Summary                                        | 72 -15     | 24 148  | <table><tr><td>Groups</td><td>{lunch.Summary}</td></tr></table>                                                                                                                                                                      | Groups | {lunch.Summary} |       |                                                         |            |     |       |                                                          |            |      |
| Groups     | {lunch.Summary}                                          |            |         |                                                                                                                                                                                                                                      |        |                 |       |                                                         |            |     |       |                                                          |            |      |

Amazingly enough, it can also be represented as:

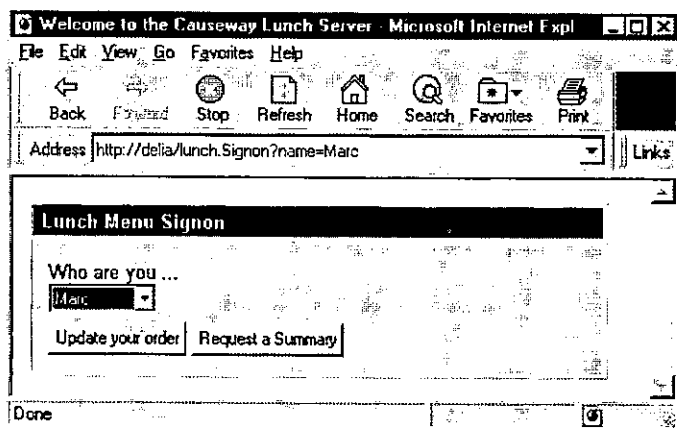
```
<table border cellpadding=2>
<td bgcolor="#000080"><font size=2 face=Arial color="#FFFFFF"><b>&#160;Lunch
Menu Signon</b></td><tr>
<td bgcolor="#C0C0C0">
<table cellpadding=4><td><table>
<form METHOD=POST ACTION="http://127.0.0.1:80/">
<th width=26><em></th><th width=26><em></th><th width=26><em></th><th
width=26><em></th><th width=26><em></th><th width=26><em></th><th
width=26><em></th><th width=26><em></th><th width=26><em></th><th
width=26><em></th><th width=26><em></th><th width=26><em></th><th
width=26><em></th><th width=26><em></th><th width=26><em></th>
<tr>
<td valign=top colspan=6><font size=2 face=Arial color="#000080">Who are you
...<br>
<select name=WHO>
<option>Jon</option>
<option>Adrian</option>
<option>Duncan</option>
<option selected>Marc</option>
<option>JoHo</option>
</select></td>
<tr>
<td><em></td><tr>
<td colspan=3><font size=2 face=Arial><input type=submit name="lunch.Update"
```

```

value="Update your order"></td>
<td colspan=3><font size=2 face=Arial><input type=submit
name="lunch.Summary" value="Request a Summary"></td>
</form>
</td></table></td></table>
</table>

```

... which the browser happily shows as:



Which is a tolerable facsimile of the original. As I noted in the previous article, there are limits to the degree of positional control you have, but for simple data-entry forms I think you can generate remarkably good designs with some simple tricks and cunning use of nested tables. Let's work slowly through *html.Make* explaining the approach as we go.

```

v Make form;defn;....
[1]  a Turn CPro 'dbx' into HTML form (approximately)
[2]  a Executes runtime expressions at build time (unlike Causeway)
[3]  UseDflt o home+>[]NSI o nc+12 o defn+form[1;]
[4]  ttl+2>defn o fmsize+4>defn o host+>hostid
[5]  method+'METHOD=POST ACTION="http://',host,':80/'
[6]  a Some reusable stuff
[7]  pmt+<font size=2 face=Arial color="#000080">'
[8]  a Outer structure for table (grey shaded, titlebar colour)
[9]  tbl+c'<table border cellpadding=2>'
[10]  tbl,+c'<td bgcolor="#000080"><font size=2 face=Arial
      color="#FFFFFF"><b>8#160;',ttl,'</b></td><tr>'
[11]  tbl,+<td bgcolor="#C0C0C0">'
      '<table cellpadding=4><td><table>'

```

This runs *UseDflt* to initialise the HTML page if it needs to, and notes which namespace it was called from, as this is where it must execute the runtime expressions to populate the form. Line [5] sets up the form to post the request to the correct server (*Δhostid* must be initialised with the address of the machine the server is running on) and line [7] is just my preference for bold blue field captions. Now we start playing games with tables!

Line [9] makes an outer table with a border and 2 pixels of padding around the text. The first cell gets a blue background (remember that the user of this form is not running on your computer, so we can only guess at their favourite titlebar colour) and the form caption as bold text. Then we start a new row with `<tr>` and make a light grey background (again guessing that the Windows default is the most likely colour). Into this cell I promptly insert a table with 4 pixels of padding, which has just one cell, containing a table! This is just to get a decent border all around the edge of the form body.

```
[12] tbl,+=c'<form ',method,'>'
[13] a Make nc=12 cells for approximate horizontal alignment
[14] tbl,+=encpc'<th width=',(1[nc+2=fmtsize]),'><em></th>'
[15] tbl,+=c'<tr>'
```

Into this inner table goes the form header, followed up by a table header made up of 12 cells (experiment with *nc* to see the effect, 12 is the lowest value that I think gives a good result) each of which is 1/12th the width of the form. We must put something into each of these cells to define the cell width, but it must not cause any output to the display, as we do not actually want the browser to space downwards here! Hence the `<em>` tag as a placeholder. The *emphasis* is cancelled at every cell boundary, so we do not end up in 12-fold italic.

```
[16] form+=1 0+form
[17] a Ensure controls are sorted col within row ...
[18] form[;3 4]++(cfmtsize)ToReal"+form[;3 4]
[19] form+=(1+pform),form[Δ100001Δ+form[;3];]
[20] cc+=cr+1 o cw+=fmtsize[2]+nc a Current column
```

Having dealt with the form, we can drop it from the definition matrix and attend to the fields and buttons. There is no particular ordering to the form (tab order tends to be top-left ... bottom right, but not rigorously) so having converted from our -ve co-ordinate system to real pixel positions with:

```

      ▽ frm+pg ToReal arg;ps;sz
[1]  a Use pagesize to turn -ve relative coords to real ones
[2]  (ps sz)+arg
[3]  sz+{(sz<0)*pg
[4]  ps+{(ps<0)*pg-sz
[5]  frm+0[ps sz
      ▽

```

... we sort the fields column-within-row, concatenate an 'id' column to the front and initialise the current column and row to 1. The cell width will also be needed to size the edit fields. Now to take each of the form controls and generate the HTML to implement it:

```

[23]  :For defn :In +form
[24]    (uid ty cap posn size props)+defn
[25]    style+{(props[;1]\c'Style'>props[;2],<0pc'' ◊ sd+(c'side')estyle
[26]    a Use the first 'Group Id' as field token, otherwise make one
[27]    uid+{(props[;1]\c'Groups'>props[;2],<c'CPro-',ty,uid
[28]    a Resolve posn and size to cell index and space over
[29]    posn+[cw+~8+posn ◊ cs+[cw+~size ◊ posn[2]->sd
[30]
[31]    :If (2>posn)<cc a Throw a new row here
[32]      tbl,+c'<tr>' ◊ cc+1
[33]      tbl,+c'<td><em></td><tr>' a Vertical padding (one row)
[34]    :End
[35]
[36]    :If (2>posn)>cc a Pad across to next control
[37]      tbl,+c'<td colspan=',(?(2>posn)-cc),'>&#160;</td>'
[38]    :End
[39]
[40]    (cr cc)+posn+1,sd+2>cs ◊ cs+2>cs a Span columns only (safer)

```

The 'Style' property is used by Causeway to hold generic formatting options - in this case we are interested in the 'side' style which switches the caption to the left (rather than above) the edit field. We dig out the first of the 'Groups' as the field identifier, generating a default id if this is missing. The position and size are snapped to our crude grid by line [29] and we slip one cell leftwards to accommodate a 'side' caption. If the horizontal position is less than the current column, we skip to the next row of the table (adding one line of vertical padding). If the horizontal position is greater than the current column we span as many 'empty' cells as are necessary to get to the right spot. Note the &#160; which is a non-breaking space - a harmless way of ensuring that the browser bothers with the spacer cells! Now we can update the current cell positions ready for the next run around the loop.

