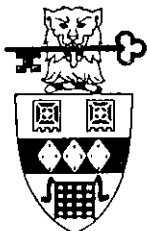


# VECTOR

## CONFERENCE SPECIAL

- APL86 – latest plans & abstracts
- Reviews – Computer Graphics & Expert Systems.
- Financial planning – doing famously with APL.
- APL2 – the migration path.
- Competitions, trivia, news & reviews.



*The Journal of the  
British APL Association*

A Specialist Group of the British Computer Society

Vol.2 No.4 April 1986

## Contributions

All contributions to VECTOR should be sent to the Editor at the address given on the inside back cover. Letters and articles are welcomed 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 submitted in duplicate and accompanied by as much visual material as possible, including a photograph of the author. Unless otherwise specified each item will be considered for publication as a personal statement by its author, who accepts legal responsibility that its publication is not restricted by copyright. Authors are requested wherever possible to supply copy in machine-readable form ideally text files on a 5¼" IBM-PC compatible diskette. For other standards, please contact the Editor beforehand. Program listings should indicate the computer system on which they have been run. APL symbols should be displayed on a separate line and not embedded in narrative. Except where indicated, items published in VECTOR may be freely reprinted with appropriate acknowledgement.

## Membership Rates 1986-87

Category	Fee p.a.		VECTOR	Passes
	£	\$	copies	
Nonvoting student membership	5		1	1
UK Private membership	10		1	1
Overseas private membership	18	27	1	1
Supplement for airmail (not needed for Europe)	8	12		
Corporate membership (UK)	85		10	5
Corporate membership (Overseas)	140	210		
Sustaining membership	360		neg	5

The membership year runs from 1st May to 30th April. Applications for membership should be made on the form at the end of the journal. Passes are required for entry to some Association events and for voting at Annual General Meetings. Applications for student membership will be accepted on a recommendation from a course supervisor. Overseas membership rates cover VECTOR surface postage and must be paid in £UK.

Corporate membership is offered to organisations where APL is in professional use. Corporate members receive multiple copies of VECTOR and are offered group attendance of Association meetings. Partaking individuals need not be identified but a contact person should be nominated 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 the editorial section of the journal and opportunities to inform APL users of their products via seminars and articles.

## Advertising

Advertisements in VECTOR should be submitted in typeset camera-ready A5 portrait format with a 20 mm blank border. Illustrations should be black-and-white photographs or line drawings. Rates are £250 per page. A6 and A7 sizes are offered subject to layout constraints.

Deadlines for advertisement bookings and receipt of camera-ready copy are given beneath the Quick-Reference Diary.

Advertisements should be booked with and sent to Steve Lyus, whose address is given beneath the Index of Advertisers.

## CONTENTS

		Page
GUEST EDITORIAL: APL86 – Be there	Dick Bowman	3
APL NEWS		
General Correspondence	Curtis Jones	5
Quick-reference diary	David Preedy	6
British APL Association news	Bowman & Camacho	7
News from sustaining members	Steve Lyus	11
International APL news	David Preedy	15
British Computer Society news	Phil Goacher	19
APL Product Guide	Steve Lyus	31
Software review – MicroSpan	Anne Wilson & Steve Phillips	41
APL86 PREVIEW		43
RECENT MEETINGS		67
Computer Graphics '85	Hollamby & Williamson	69
Expert Systems Conference	Peter Davies	77
GENERAL ARTICLES		79
Case Study: APL as a System		
Development Language	Christine Brewster	81
Imperial Group 5-year planning	Steve Lyus	93
TECHNICAL SECTION		101
Technical Editorial	Jonathan Barman & David Ziemann	102
Technical Correspondence	Henriod, Sullivan, Vesperoni	103
Competitions: Range union result	Jonathan Barman	105
"Watch your step" – New competition	David Ziemann	109
Surely there must be a better way	David Ziemann	111
Fuzzy look-up	Simon Barker	111
Phonetic searching	David Ziemann	112
APL Trivia: Life, the Universe ...	David Ziemann	115
Remarks on "Towards a better APL"	Claude Henriod	115
Migrating VS APL to APL2 under MVS/TSO	Mark Lenihan	123
Running VS APL under TSO	David Piper	127
Index to Advertisers		131
Membership Application Form		132

## BACK NUMBERS OF VECTOR

Back numbers of VECTOR are available from the BCS. If you don't have them all, now is the time to complete your collection. Apart from the technical contents, every issue includes book and product reviews, letters, news and a competition.. Send in your order before they run out. These will one day be unobtainable collectors' items, like the early issues of Quote Quad.