The next code section handles the captions (top by default, or at the right) ...

```
[42]      a Handle top or side captions
[43]      :If (c<ty>)c'TX' 'NM' 'SP' 'DT' 'HM' 'SL' 'LS' 'NP'
[44]          :If sd
[45]              tbl,+c'<td valign=top align=right>',pmpt,cap,
                  '</td><td colspan=',(c<cs>),'>'
[46]          :Else
[47]              tbl,+c'<td valign=top colspan=',(c<cs>),
                  '>',pmpt,cap,'<br>'
[48]          :End
[49]      :End
```

Side captions get right-aligned in the cell just to the left of the natural position for the field, and are then followed up by the field, set to span as many cells as are required for the width. Top captions just share the cell with their field, and take the default left alignment. The <br> forces a newline within the cell.

```
[51]      :Select ty
[52]          :Case 'HDN' a Hidden field - just to store transaction info!
[53]              data+home Exec(props[;1]c'Value')>props[;2],c''
[54]              tbl,+c'<input type=hidden name="'&uid,'" value="'&(c<data>),'>'
```

Now we must make the HTML for the field. The first thing to do is to execute the appropriate runtime properties to get the actual data values:

```
vr+space Exec expr
[1]      a Execute User/Runtime <expr> in namespace <space>
[2]      a Null is the default return
[3]      r+''
[4]      :If ~0cexpr
[5]          expr+(expr[;1]c'Expression')>expr[;2],c''
[6]          :Trap 0
[7]          r+space+expr
[8]          :End
[9]      :End
v
```

This is then inserted into the HTML, in the case of the 'Hidden' field it is just formatted and ravelled. The 'Label' is very much the same:

```
[55]      :Case 'LB' a Simple text label
[56]          data+home Exec(props[;1]c'Caption')>props[;2],c'cap
[57]          tbl,+c'<td colspan=',(c<cs>),'>',data,'</td>'
```

... only here we must execute the 'Caption' property.

```

[58]      :CaseList 'TX' 'NV'      A Captioned text entry
[59]      A Field size is in characters
[60]      data+home Exec(props[;1]:'Text')>props[;2],c''
[61]      tbl,+'<input type=text size=',(↑[8+~2>size),
          ' name="'',uid,'" value="'',(↑data),'></td>'

```

This takes a wild guess that a character is 8 pixels wide, as the field cannot be set simply to occupy the whole table cell! Fortunately the table will expand to fit the cell if we make it too wide. Now for the numeric and notepad fields:

```

[62]      :Case 'NM' A Numeric input
[63]      data+home Exec(props[;1]:'Number')>props[;2],c''
[64]      tbl,+'<input type=text size=',(↑[8+~2>size),
          ' name="'',uid,'" value="'',(↑data),'></td>'
[65]      :Case 'NP' A Notepad
[66]      data+home Exec(props[;1]:'Text')>props[;2],c''
[67]      :If 2>|=data o data+,<data o :End A Ensure vtv
[68]      tbl,+'<textarea name="'',uid,'" ',
          (↑Rows=' ' Cols='',"~"[14 5+~size),
          ' wrap=virtual>',↑data
[69]      tbl,+'(↑data),c'</textarea></td>' A Line-breaks!

```

Again, the size is in characters, and this is very indeterminate as some browsers use fixed-pitch here, and Netscape-4 uses a really tiny font! The next block is for the drop-selection:

```

[70]      :Case 'SL' A Drop selection (single select via index)
[71]      data+home Exec(props[;1]:'Index')>props[;2],c''
[72]      ref+home Exec(props[;1]:'Items')>props[;2],c''
[73]      tbl,+'<select name="'',uid,'">'
[74]      tbl,+'((' <option'),"((↑pref)↑data)"/"
          c' selected'),'>',"ref,"c'</option>'
[75]      tbl,+'</select></td>'

```

This gets two properties to populate the list and mark the selected option as the one matching the current index. This continues down to:

```

[96]      :CaseList 'EN' 'QT' 'AC' A OK Button etc
[97]      tbl,+'<td colspan=3><font size=2 face=Arial>
          <input type=submit name="'',uid,'" value="'',cap,'"></td>'
[98]      :Else
[99]      '*** unsupported object class - ',ty
[100]      :EndSelect
[101]      :EndFor
[102]
[103] A Tidy up the tail (close all open tags) ...
[104] tbl,+'</form>' '</td></table></td></table>' '</table>'

```

... which closes the form and all three nested tables.

It is interesting how a change in coding style can follow an enhancement to the interpreter - in common with most APLers I always felt uneasy with functions longer than 50 or so lines. Now we have Select - Case (with appropriate indenting) I have no difficulty at all with functions of well over a hundred lines. In this example, I had originally intended to make subfunctions for each object class; in fact it is much easier to add in new :Case or :CaseList blocks into the single large function which has ready (and obvious) access to all the relevant local variables.

As a second example, here is the 'Seminar Registration' (Vettore page 101) as a normal Causeway form and as shown in the browser after conversion:

```

name←'Adrian'
age←42
lang←'Cobol' 'Visual Basic' 'Java' 'Dyalog APL'
pref←4
used←0 0 1 1
smf←'Male' 'Female'
sex←1
notes←''
proficient←1

```

The screenshot shows a window titled "Causeway Seminar Registration". Inside, there are several input fields and controls:

- Name:** A text box containing "Adrian".
- Age:** A text box containing "42".
- Sex:** Two radio buttons labeled "Male" and "Female". "Male" is selected.
- Known:** A list box containing "Cobol", "Visual Basic", "Java", and "Dyalog APL". "Dyalog APL" is highlighted.
- Preference:** A dropdown menu showing "Dyalog APL".
- Proficient:** A checkbox labeled "Proficient" which is checked.
- Notes:** A large text area for notes.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Note that I have kept this fairly simple, although it does mix 'side' and top-aligned captions and is fairly ambitious with the notepad field.

Here is the HTML version:

I am not sure about the raised border around the radio group, but apart from this the alignments have worked out quite nicely. Now that I have filled in some details, let's submit the form to the Causeway server.

## Processing an HTTP 'Post' Request

My 'Receive' function started life as the demonstration code that came with Dyalog 8.1, and then grew lots of weird appendages to deal with the more interesting HTML variations! Here are the first few lines ...

```

v Receive msg;sock;crlf;req;cmd;qry;len;htm;pos;propval;url;fn
[1]  a Collect data and process it when a complete package is received.
[2]  a 'GET' is easy - always ends with crlf,crlf
[3]  a 'POST' uses this to split the header portion. We should check that
[4]  a      the length in the 'Content-Length:nnn' has all arrived.
[5]
[6]  sock+=msg o req+sock Accumulate 3=msg o crlf+AV[4 3]
[7]  Log'Request received via ',sock,' (',(vpreq),' bytes)'
[8]
[9]  :If ~(4pcrlf)greq a Is message obviously incomplete?
[10]      :Return
[11]  :EndIf

```

Line [6] only runs more than once if the message is longer than one TCP/IP block (usually 8K) - it switches into the namespace of the socket and accumulates the message into a buffer, returning the entire buffer:

```

vnew+this Accumulate txt
[1]  a Build buffer which is held as instance data in <this>
[2]  []CS this
[3]  buffer,+txt
[4]  new+buffer
v

```

Log just echoes its argument to the session, then we exit if there is no {cr,lf} pair anywhere in the message. Clearly there is more to come!

Assuming that we stand some chance of having collected the message (at this point we cannot know for certain) it can be partitioned and analysed:

```

[13]  req+2+"(cr|lf$req)<req o cmd+>req
[14]
[15]  :Select s+cmd
[16]  :Case 'GET /'

```

I am going to ignore 'GET' for the moment (it is the easy case) and skip to the 'POST' section ...

```

[43]  :Case 'POST ' a May be incomplete, so check the data part
[44]  gry+(1+(>op"req)\0)>req
[45]  a NB. IE3 uses '-Length' and Netscape uses '-length' here.
[46]  pos+|/(15+"req)\ 'Content-length:' 'Content-Length:'
      o →(pos>preq)+0
[47]  len+15+pos>req o →(len+pqry)+0

```

If the message contains any lines which look like 'Content-length' and the part after the last null line is the right length, then we have a complete message, otherwise quit and wait for some more to arrive! Then go ahead and run it:

```

[48]  a Looks OK, as the expected element is the right length
[49]  htm+ '<html><body><h3>Command Requested:</h3>',cmd,'<hr>'
[50]  a Extract the business end of the message (dehexed)
[51]  gry+'I',BackFlip gry
[52]  propval+(wp~$2,[0.5*pw])1+"gryc~gry='I'

```

We saw *BackFlip* in *Vettore* - it reverses the normal MIME encoding. The result will be delimited by I-beam characters and will always have an even number of sections. Line [52] partitions it and reshapes to a 2-column matrix of tokens and values. Now to see if there are any functions in there ...

```

[53]      :If 0<pfn+'FindFn propval[;1]
[54]      :Trap Δerrors a Report errors to browser
[55]      htm←sock(afn)propval
[56]      :Else
[57]      htm←Frame fn ShowError ΔDM
[58]      :End
[59]      :Else
[60]      gry←ε(,propval),“(p,propval)p” = ‘ ‘<br>’
[61]      htm←gry,‘<hr></body></html>’
[62]      :End

```

First it goes looking for functions in the token list:

```

vfn←url FindFn tokens
[1]  a See if the URL or any token is a function in this WS
[2]  a Expects rooted (but not necessarily qualified) names
[3]  a Returns fully qualified name if it finds one
[4]
[5]  :If 3=ΔNC‘#.’,url a # is invalid in a url
[6]  fn←‘#.’,url
[7]  :Return
[8]  :End
[9]
[10] a Check for names from images in forms (foo.x and foo.y)
[11] a Always come in pairs, so we only need to strip one of them.
[12] tokensv←{w~2x-‘.x’=‘2+w}”tokens
[13]
[14] a Ensure all names (which must be rooted) are ‘#.’ based
[15] tokens←{w,“‘#.’/“‘#’*1+w}”tokens
[16]
[17] a Now see if there is a matching function out there ...
[18] a First hit wins here!
[19] fn←{3~ΔΔNC”tokens>tokens,c”
v

```

If nothing matches, the code drops through and returns the formatted request to the browser – handy for remote debugging! Otherwise it sets a trap (Δerrors would probably be 0 to trap all errors, or could be null if we wanted to fix errors as they happened) and executes the function it found, passing the socket and the token list. If it fails (and the trap is set) it returns a formatted error to the browser:

```

vhtm←fn ShowError dm
[1]  a Echo APL error to browser (more diagnostics would be better)
[2]  htm←‘<h3>’,(Δdm),‘</h3>executing function {’,fn,‘}<br><br>’
[3]  htm←‘<code><font face=APL2741>’,(Δdm),‘</font></code><hr>’
v

```

... in an APL font, or Courier if this is not installed on the client machine.

The rest is up to the application! In this case it will run *#seminar.Accept*, which can process the token-value list in any way it pleases, returning some appropriate HTML to the socket:

```

vhtm+{sock}Accept data
[1]  A Log the new attendee in the database
[2]  SE.Disp data
[3]  htm+#.Frame'Registration accepted - thanks!'
v

```

OK, it lied! All it actually did was:

|                       |                                |
|-----------------------|--------------------------------|
| <i>CPro-TX1</i>       | <i>Adrian</i>                  |
| <i>CPro-NM2</i>       | <i>45</i>                      |
| <i>CPro-RB3</i>       | <i>Male</i>                    |
| <i>CPro-LS4</i>       | <i>Java</i>                    |
| <i>CPro-LS4</i>       | <i>Dyalog APL</i>              |
| <i>CPro-SL5</i>       | <i>Dyalog APL</i>              |
| <i>CPro-YN6</i>       | <i>on</i>                      |
| <i>CPro-NP7</i>       | <i>Please note that I h...</i> |
| <i>seminar.Accept</i> | <i>OK</i>                      |

... but you can see that the code needed to do some basic checking and database update is very routine stuff and of no further interest.

## Coming Soon ...

The next article(s) in the set will show how you can use the new GIF capabilities in Dyalog 8.2 to provide a graphical 'drill-down' capability from charts produced by *RainPro*. If you want to read ahead, *HTML.ZIP* on the Causeway web site has all the examples from this Vector and many of those from the next one.

## References

- [1] *The HTML Sourcebook*, Ian S. Graham, John Wiley 1995
- [2] *HTML Basics for APLers - Forms*, Vector 14.4 pp 91 - 102

# APL and the Mountain

by Anne D. Wilson

## Introduction

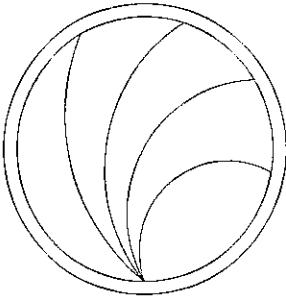
As a mathematician who loves APL, is obsessed with needlework and hooked on software engineering; it was natural to combine all my interests, so that each individual activity is used for the good of the whole. I used the mathematical functionality of APL with the trigonometry learnt in my youth and some *turtle graphic* techniques; then exploited the facilities provided by both Windows and my trusty printer to produce a system for designing patchwork, which may be of interest to mathematicians, programmers, needlework designers and students of software engineering alike. The system has been extended to draw and print all-over patchwork patterns, regular shapes, various graph papers and oriental rug designs; the next extension will be to design bobbin lace templates.

This article describes the method of producing shapes based on intersecting circles, including the derivation of the formulae used. There is a discussion on some of the design principles used in the system, and a few comments on testing. Even though this is a recreational application I have used a professional approach to analysis, programming, and testing.

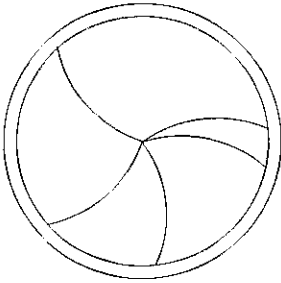
## Background

*What is this mountain?* Many years of sewing for many daughters has resulted in a boxroom full of small pieces of material; this is known as *Anne's material mountain*. As a farmer knows his herd of cows so I know my scraps; their colour, textures and patterns. I could no more bin my *mountain* than the farmer could kill his cows mindlessly. The solution is obvious - kill the cows for meat, and make my scraps into patchwork.

I decided to use some of an excellent crop of lavender to fill pin cushions for Christmas presents and two designs of intersecting circles evolved. Armed with pencil, compasses, ruler and paper I started drawing paper patterns, but this soon became tedious, so I handed the task over to my computer.



In the *equal areas* design arcs of dissecting circles divide the *main or bounding* circle; these circles each have their centre on the circumference of the main circle, but their radii vary; they have a common intersection on the circumference of the main circle. Only one item of data is required to draw each of these arcs; the radius or the position of the centre, or the angle subtended by the chord to the centre of the main circle will suffice. The parameters required are the radius of the main circle and the number of divisions.



The *equal radii* design uses arcs of circles with the same radius as the *main or bounding* circle. Their centres are at randomly generated points on the circumference of the main circle, and therefore they all intersect at the centre of the main circle. The same parameters of radius and the number of divisions are required for this design.

## Practical Details

### Computer Requirements

I used a 486 Compaq with a colour screen. Colour enhances the appearance, but adds nothing to the functionality, nor is it printed, although I use filling patterns for some designs.

The volumes of data are small, so there should be no problem with memory or space.

### Printer

The system was written using an Epson LQ500 dot matrix printer for output. This is a 24 pin dot matrix printer, with 180 dots per vertical inch, but references to any of its parameters are by way of variables. The resulting print is accurate to the nearest pixel.

### Dyalog APL

I used Dyalog APL for Windows Version 7.0, a product for which I have much praise, and little criticism. Programs take on Windows compliant features almost

automatically. Later version may have even better facilities, but I enjoy using this version. The tutorials and the presentation of information in the manuals made it a *Teach Yourself* package par excellence.

### Windows and the Clipboard

Microsoft Windows 3.1 provided many of the facilities used in Dyalog APL and controls output to the printer. Every attempt has been made to produce a system conforming to Microsoft standards. Output to the Clipboard facilitates export and import of the pictures, but I have problems with *cropping* and the circularity of exported shapes.

MDI (multiple document interface) allows many windows to be opened simultaneously, which is useful for a patchwork designer.

### Mathematical Calculation

#### Equal area design

It seemed sensible to have all the pieces (a technical patchwork term with an obvious meaning) of equal area. Working out the necessary equation I soon realised there was no simple solution and that an iterative method was required.