The prices inclusive of postage and packing are as follows:

	UK	Prices in Pounds Sterling	
		Surface (inc. Europe)	Airmail (outside Europe)
Single issues	3	3.75	5.75
Volume 1	10	14.00	22.00
Volume 2	10	14.00	22.00

Please send sterling cheques or money orders (payable to The British APL Association) to the Treasurer:

Mel Chapman, 12 Garden Street, Stafford ST17 4BT.

Don't forget to include your name and address and to be clear which VECTORS you want.

## Guest Editorial: APL86 – Be there

*by Dick Bowman*

As I write this (on a particularly sour January day in London) I realise that the nucleus of the APL86 committee has been working on the conference for about two years already; we're now only about five months away from the moment of truth. In any effort like this the questions to be asked are has it been worth it for us, and will it be worth it for you when you come? Of course it has, and of course it will be. What else would you expect?

Timing's an interesting thing – by the time you read this we're going to be a great deal further advanced in our detail planning than we are now but already those of us who've been looking at the many papers contributed have read much fascinating material that will only reach the delegates in July. If there's one regret we have it's that even after we've extended the conference to run the whole week from Monday to Friday with an additional optional pre-conference Education Day we've still had to leave out so much that was useful, fascinating and unique. We hope that the authors will allow us, and the VECTOR group can find room, to print many of these overflow papers in the coming months.

As one of our predecessors told us, the programme's 25 percent of a conference, the exhibition's another 25 percent, and the atmosphere makes up the rest. We know about the programme, and we know that we've been so heavily booked by exhibitors that we've had to expand from two exhibition halls to three. You can read some of the details later on, and you should have received your delegate pack some weeks ago. The other fifty percent is coming from you.

The committee has set up an environment – we have an excellent venue in the heart of Manchester – whatever you want to do at a conference we can almost certainly offer the facilities. Tell us your plans before you arrive and we'll do everything we can to make sure you meet the people who'll transform them into reality. Some of the preceding APL conferences have built up an imposing precedent – we met someone only loosely connected with the APL world – when we told him that we were organising APL86 his comment was “Oh yes, I went to APL79, it was the best computer conference I've ever been to (and the only APL conference)”.

We believe that APL86 is going to be like that; book early though – Manchester's a big place but even there we might run out of space.

## APL 86 EDUCATION OPEN DAY

## PROGRAMME

Except during the opening and closing plenary sessions there will be a large display of micro-computer implementations of APL available for hands-on trials with knowledgeable attendants to give assistance and demonstrations and a wide range of books to browse and literature to carry away. Thus, any delegate feeling the formal session is not his 'cup-of-tea' can instead take a leisurely look at the scope of the language.

- 9.0-9.30           Registration
- 9.30-10.00       Dr Kenneth E. Iverson - the man who started it all by inventing APL - will give the opening address "Executable Analytic Notation in Teaching Programming and Mathematics"
- 10.00-10.30      Professor Alan Hawkes, University College of Swansea "APL in action - examples of its use in mathematics and statistics"
- 10.30-11.00      Norman Thomson, IBM UK Labs Ltd  
"APL makes light of Tables"
- 11.00                               COFFEE
- 11.15-11.45      Ken Johnson, A.I.Applications Institute, Edinburgh  
"You Can't Get There From Here"
- 11.45-12.15      Anthony Camacho, Secretary British APL Association  
"Towards a better BASIC - the loop as an impediment to thought and what to do about it".
- 12.15                               Question and answer session until . . .
- 12.30                               LUNCH
- 1.30-2.15        Professor Robert Spence, Imperial College & Dr Frank Evans, South Bank Polytechnic  
"Elegant and Effective - Information Theorists and Engineers should use a common language"
- 2.15-2.45        Romilly Cocking, Cocking & Drury Ltd  
"Approaches to Teaching APL"
- 2.45-3.15        Bernadette McAlinden, TESCO Stores  
"How a 3-Day Course led to DRACULA"
- 3.15                               TEA
- 3.30-4.15        Roger Knott, University of Technology, Loughborough  
Roy Atherton, Head of IT, Bulmershe College H.E., Reading  
"Is this a private fight or can we all join in?"  
Discussion and question and answer session on LOGO/COMAL
- 4.15-5.00        Raymond Polivka, I.B.M. Education, New York  
"A Teacher Talks about Teaching APL"