The main circle is divided by a series of  $(n-1)$  arcs into equal areas. The problem can be reduced to finding a series of arcs which cut off areas of

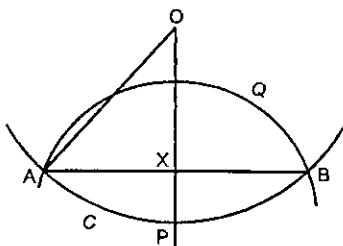
$$\pi R^2 \times (1/n, 2/n, \dots, (n-1)/n)$$

The main circle is                      C centre O radius R

The dissecting circle                Q centre P radius r

Circle Q goes through A, the common intersection point for all the circles

The advantage of using radians is that all references to  $\Pi$  are contained within the definition of the angles, not in length and area formulae.



Let angle AOB =  $\alpha$

Angle APB is subtended on the circumference of P

$$\therefore \text{angle APB} = \theta = \pi - \alpha/2$$

$$OA = OB = R$$

$$PA = PB = r$$

$$AB = 2 R \sin \alpha/2 = 2 r \sin \theta/2 = 2 r \cos \alpha/4$$

$$\therefore r = 2 R \sin \alpha/4$$

$$\text{Area of } \triangle AOB = \frac{1}{2}(2R \sin \frac{\alpha}{2}) \times (R \cos \frac{\alpha}{2}) = \frac{1}{2} R^2 \sin \alpha$$

$$\text{Area of sector AOB} = \frac{1}{2} R^2 \alpha$$

$$\therefore \text{area segment ACB} = \frac{1}{2} R^2 \alpha - \frac{1}{2} R^2 \sin \alpha = \frac{1}{2} R^2 (\alpha - \sin \alpha)$$

$$\therefore \text{area segment AQB} = \frac{1}{2} r^2 \theta - \frac{1}{2} r^2 \sin \theta = \frac{1}{2} r^2 (\theta - \sin \theta)$$

The required area is

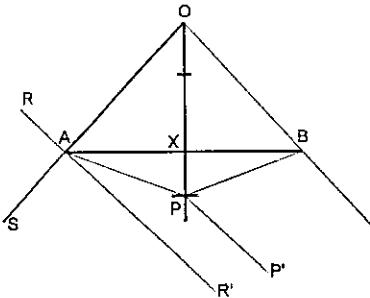
$$\frac{1}{2} R^2 (\alpha - \sin \alpha) + \frac{1}{2} r^2 (\theta - \sin \theta)$$

Two iterations are used:

- outer loop for once for each dissecting arc;
- inner loop for successive approximations to find an angle giving an acceptable area.

Specifying the ends of the arcs requires further geometric calculations. For each circular arc the angles subtended from the centre of its circle needs to be calculated.

The line OA is taken as one axis of a co-ordinate system with origin at A. RAR' is drawn at right angles and gives the direction from which the angles are measured. PP' is parallel to RAR'. Note angle SAP is an external angle to triangle AOP,



$$\begin{aligned}\text{Angle R'AP} &= \text{SAP} - \pi/2 = \pi - \alpha/2 - \theta/2 - \pi/2 \\ &= \pi/2 - \alpha/2 - (\pi/2 - \alpha/4) = -\alpha/4\end{aligned}$$

Direction PA = angle P'PA =  $\pi + \alpha/4$

$$\text{angle P'PB} = \text{P'PA} - \theta = \pi + \alpha/4 - (\pi - \alpha/2) = 3\alpha/4$$

I am constantly amazed at how geometric values simplify - would you ever guess that one of these directions would be three quarters of the angle at the centre?

### Equal radii design

The algorithm for the *equal radii* design is simpler. A geometric property of such circles is:

if O1 the centre of circle C1 lies on the circumference of C2 then O2 the centre of C2 lies on the circumference of C1.

To specify  $n$  circular arcs I generated  $n$  random numbers in the range  $0.....2\pi$  ensuring they were not too close together:

$$\pi/2 \times (n - 1) / n$$

The remaining code was lifted from the *equal area* function, see below on copying code.

### System Design and Testing

Testing should be incorporated into the entire life cycle of a system; much as breathing is incorporated into our entire lives. The advent of personal computing and the ability to trace the execution of the program makes possible a very immediate ongoing testing environment. All low level functions should be rigorously tested, and re-used as *black boxes*. If a system has complex logic this also must be rigorously tested; but this is a simple system and the menu bar acts as the high level design (see below) which makes it almost impossible to introduce errors at that stage!

Design is still important, especially:

- at the top level of design, at one time the job of the systems analyst;
- when coding complex algorithms.

Indeed it is very important to me to have my own systems easily alterable as I may not look at them for years and no-one else knows anything about them.

The use of Windows has to some extent removed the top levels of design. Whereas, once the programmer coded the initial entry into a system, and organised the relationships between different modules, and the flow of control around the system; with Windows this *just kinda happens*. The culture shock in coming from large mainframes to Windows lay mainly in the multitasking feel of Windows. Database systems address the problem of whether the customer can wait until tomorrow to see the effect of today's withdrawal on his balance; this is the same problem as whether one open window needs to reflect a change in another open window.

Copying code across from one system to another results in **standardised** systems; copying from one function to another leads to standardisation within a system; OOPS adopt this idea. Of course, it saves on coding time, on testing and on planning; the downside comes if the original version is not good enough. Another, and usually better, method is to have low level functions for frequently used code; these will characteristically be called from several places; I visualise these as my dialect of the language, i.e. of AdwPL. The technical principal behind this is **bottom up development**, as Myers [1] says "*Bottom-up development and top-down development are implementation, not design, strategies*". I work on the principle – the first time I code directly into the function, the second time I think it would be a good idea to have it separate, and the third time I pick the code out from the earlier two and put it into a separate function. Such functions should be rigorously tested when written, thereafter they will receive repeated testing as they are used by higher functions. The developer should be aware of any opportunities to include an OO (object oriented) approach.

One advantage of writing systems for oneself is that you can put time and effort into making it user friendly, and error free, rather like *running in a new engine*. Paper and pencil designs are very necessary for the overall design of a system, but they fail to *rub the corners* off when it comes to using a system. It is only when using it that you realise what would make life easier, and what is annoying to the user; for example I once added a function DISPLAY whose only action was to call DISPLAY because I frequently mistyped! But all alterations must be fully tested, and must **not** impact on the existing code. Currently I am trying to introduce keyboard only systems as my mouse is dead; anyway I prefer using fingers to pushing mice.

### First Steps in Design

Knowing some of the functions that I want gives me an idea of the top level functionality of the future system. Rather than write this down I let the *menu bar* of Windows act as

**Documentation** – both using the menu item, and the *microhelp* for more details.

**Top Down design** – by starting with the menu bar, and adding items under each level, although this is only suitable for 3 levels of subdivision.

**Standardising** – using the *File, Quit, Edit* etc. that are familiar to all Windows users.

**Aide de memoir** – storing stubs, without code, for functions I am still planning to include.

While developing and testing the system I test and refine the items included in the menu bar. It is necessary to restrict the levels in this imaginary *top down design* tree to:

1. main menu bar;
2. drop down menuitems;
3. *separators* between associated menuitems;
4. second level of menuitems (used with discretion).

If used, MDI windows should be part of the original design, otherwise all references to window objects may have to be altered if they are included in later. But once you have used them you will never want to go back to single window systems!

### Variables

Data, including variables, has a natural property of *hardness* which goes from *soft* for data which is updated, such as names and addresses, to *rockhard*, i.e. carved in stone (or rom), for items which can never be altered, such as the name of the original author of a system. In between are variables which control the execution of the program - screen colours, printer characteristics, the last setting of variables. The way in which these variables are included, and the method of altering them has repercussions on security considerations, on the ease with which a system can be used, and on the ease with which it can be changed to meet changes in circumstances. Below are notes on how some variables are treated, given in order of hardness, starting with the hardest.

**Hard coded variables.** This is the most difficult variable to amend. Looking at some (of my) code I noticed the vector 0 0 255, setting a particular colour for my computer, it may not be valid for another; far better to assign the value into a variable with an obvious name like *Red Green or Blue*. (N.B. must change my program.) That was bad! If the system requires that variables are coded into it, perhaps values for fonts of different size and weight, they should be coded in a function called by the first executed function, with meaningful names; then both function and variables are easily found if an alteration is required.

**Administrative Variables.** Use a menuitem entitled **Options** or **Administration**. I like to display a form which contains environmental variables about printers, page sizes, how to save data, scrollbars etc. It is customary to have default values which can be used to reset values; these may be coded into the program. Values are saved onto a file, updating occurs both when the values are altered, and on exit from the system. The file is read when the system is fired up; default values being used if the file is not available. If each user has their own file then they can create their own environment.

**Local Variables.** When drawing the circular designs for my patchwork there are 2 variables which will be changed frequently viz. the diameter of the circle for printing, and the number of dissecting arcs. Both of these have a limited range of values, and I have made use of scroll bars, which appear in the same window as the diagrams. Scroll bars restrict the user to entering only valid data - which means no validating in the program (hurrah!)! Values specifying the maximum, minimum and step sizes can themselves be held as variables; note that they will be *harder* than the variables they are describing. The scroll bar variables are stored on my personal file; this removes the annoying habit computers have of forgetting what you did last time. The appearance of scroll bars should follow a logical pattern, so that the user knows what to expect from past experience - standardisation again.

## Problems

Problems I encountered were due to inexperience on my part. My concentric circles were not concentric, and my mutually intersecting circles did not intersect until I assigned to the *Coord* property the value of *Pixel*. This applied to screen, printer and clipboard.

With Windows there are implementation restrictions on the number of objects that can be sent to the screen. Dyalog APL processes multiple arcs and *polylines* very efficiently, but if they are written individually the Windows processing time can be unacceptable for screen, printer and clipboard. Whilst this involves the encoding problem it does save loops, so must be a good habit; I am not alone in having problems with encoding and decoding - will I ever crack it?

## Sewing

An article about patchwork would be incomplete without details of making the pincushions. Having printed off the shapes, cut out the resulting paper pattern, allowing for turnings. If non-frayable material is used turnings are only needed on one side. Pin the pieces together, overlapping adjoining pieces, Hem neatly. Embroider with herringbone stitch over the joins. Cut out a larger circle for the sides and bottom. Run a gathering thread around the edge, and gather to the diameter of the top. Cut out 2 rounds of wadding, pack the lavender in the centre. Join bottom, top and wadding together, and smell!

## Reference

- [1] Glenford J. Myers: *Reliable Software through Composite Design*: Van Nostrand Reinhold 1975

---

# TECHNICAL SECTION

This section of VECTOR is aimed principally at those of our readers who already know APL. It will contain items to interest people with differing degrees of fluency in APL.

## Contents

|                                           |                   |     |
|-------------------------------------------|-------------------|-----|
| Hackers' Corner: How Big is that Picture? | Adrian Smith      | 128 |
| Technical Correspondence                  | Sutton, McDonnell | 130 |
| At Play with J: to Summarize              | Gene McDonnell    | 132 |
| Utilities for the Dyalog Session          | Ray Cannon        | 138 |

---

# Hacker's Corner: How Big Is That Picture?

by Adrian Smith ([causeway@csi.com](mailto:causeway@csi.com))

## The Problem

People have been badgering me for some time to add support for Windows metafiles to *RainPro* and *NewLeaf* – very handy for company logos or fancy graded fills in true PowerPoint style. I made the mistake of getting cornered at Rome, and compounded the error by saying how simple it would be to put this stuff in. Then I discovered one minor problem – most modern metafiles have a 'placeable header' which gives the designed size of the image; Dyalog lets you set this property for images which you generate, but always reports (0 0) as the 'RealSize' of files which already exist. Hmmm.

## The Documentation

Here is what is says in the Windows SDK help file:

A placeable Windows metafile is a standard Windows metafile that has an additional 22-byte header. The header contains information about the aspect ratio and original size of the metafile, permitting applications to display the metafile in its intended form.

The header for a placeable Windows metafile has the following form:

```
typedef struct {  
    DWORD    key;  
    HANDLE   hmf;  
    RECT     bbox;  
    WORD     inch;  
    DWORD    reserved;  
    WORD     checksum;  
} METAFILEHEADER;
```

Following are the members of a placeable metafile header:

|            |                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------|
| <b>key</b> | Specifies the binary key that uniquely identifies this file type. This member must be set to 0x9AC6CDD7L. |
| <b>hmf</b> | Unused; must be zero.                                                                                     |

|                 |                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bbox</b>     | Specifies the coordinates of the smallest rectangle that encloses the picture. The coordinates are in metafile units as defined by the inch member.    |
| <b>inch</b>     | Specifies the number of metafile units to the inch. To avoid numeric overflow, this value should be less than 1440. Most applications use 576 or 1000. |
| <b>reserved</b> | Unused; must be zero.                                                                                                                                  |
| <b>checksum</b> | Specifies the checksum.                                                                                                                                |

Clear as mud, really. We can make a few wild guesses about that 'bbox' data type, so let's take the usual APL approach, read the first 22 bytes of the file as integers and just flail around until the numbers look sensible!

## The Implementation

The objective was to make a function which takes the name of a metafile, detects the placeable header and returns the size in (x,y) co-ordinates to the nearest 1/10th of a point (1 point = 1/72 of an inch). Here is the final version:

```

vr←metasize file;magic;hdr;tn;inch
[1]  ⍠ Find the size of a placeable WMF from the 22-byte header
[2]  ⍠ Returns size in points (to nearest 1/10th point)
[3]  ⍠ If there is no placeable header, just return 1" square.
[4]  r←72 72 ⍠ magic+'9AC6CDD7' ⍠ tn←1+⌊0,⊞NNUMS
[5]  file,+(~'.\file')\'.wmf'
[6]  file ⊞NTIE tn
[7]  hdr←⊞NREAD tn 83 22 0
[8]  ⊞NUNTIE tn
[9]  ⍠ Convert to unsigned ...
[10] ((hdr<0)/hdr)+((hdr<0)/256+hdr
[11] ⍠ Check against the magic word ...
[12] :If magic^.=,⊆⊆'0123456789ABCDEF'[1+16 16⌈4+hdr]
[13]   inch+256⌊⊆2+14+hdr ⍠ Generally 1000
[14]   r+256⌊⊆4 2⊆8+6+hdr ⍠ Remember the flip (words reversed)
[15]   r←-⌈⊆2 2⊆r
[16]   ⍠ Fix wraparound at 2*16 ...
[17]   ((r<0)/r)+((r<0)/(2*16)+r
[18]   r+0.1×⌊0.5+720×r+inch
[19] :End
v

```

The only nasty trip wires are addressed by line [10] and line [17] which handle the wraparound and return unsigned values. Other than this, it is simple and reliable, but do remember all those 'reverse' symbols to get your egg big-end up!

# TECHNICAL CORRESPONDENCE

## Random Link Generator

From: John Sutton

26 June 1998

In Eugene McDonnell's piece on a Random Link Generator (*Vector* Vol.14 no.4 pp. 122-127) he refers to the treatment of an idea of G.J. Mitchell and D.P. Moore in several editions of Knuth's book *Seminumerical Algorithms* (vol.2 of *The Art of Computer Programming*).

My copy of the 1969 edition has, at the end of the preface on p.vi, an Author's note "In this second printing, I have incorporated several hundred improvements to the text and exercises while retaining the original page numbering." In the index there are entries for Mitchell, Gerard Joseph Francis Xavier and Moore, Donald Philip which both point to page 464. That page is in the 'Answers to Exercises' and contains a brief mention of their 1958 idea for the case that  $m$  is a power of 2. I enclose a copy of page 464.

The results of this exercise...suggest[s] the following way to obtain the efficiently computable sequence with a very long period... If  $x^k + x^j + 1$  is a primitive polynomial modulo 2, the sequence defined by  $X_n = (X_{n-j} + X_{n-k}) \bmod 2^*$  has a period length between  $2^k - 1$  and  $2^{k-1}(2^k - 1)$ , if  $X_0, \dots, X_{k-1}$  are not all even. For example, the sequence defined by  $X_n = (X_{n-24} + X_{n-53}) \bmod 2^*$  will have a period length of at least  $2^{55} - 1$ , which is the period of its unit digits, and the actual period length will probably be much larger. This idea, due to G.J. Mitchell and D.P. Moore in 1958 (unpublished), has been used with apparent success to generate full-word random numbers; the sequence clearly has some locally nonrandom subsequences which hardly ever occur. When  $k = 127$  we may take  $j = 1, 7, 15, 30, 63, 64, 97, 112, 120$  or 126.