## General Correspondence

*The VECTOR working group welcomes correspondence on any topic affecting the APL community. All such letters should be addressed to the Editor and should indicate whether they are intended for the general or the technical section. Letters containing APL code will normally appear in the Technical Section of VECTOR, and authors are asked to observe the requirements on the inclusion of APL code stated on the inside cover. The Editor reserves the right to edit any letter unless the writer states that it is to be published in full or not at all.*

From Curtis Jones

9th February 1986

Sir: The October VECTOR arrived by surface mail yesterday. My big disappointment was not seeing "a better way" to carry depreciation forward in response to the July STMBABW. I couldn't find one, myself!

I took a crack at the prize competition even though the problem came after the deadline. I'll send it anyway so you know people are reading!

Thanks to all who have contributed to putting out VECTOR.

Yours,

Curtis A Jones,  
apt. 1, 210 South 12th Street,  
San Jose,  
California.

*Ed: Apologies for the late arrival of your copy. Regular readers will have observed the long-standing tradition that VECTOR is distributed on or about the 75th day of the relevant month! We obviously cannot control the performance of the postal authorities. To their credit they delivered your reply within a week even though it was mis-directed via Jakarta!*

*We did have specific problems with the distribution of VECTOR 2, No. 2 as our normal mailing facilities were withdrawn at short notice and had to be replaced by a specially enrolled sub-committee of "volunteers". You might also be interested in our new arrangements to accept subscriptions in dollars (see British APL Association News).*

*Finally I would like to second your thanks to all those who have contributed to VECTOR over the last couple of years. By my estimation we have had 86 contributors since VECTOR was established and many of those have been new names to most of us. Please continue sending in your contributions.*

## Quick-Reference Diary

*compiled by David Preedy*

<b>Date</b>	<b>Venue</b>	<b>Event</b>
<b>1986</b>		
23 May	London	British APL Association meeting Ken Iverson – Special Guest Speaker
1-3 July	Wembley	Knowledge Based Systems '86 Organized by Online Conferences
7-11 July	Manchester	APL86 – APL in Action
19 September	London	British APL Association meeting
17 October	London	British APL Association meeting
21 November	London	British APL Association meeting
<b>1987</b>		
16 January	London	British APL Association meeting

### Dates for future issues of VECTOR

	<b>Vol. 3 No. 1</b>	<b>Vol. 3 No. 2</b>	<b>Vol.3 No.3</b>	<b>Vol.3 No.4</b>
Copy date	18 Apr 86	25 Jul 86	17 Oct 86	30 Jan 87
Ad. booking	9 May 86	22 Aug 86	14 Nov 86	22 Feb 87
Ad. copy	23 May 86	29 Aug 86	21 Nov 86	27 Feb 87
Distribution	July	October	January	April



## British APL Association News

### Notice of Annual General Meeting and Elections to the Committee

The British APL Association AGM will be held on May 23rd 1986 at the Royal Overseas League, Park Place, (off St. James's Street), London SW1, starting at 2pm. As in previous years the AGM will be followed by one of the Association's regular technical meetings.

All the posts on the committee are due to be filled by election at the AGM. The posts are:

- Chairman (who must be a member of the British Computer Society)
- Treasurer
- Secretary
- Activities Officer (who organises meetings and special events)
- Journals Officer (who is chief editor and organiser of VECTOR)
- Publicity Officer (who arranges promotion and coverage of the Association's activities)
- Technical Officer (who organises technical vetting and reviews of products, articles and papers)
- Education Officer (responsible for promoting education in APL and APL in education)

Each officer is encouraged to form a working party of Association members to help with the job and to provide a deputy to report at any committee meeting that circumstances make it impossible to attend in person. All officers are asked to help with the Association's work in other ways.

Every candidate for election to the committee must be proposed and seconded by Association members and is expected to provide a brief statement outlining their suitability and their intentions should they be elected. Every candidate is required to be a paid-up member and to recognise that it is a duty of a committee member to attend every committee meeting and to perform the appropriate tasks.