I don't have access to copies of the other editions but it is interesting that Knuth's thinking on this had begun to change even by the second printing which appears to be in the same year.

John Sutton  
16a Overland Road  
Mumbles Swansea SA3 4LP

From: Eugene McDonnell

26th August 1998

Kudos to Mr. Sutton for having the diligence to do what I should have done, that is, to look in the index of Knuth's *Seminumerical Algorithms* for the names Mitchell and Moore. Had I done so, I would have seen that the names were even in my copy, which was the first printing of the book, along with the text cited by Sutton. My attention, however, was paid to Knuth's caution in the main part of the book in Section 3.2.2, where he says, "At the present time, not very much is known about such an additive number generator; before its use can be recommended, it will be necessary to develop the theoretical results necessary to prove certain desirable randomness properties..."

It is also of interest to note that between 1969 and 1993 the algorithm had changed from an additive one to a subtractive one, whence the name "the subtractive method", used in Knuth's *The Stanford Graph*, Section GB\_FLIP 4.

Eugene McDonnell  
Palo Alto  
eemcd@aol.com

---

# At Play with J: to Summarise

*by Gene McDonnell*

The key adverb in J, represented by slashdot ( $/.$ ) is defined in the Dictionary as:

$x\ u/.y$  is  $(=x)\ u@#y$

that is, items of  $x$  specify keys for corresponding items of  $y$  and  $u$  is applied to each collection of  $y$  having identical keys. For example:

1 2 3 1 3 2 1  $</.$  'abcdefg'

|     |    |    |
|-----|----|----|
| adg | bf | ce |
|-----|----|----|

This may be clearer if we look at the separate parts.

$=x$

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |

The first row of this has 1s in the first, fourth, and seventh positions, so when used as the left argument to copy (the dyad of  $\#$ ), and applied to  $y$ , yields its first, fourth, and seventh items, or 'adg'; similarly the second row yields 'bf' and the third row yields 'ce'. Each of these is then boxed and the three are catenated together, yielding

|     |    |    |
|-----|----|----|
| adg | bf | ce |
|-----|----|----|

The basic idea remains the same when  $u$  changes from box to a different monad. For example, if we replace box by tally (the monad of  $\#$ ) we get:

$x\ \#/.y$   
3 2 2

The same three groupings are selected, but instead of being boxed they are tallied, or counted, yielding the count of each group; three in the first, and two in the second and third.

The key adverb was not in the initial version of J. It came in later at the request of the J user community, notably Joey Tuttle. Joey's interests were not merely theoretical; he had practical ends in view. He was in the business of analysing huge amounts of data and summarising time and amount fields by accounts. We'll use an abbreviated version so we can fit the data onto a small page. Suppose we have three accounts, 1001, 1002, and 1003, and suppose further that we have a table whose rows give an account number and an amount:

```
acct=:1001 1002 1003
table=:((?10#3){acct),. ?10#100
table
1001 51
1003 83
1002 3
1002 5
1001 52
1001 67
1003 0
1003 38
1003 6
1002 41
```

To summarise this table by account, we transpose (|:) it, so that the accounts are in the first row, and the amounts in the second, then insert (/) sum key (+//.) between the account row and the amount row:

```
+//./|:table
170 127 49
```

If you check the first amount in the sum, 170, you can verify that it is indeed the sum of the three amounts associated with the first occurring account, 1001, that is, it is the sum of 51, 52, and 67. Similarly the second amount 127, is the sum of the four amounts associated with the second occurring account, 1003, that is, it is the sum of 83, 0, 38, and 6. Lastly, the third amount, 49, is the sum of the three amounts associated with the third occurring account, 1002, that is, 3, 5, and 41.

This result may not be completely satisfactory, since the amounts are not in the order of the accounts: they are in the order in which they fortuitously occur in the table. One way to remedy this is to place some dummy rows at the beginning of the table, one for each account, with the accounts in the desired order, and with the amounts set to zero (acct,. 0).

```

(acct,.0),table
1001 0
1002 0
1003 0
1001 51
1003 83
1002 3
1002 5
1001 52
1001 67
1003 0
1003 38
1003 6
1002 41

```

Now when we summarise the amounts will be in account order.

```

+//.//:(acct,.0),table
170 49 127

```

We can produce a summary of accounts and amounts by prefacing the above with the list of account numbers and stitching (.,) the lists together.

```

acct,.,+//.//:(acct,.0),table
1001 170
1002 49
1003 127

```

This gives you the theory and the practice of the key adverb, so it's time to play, and incidentally to learn another way to use key.

How are the digits of *pi* distributed? If the digits were distributed evenly, then the frequency of occurrence of all digits would be about 10%. J enables you to compute as many digits of pi as you have room for and time for. A convenient way to obtain *n* digits of pi is to subtract 1 (<:) from *n*, make this an extended integer (x:) use this as an exponent of 10 (10^), apply floor atop pi times (<.@o.) and take the format (":) of this:

```

dp=: monad def '":<.@o.10^x:<:y.' NB. digits of pi

```

Try this on a small integer:

```

q10=: dp 10
q10
3141592653

```

This is correct. Try it on a somewhat larger integer:

```
q30=: dp 30
q30
314159265358979323846264338327
```

Checking this against the value of pi to many places in a table such as may be found in a volume of Knuth's *The Art of Computer Programming* shows that q30 is accurate, too.

Let's compute some more (q3000 may take several minutes):

```
q100=: dp 100
q300=: dp 300
q1000=: dp 1000
q3000=: dp 3000
```

Now let's see how the digits are distributed in each of these, in order. We need a digit distribution function. This is where a new use of key comes in. In order to make the result of the digit distribution function be in the right order, we'll preface the argument with d, a list of the decimal digits in order.

```
d=: '0123456789'
```

To get the distribution we preface the formatted digits of pi with d (d.y.), then apply count (#) key (/.) reflexive (~) to this, giving us the distribution of d.y., then subtract 1 (<:) to adjust the count for the presence of d.

```
dd=: monad def '<: #/.~d.y.'
```

And try this out on q10, which is easy to verify by eye:

```
/:~q10
1123345569
dd q10
0 2 1 2 1 2 1 0 0 1
```

No zeros, two ones, one two, two threes, one four, two fives, one six, no sevens or eights, and one nine. Now let's see the digit distribution of each of the other lists of pi digits.

```

dd q30
0 2 4 7 3 3 3 2 3 3
dd q100
8 8 12 12 10 8 9 8 12 13
dd q300
26 30 35 31 37 27 31 19 34 30
dd q1000
93 116 103 103 93 97 94 95 101 105
dd q3000
259 308 303 266 318 315 302 287 310 332

```

These look somewhat reasonable, but it would be better to see how closely each gets to having 10% of each digit, using a function `pd`, which takes a distribution as argument, and yields the percentage of each value, rounded to the nearest one per cent. Do this by dividing the values by the sum of the values (`y.%+/y.`), multiply this by 100 (`100*`), to get percentages, and round, getting the nearest percentage, by adding a half (`0.5+`) and taking the floor (`<.`).

```

pd=: monad def '<.0.5+100*y.%+/y.'
pd dd q10
0 20 10 20 10 20 10 0 0 10

```

We can compare this to `dd q10` and see that it is simply the same values multiplied by 10 to give percentages, as desired. There are so few digits to take into account that it is difficult to say whether the distribution is even or not. Trying the next distribution, of thirty values, still leaves us uncertain.

```

pd dd q30
0 7 13 23 10 10 10 7 10 10

```

There are no zeros among the first thirty digits, and a lot of threes. Probably still not enough digits.

```

pd dd q100
8 8 12 12 10 8 9 8 12 13

```

Except for the large number of nines, this is beginning to look quite even.

```

pd dd q300
9 10 12 10 12 9 10 6 11 10

```

Here, the number of sevens seems too low. Let's keep looking.

```

pd dd q1000
9 12 10 10 9 10 9 10 10 11

```

Ones seem a bit high, but I'd say this distribution is even enough.

```
pd dd q3000
9 10 10 9 11 11 10 10 10 11
```

With 3000 digits to distribute, we can say with some satisfaction that this represents an even distribution.

Before we part, let's look at a consecutive portion of these digits:

```
(762+i.6){q1000
999999
```

Hmmm. Well, yes, that's not *too* unusual. In fact, if such strings *didn't* occur every now and then, it would argue against randomness.

---

# Utilities for Dyalog Developers

by Ray Cannon (Ray\_Cannon@compuserve.com)

One thing I missed when I first started working with Dyalog APL under Windows after many years of STSC/Manugistics mainframe APL, was *User Commands*.

In an attempt to fill the gap left by Dyalog's lack of a User Command Processor, I have developed a technique to run user-written commands from within the `⎕SE` namespace, (normally from the toolbar or menu bar or PF key) without having to save them permanently in the "session namespace" (`⎕SE`).

## Overview

I have a workspace called `SE.DWS` which is always loaded when I first start the development version of Dyalog APL. This WS copies a very few utility functions into `⎕SE`, sets up my session Toolbar, session Menu, PF keys, dynamic page width, an external variable (to act as an APL dedicated "clipboard") and then loads any requested workspace (either named in a command line parameter or as the workspace selected via File Manager).

The most important function added to the `⎕SE` namespace is `Ucmd`. This takes a character string as a right argument and an optional left argument (which will be explained later).

```

v r+{ltarg}Ucmd
  string;niladic;bo;cmd;arg;bin;file;wsid;namespace;tempns
[1]  aEmulate STSC_s r+⎕UCMD string
[2]  bo+^string* ' ' aMust use space delimiter
[3]  cmd+bo/string
[4]  niladic+^/bo
[5]  arg+1+(-bo)/string
[6]  aCopy command to local namespace "tempns"
[7]  namespace+'tempns'
[8]  bin+namespace ⎕NS''
[9]  wsid+(2 ⎕NQ'.' 510 'DIALOG'),'\\ws\\ucmd\\',cmd
[10] copyinto
[11] arun main function in cmd with correct valence
[12] :If 2=⎕NC'ltarg' ⋄ ⧿r+ltarg tempns.',cmd,' arg'
[13] :ElseIf ~niladic ⋄ ⧿r+tempns.',cmd,' arg'
[14] :Else ⋄ ⧿r+tempns.',cmd
[15] :End

```

v

The string consists of the name of the command to be processed and any arguments. (For example *Fns a\** where *Fns* is the command and *a\** the argument. Note that the case of the function name is important.)

*Ucmd* parses its arguments and then creates a temporary (i.e. localised) namespace called `SE.tempns`.

*Ucmd* then locates a workspace with the same name as the command from a standard known location. (I use the value of Dyalog's configuration parameter *DYALOG* as the root and add to this the extra path required to locate the *Ucmd* utility workspaces. e.g. *C:\DYALOG73\ws\ucmd\Fns.dws*. Note that the case of the function name, now used as the workspace name, is unimportant here.)

*Ucmd* then copies into the `SE.tempns` namespace the contents of a workspace it just found. It then runs a function with the same name (and case) as the command with the supplied argument.

This means that a single workspace (say *FNS.DWS*) could contain several different function commands (say *FNS*, *Fns*, *FnS*, *fns* etc.). I use this technique to call several different cover functions to a basic command. (i.e. to list the functions in the current namespace; list functions in the *ROOT* namespace; list functions in the namespace putting the result in the clipboard; list all functions matching a set of criteria read from the calling window.....)

I have set up my Toolbar, PF keys and Menus to support calling the *Ucmd* function in a number of different ways:

- a) Single click menu item. This calls the command directly, if appropriate, using the "currently selected object" as an argument. Due to limited space on the menu bar, only the most useful commands are supported in this manner. e.g.:
  - **NSfind** - To locate all occurrences of the current object in the current namespace.
  - **Latest** - To list all functions modified today.
  - **Fns** - To list all fns in the current namespace.
  - **Vars** - To list all vars in the current namespace.
  - **Put1** - Adds a copy of the "currently selected object" to the "APL Clipboard".
  - **GetAll** - Fixes all objects held in the "APL Clipboard" into the current namespace.

b) Select a command from a drop-down list, getting any required parameters via a window. *e.g.*:

- **FNS** – Input pattern-matching selection criteria of function (and operator) names, and name of namespace to search and then list all functions (and operators) found.
- **VARS** – as for FNS but on variables.
- **OBS** – as for FNS but on other objects.
- **COMP** – Input the (fully qualified) names and locations (default to the current workspace) of the objects to be compared.
- **WSCOMP** – Input the name of a workspace to compare to the current workspace, and list differences found at the object name level.
- **WSFIND** – Locate the text in the workspace/namespace, with full control over what and how the search is performed (*e.g.* ignore case, in comments only).
- **GETCLIP** – Store the results of the last command in the (supplied) named variable.
- **TREENS** – List the namespace/GUI object tree structure from the (supplied) named node (default #).
- **COPYUTILS** – Copy from the named workspace the named function (and all other functions it calls) into the current namespace.

c) Press a PF key to perform standard requests. *e.g.*:

```
□SE.Ucmd tempns SE.Ucmd ' Vars dbx_* '
```

Lists all variables starting with *dbx\_* (i.e. Causeway panel definitions).

**NOTE.** The results of the commands are often “unformatted” (*e.g.* a simple text matrix, with one item per row). This is what would be stored by GETCLIP above. However, when actioned from the Toolbar or Menu bar, the results displayed in the session are always “formatted” (*e.g.* a long list would be “folded” into several columns across the page, the number of columns depending on *□PW* (which in turn is dynamically linked to the width of the session window). The function *□SE.Show* performs this action.

Recap: the command

```
□SE.Ucmd 'Fns a*'
```

results in the contents of workspace C:\dyalog73\ws\ucmd\fns.dws being copied into temporary namespace □SE.tempns. It then executes the following string:

```
□SE.Show □SE.tempns.Fns 'a*'
```

Upon completion, the local namespace is removed when the parent function *Ucmd* is exited.

I have placed a zipped file containing my most up to date UCMD directory (..\ws\ucmd\\*.\*) including the all important *SE.DWS* on the *Vector* web site; access is via the Contents page listing for this edition of *Vector*. Once unzipped and stored under your Dyalog directory, (you will need to add the UCMD directory to your WSPATH,) then the action of )LOAD *SE* should modify your current session making all the *Ucmd* utilities available.

## Notes

There is no need to save your modified session, as *SE* is designed to modify it each time it is loaded. Take special care not to overwrite your default session with the modified one (I have overwritten DEF\_UK.DSE more than once!).

If you ever use file manager to select a workspace to run under Dyalog, the REGEDIT file SE.REG, can be used to modify your registry so that Dyalog automatically loads the *SE* workspace before loading the selected workspace. The file supplied is for Windows 3.x and will need to be modified for your correct paths. I am unable to supply a Win95/NT version of this file, so you will need to modify the appropriate registry entry by hand (via the program REGEDIT.EXE).

The *SE* workspace has its own built in defaults, some of which may be modified by adding entries into the Dyalog APL .INI or registry, thus:

```
[UCMD]
;used by SE.DWS if found in APL.INI or the registry to override its own defaults
;"ops" Used to select and order the UCMD commands on the ToolBar / MenuBar
ops=nsfind fns latest from vars put get combo
;"state" Determines the windows size during loading
;state=Minimize Maximize
;"size" Determines the final size of the Dyalog window. In pixels
size=600 800
;"posn" Determines the initial position of the Dyalog window. In pixels
posn=0 0
;Note if required, you can use either ("state") or ("size" and "posn") but not both.
```

## Disclaimer and Copyright Notice

I did not write all the material contained in the zipped file.

In particular the workspaces *COMP* and *WSFIND* contain functions written by Rex Swain. Dyalog distributes them with its APL in the Outerproducts TOOLS directory. I have merely added cover functions to enable the *UCMD* interface to work.

The workspace *FILECHK* was explicitly written by Bob Hoekstra for use with this interface.

Sorry, but because of the dynamic nature of these commands (I am constantly making changes, fixing bugs, trying out new commands etc.) the help file is unlikely to be fully up to date.

The code is supplied without any warranties or support. If you have problems with it, please let me know and I may (time permitting) try to fix it. If you write any commands of your own, you may send them to me, and I will try to make them available via the *Vector* web site.

## Index to Advertisers

|                     |    |
|---------------------|----|
| Dyadic Systems Ltd  | 6  |
| Russian Fund Appeal | 80 |
| Strand              | 2  |
| Vector Back Numbers | 16 |

All queries regarding advertising in VECTOR should be made to Gill Smith, at 01439-788385, Email: [apl385@compuserve.com](mailto:apl385@compuserve.com).