Advance nominations should be sent to the Secretary, 2 Blenheim Road, St Albans, Herts., AL1 4NR. None of the above is intended to preclude last-minute proposal and seconding (at the AGM) of an individual in good standing for any office.

## Committee Meeting Summary

*by Dick Bowman and Anthony Camacho*

The Committee writes a summary of the year's committee meetings for the last VECTOR in each year; here is a table of attendance at meetings:

		28 May	12 July	13 Sep	15 Nov	17 Jan
		85	85	85	85	85
Chairman:	R J Bowman	x	P	P	P	P
Secretary:	A J Camacho	x	P	P	P	P
Treasurer:	M J Chapman	x	P	P	P	P
Activities:	S Wilkinson	x	P	P	P	P
Journal:	D Preedy	x	A	P	P	A
Promotion:	R J Cocking	x	P	A	P	A
Education:	R Gray	x	P	A	P	P
Technical:	D Ziemann	x	P	P	A	P

P=Present

A=Apologies (28 May, pre-AGM, last year's committee)

One notable factor of the year is that we end it with the same committee as we began it with. Once again, committee meetings have normally occupied a complete morning; these summaries are a very brief account and do not reflect the considerable amount of work done for the Association outside committee meetings and indeed by many people who are not mentioned above.

### 28 May 1985

Aside from some preparation for the afternoon's AGM there was a lot of discussion about information gained from APL85 (PSG/RJB/MC/DG/DZ had all attended). APL86 is being organised by a committee formally quite separate from the BAA committee proper – although it shares some of the same faces and reports back (via RJB) to all BAA committee meetings.

### 12 July 1985

There was a brief handover meeting on 12 June between old and new committees. VECTOR costing and production were discussed at length (VECTOR was costing us more than our income). Time was also spent on discussing ways of arranging steady streams of VECTOR articles and topics for technical meetings. DG was keen to organise an "APL in Education" day. As a promotional exercise a "buy 3 – get 1 free" membership offer was agreed.

**13 September 1985**

'APL Publications' announced that it would cease trading – the committee considered how to provide a source of supply of APL books for members. Alternative quotations had been received for VECTOR typesetting and printing which were discussed; a changeover plan was agreed which appeared to reduce costs significantly and which would not disrupt the quite tight scheduling implicit in quarterly publishing. Entries in the 'free APL computer' offer were discussed and the lucky recipient chosen.

**15 November 1985**

There was a brief brainstorming session about how to increase membership and we listed the ideas for consideration. The handover of the Association's book stock (collected by AJC from APL Publications) to Mine of Information was agreed and that MoI could have the same advertising space in VECTOR as APL Publications had previously had. These arrangements will be reviewed after a trial period. Budget proposals for 1986/7 had been collected and possible membership rates were discussed.

**17 January 1986**

Promotional ideas had been expanded and it was agreed that we would begin by taking the following action:

- a) We would advertise the BAA outside the APL community subject to RJC identifying a medium we could afford.
- b) VECTOR will exchange mutual advertising with counterparts in other countries.
- c) We will now accept subscriptions in US dollars.

As these actions get under way we anticipate putting additional ideas into operation. VECTOR production costs had improved as expected with the switch of printer for the previous issue – the current issue was with new typesetters. Handover of 'APL Publications' had been completed.

**Membership Cards**

We will not be issuing membership cards this year (a fine time to tell you, some might think); however membership cards will be issued for the 1986/7 year. Membership cards fulfil several functions, confirmation of subscription receipt (note that we are requesting subscription payment at the beginning of the year now, and will not be sending any of next year's VECTORS until we get it), a record of your membership number and as a pass to any British APL Association events which bear a fee or are otherwise restricted to members. In the interim, please note that your membership number is shown on the address label of all BAA mailings.

**Dollar subscription rates**

With our rapidly growing overseas membership and in response to observations from existing U.S. members, the Association treasurer has decided to open a U.S. dollar bank account. This means that we are pleased to accept overseas membership subscriptions in dollars or in sterling. The following table gives the relevant rates for the coming year (1986/87) and it is hoped that our many U.S. members will take advantage of this service.