### Submitting Material to Vector

The Vector working group meets towards the end of the month in which Vector appears; we review material for issue n+1 and discuss themes for issues n+2 onwards. Please send the text of submitted articles (hardcopy with diskette as appropriate) to the Vector Working Group via:

Vector Administration, c/o Gill Smith  
Brook House  
Gilling East  
YORK, YO6 4JJ  
Tel: +44 (0) 1439-788385  
Email: [apl385@compuserve.com](mailto:apl385@compuserve.com)

Authors wishing to use Word for Windows should contact Vector Production for a copy of the APL2741 TrueType font, and a suitable Winword template. These may also be downloaded from the Vector web site at [www.vector.org.uk](http://www.vector.org.uk)

Camera-ready artwork (e.g. advertisements) and diskettes of 'standard' material (e.g. sustaining members' news) should be sent to Vector Production, Brook House, Gilling East, YORK YO6 4JJ. Please also copy us with all electronically submitted material so that we have early warning of possible problems.

# British APL Association: Membership Form

Membership is open to anyone interested in APL. The membership year normally runs from 1st May to 30th April, but new members may join from 1st August, November or February if preferred. The British APL Association is a special interest group of the British Computer Society, Reg. Charity No. 292,786

Name: \_\_\_\_\_  
 Address: \_\_\_\_\_  
 \_\_\_\_\_  
 Postcode / Country: \_\_\_\_\_  
 Telephone Number: \_\_\_\_\_  
 Email Address: \_\_\_\_\_

Category (please tick box) to run from: 1st May ☐ August ☐ Nov ☐ Feb ☐

|                                                    |      |                          |
|----------------------------------------------------|------|--------------------------|
| UK private membership .....                        | £12  | <input type="checkbox"/> |
| Overseas private membership .....                  | £14  | <input type="checkbox"/> |
| Airmail supplement (not needed for Europe) .....   | £4   | <input type="checkbox"/> |
| UK Corporate membership .....                      | £100 | <input type="checkbox"/> |
| Corporate membership overseas .....                | £135 | <input type="checkbox"/> |
| Sustaining membership .....                        | £430 | <input type="checkbox"/> |
| Non-voting UK member (student/OAP/unemployed only) | £6   | <input type="checkbox"/> |

## PAYMENT — in Sterling or by Visa/Mastercard/JCB only

Payment should be enclosed with membership applications in the form of a UK Sterling cheque to "The British APL Association", or you may quote your Mastercard, Visa or JCB number.

I authorise you to debit my Visa/Mastercard/JCB account

Number:         Expiry date:  |

for the membership category indicated above,

- ☐ annually, at the prevailing rate, until further notice  
☐ one year's subscription only

(please tick the required option above)

*Data Protection Act:  
 The information supplied may be  
 stored on computer and processed  
 in accordance with the registration  
 of the British Computer Society.*

Signature: \_\_\_\_\_ Send the completed form to:

British APL Association, c/o Rowena Small, 8 Cardigan Road, LONDON E3 5HU, UK

## The British APL Association

The British APL Association is a Specialist Group of the British Computer Society. It is administered by a Committee of officers who are elected by a postal ballot of Association members prior to the Annual General Meeting. Working groups are also established in areas such as activity planning and journal production. Offers of assistance and involvement with any Association matters are welcomed and should be addressed in the first instance to the Secretary.

### 1998/99 Committee

|                           |                                                                    |                                                                                                |
|---------------------------|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Chairman                  | Anthony Camacho<br>0117-973 0036<br>100612.1057@compuserve.com     | 11 Auburn Road, Redland,<br>BRISTOL, BS6 6LS                                                   |
| Secretary                 | Ajay Askoolam<br>01737-771643<br>106173.3347@compuserve.com        | 42 Hanworth Road, Redhill,<br>Surrey RH1 5HT                                                   |
| Treasurer                 | Nicholas Small<br>0181-980 7870<br>treas.apl@bcs.org.uk            | 8 Cardigan Road,<br>LONDON E3 5HU                                                              |
| Journal Editor            | c/o Vector Administration<br>01439-788385<br>apl385@compuserve.com | Gill Smith, Brook House,<br>Gilling East<br>YORK YO62 4JJ                                      |
| Projects and<br>Publicity | Dr Alan Mayer<br>01792-205678x4274<br>a.d.mayer@swansea.ac.uk      | European Business Management School,<br>Swansea University,<br>Singleton Park, SWANSEA SA2 8PP |
| Webmaster                 | Ray Cannon<br>01252-874697<br>100430.740@compuserve.com            | 21 Woodbridge Road,<br>Blackwater, Camberley,<br>Surrey GU17 0BS                               |
| Activities                | Jon Sandles<br>01904-612882<br>jon_sandles@compuserve.com          | 138 Burton Stone Lane,<br>YORK YO3 6DF                                                         |
| Education                 | Dr Ian Clark<br>01388-526803<br>100021.3073@compuserve.com         | 9 Hill End, Frosterley<br>Bishop Auckland<br>Co. Durham DL13 2SX                               |
| Administration            | Rowena Small<br>0181-980 7870<br>treas.apl@bcs.org.uk              | 8 Cardigan Road,<br>LONDON E3 5HU                                                              |

### Journal Working Group

|               |                                                                                                                                                                                                                          |                                                |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Editor:       | Vacant Post                                                                                                                                                                                                              | see above                                      |
| Production:   | Adrian & Gill Smith                                                                                                                                                                                                      | Brook House, Gilling East, YORK (01439-788385) |
| Advertising:  | Gill Smith                                                                                                                                                                                                               | Brook House, Gilling East, YORK (01439-788385) |
| Support Team: | Jonathan Barman (01488-648575), Richard and Adam Weber (0121-3546550),<br>Anthony & Sylvia Camacho, Ray Cannon (01252-874697), Marc Griffiths (01653-691745),<br>Bob Hoekstra (01483-771028), Jon Sandles (01904-612882) |                                                |

Typeset by APL-385 with MS Word 5.0 and GoScript

Printed in England by Short-Run Press Ltd, Exeter

## VECTOR

VECTOR is the quarterly Journal of the British APL Association and is distributed to Association members in the UK and overseas. The British APL Association is a Specialist Group of the British Computer Society. APL stands for "A Programming Language" — an interactive computer language noted for its elegance, conciseness and fast development speed. It is supported on most mainframes, workstations and personal computers.

## SUSTAINING MEMBERS

The Committee of the British APL Association wish to acknowledge the generous financial support of the following Association Sustaining Members. In many cases these organisations also provide manpower and administrative assistance to the Association at their own cost.

Causeway Graphical Systems Ltd  
The Mallings, Castlegate,  
MALTON, North Yorks YO17 7DP  
Tel: 01653-686760  
Fax: 01653-697719  
Email: causeway@csi.com  
Web: www.causeway.co.uk

Compass Ltd  
Compass House  
60 Priestley Road  
GUILDFORD, Surrey GU2 5YU  
Tel: 01483-514500

Dyadic Systems Ltd  
Riverside View, Basing Road,  
Old Basing, BASINGSTOKE,  
Hants, RG24 0AL  
Tel: 01256-811125  
Fax: 01256-811130  
Email: sales@dyadic.com  
Web: www.dyadic.com

HMW Trading Systems Ltd  
Hamilton House,  
1 Temple Avenue,  
LONDON, EC4Y 0HA  
Tel: 0171-353 8900  
Fax: 0171-353 3325  
Email: 100020 2632@compuserve.com

Insight Systems Aps  
Nordre Strandvej 119C  
DK-3150 Hellebæk  
Denmark  
Tel: +45 49 76 20 20  
Fax: +45 49 76 20 30  
Email: info@insight.dk  
Web: www.insight.dk

APL2000  
Rapid Application Development  
6610 Rookledge Drive  
Suite 502  
Bethesda MD 20817  
USA  
Email: sales@apl2000.com  
Web: www.apl2000.com

Sollon Associates Ltd  
Groot Blankenberg 53  
1082 AC Amsterdam  
Netherlands  
Tel: +31 20 646 4475  
Fax: +31 20 644 1206  
Email: sales@sollon.com

Dutch APL Association  
Postbus 1341  
3430BH Nieuwegein  
Netherlands  
Tel: +31 347 342 337