**Overseas subscription rates 1986-87**

Private membership	\$ 27 (air-mail supplement \$12)
Corporate membership	\$210
Library subscription	\$125

## News from Sustaining Members

*Compiled by Steve Lyus*

"Sustaining Membership" of the British APL Association is available to any company trading in APL products or services. It provides a tangible way for such companies to express their commitment to APL and to promote increased interest and activity in APL and in the Association.

As well as receiving public acknowledgement for their sponsorship of the Association, Sustaining Members receive bulk copies of VECTOR for distribution amongst their customers and are offered the opportunity to submit news material for this section of the Journal. They are also able to submit announcements for Association meetings (subject to the approval of the Activities Officer) and are invited to inform APL users of their products via seminars and articles. I am always surprised how much time I have to spend persuading them to contribute!

The Committee of the British APL Association would like to acknowledge the generous support of the following organisations that have become Association Sustaining Members.

APL People	17 Barton Street, Bath, Avon. Tel. 0225-62601
APL Software Technology(UK) Ltd.	14 Rosewood Avenue, Alveston, Bristol, BS12 2PP Tel. 0454-415737
Cocking & Drury Ltd.	16 Berkeley Street, London, W1X 5AE Tel. 01-493 6172
Dyadic Systems Ltd.	Park House, The High Street, Alton, Hampshire. Tel. 0420-87024
Inner Product Ltd.	Eagle House, 73 Clapham Common South Side London, SW4 9DG. Tel. 01-673 3354
MetaTechnics Systems Ltd.	Unit 216, 62 Tritton Road, London, SE21 8DE Tel. 01-670 7959
MicroAPL Ltd.	Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London, SW8 5BP. Tel. 01-622 0395
I.P. Sharp Associates	10 Dean Farrar Street, London, SW1. Tel. 01-222 7033

### **APL People Limited**

With H Walton Technical Systems well established, APL People's new project for 1986 is to start up 'APL Tran-Plan' – a company through which to develop their consultancy skills in the field of transportation planning. They already have some exciting projects in the pipe-line for APL Tran-Plan, such as the planning of car-parks in Chicago and of railways all over the world – North America, Pakistan and even Botswana. Having been within a hair's breadth of winning the contract to design Bangkok harbour, they at least have the consolation that they're not all fighting over who goes out there!

Meanwhile, they are continuing to build on their solid consultancy base, taking on more permanent staff and having more freelancers joining the ranks to cope with the increasing flow of work in 1986. They continue to offer their placement and advisory services to those contemplating a change of job or simply wanting some unbiased but informed opinions about career options in 1986.

### **Dyadic Systems Limited**

Dyadic Systems have just announced the implementation of Dyalog APL for the new IBM 6150/51 computers. The 6150/51 range (known as the IBM PC RT in the United States) is based upon a true 32 bit Rise (reduced instruction set) processor rated at 2 MIPS, and runs AIX, a Unix System V operating system. The machines, which will be generally available in May, come in either desktop or floor-standing units, and support up to 8 users. A complete system including Dyalog APL starts from around £15,000.

Dyalog APL for the IBM 6150/51 places no practical restrictions on APL object size or on workspace size. Performance is meant to be impressive, even on the pre-production models which Dyadic has tested.

Dyalog APL supports the full APL character set on the system console, which may be the current IBM 5151 PC display, the 5154 PC enhanced colour display, or one of the 3 new all-points addressable graphics displays. Additional users are supported by connecting standard ASCII terminals (or other PCs) to RS232 ports, and Dyadic will configure systems using the new Lynwood j300 and j500 APL/graphics terminals.

Dyadic will provide interfaces between Dyalog APL and IBM supported software packages for the system. These include the IBM SQL/6150 Data Base system and the IBM 6150/51 Professional Graphics Series. Interfaces for other third party software such as RS/1, UNIRAS, Applix, and the IMSL Fortran libraries are also planned.

### MetaTechnics Systems Limited

MetaTechnics Systems announce new versions of MetaScreen and MetaScreen2. The new versions are fully compatible with release 5 of STSC's APL\*PLUS/PC. They provide greater integration with the APL keyboard and contain several new features.

MetaTechnics have released their IEEE-488 (G.P.I.B) interface package. This is a set of utility functions, many written in assembler, which enable communication from an APL application program to laboratory instruments, process control equipment and other peripherals such as plotters. The package runs with the IEEE-488 interface card supplied by Tecmar but can be adapted to work with other interface cards.

MetaTechnics Systems Ltd can supply the NEC uPD70108 microprocessor chip at £15. This chip is a plug-in replacement for the 8088 processor fitted to the IBM PC (and compatibles) and can increase the speed by between 15% and 35% when running APL\*PLUS/PC.

### MicroAPL Limited

MicroAPL's new marketing department is making plans to expose what it calls the best kept secret in the computer world – APL. Gerard Paul-Clark, their marketing manager says "Now that APL is available on high-volume 68000-based machines (i.e. QL, Atari, Amiga, Mac.), we feel it is our duty to take advantage of this new mass audience. APL is a development tool that once seen is immediately addictive but the barriers to newcomers are the weird symbols and the equally weird APL community."

MicroAPL is planning to get this mass audience to try APL with the keyword version of APL.68000, which they are calling Liberty. Currently in its final 'product engineering' stage, it will be launched in the near future amidst a fanfare of advertising in the popular micro magazines. Prices will start from £99.95 including VAT for either Liberty or the symbolic version.

On the hardware front, MicroAPL announced the Aurora supermicro back in December. This is the successor to the Spectrum and is a modular system that MicroAPL claims is "designed by experts to be designed by you". The Aurora starts off with 1-16 users for the first processor board and as boards are added, so can more users be accommodated. The 68020 processor used is so powerful that the Aurora is claimed to perform at least 4 times faster than the straight 68000. David Eastwood, MicroAPL's Systems Division director, comments: "The Aurora is such a solid system that we are coming down off our APL perch and aiming to include the general market, there being many other languages available (FORTRAN, BASIC, Pascal, C)."

For the time being, MicroAPL will continue to sell to its prestigious customer base but Eastwood says "selling through what have become known as value added resellers is not beyond the realms of possibility".

# In every industry, there has to be a leader

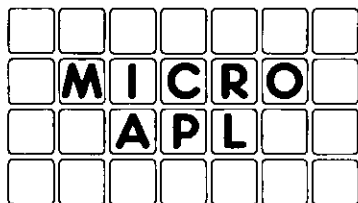
**Personal APL** – Our version of APL for the **Sinclair QL** brings full-powered APL at the lowest possible price.

**PC-APL** – MicroAPL supplies STSC's **APL\*Plus/PC** with full backup, support, and ancillary software.

**Departmental APL** – For the ultimate in **performance and power**, our multi-user APL supermicros can handle even the largest applications.

**Company-wide APL** – With unrivalled experience in networking and communications, MicroAPL can offer a **comprehensive APL facility** linking PCs, supermicros and mainframes.

**Consultancy in APL** – MicroAPL's commitment to quality doesn't stop at our hardware products. Our uniquely experienced team of APL consultants will undertake any APL software project, from one day's help with a specific problem, to major team developments.



**MicroAPL Limited**

Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London SW8 5BP.  
Telephone: 01-622 0395



## International APL news

*Compiled by David Preedy*

### AFCET

We have received a couple of brochures from the APL working group of AFCET – Association Francaise pour la Cybernetique Economique et Technique – who are organising various working groups and seminars on APL. Following a successful seminar at Cap d'Agde in May 1985, they are planning a similar event in June this year. If my French translation serves me well, the format is planned to include a 2-day APL course directed toward the APL2 families (IBM, STSC, Dyadic, ...), hardware and software demonstrations, and the conference proper. Full details can be obtained from:

AFCET, 156 Boulevard Pereire, 75017 Paris.

We also received the programme for a 2-day seminar they organised last December. Following an initial introductory session, the flavour moved strongly towards the areas of graphics and artificial intelligence with papers on such topics as:

APLIAS – a language for manipulating images constructed in APL

SEXPOL – prototyping expert systems in APL

APL for development of Artificial Intelligence utilities

The power of expression of LISP and APL

APL, Graphics and Artificial Intelligence

APL/Prolog comparison for writing a medical expert system

The material sounds very interesting and we would welcome the opportunity to print a review of the conference by anybody who attended it.

AFCET also advertised the availability of proceedings and video-tapes of some of their previous events on topics such as APL & Databases, Graphic techniques & APL, APL & microcomputers, APL and Infocentres, and Programmer productivity in APL.

### Danish APL Group

We have also received the following details from the Danish APL Group.

**Name:**

The APL Group  
Danish Data Processing Association  
14, Kronprinsensgade  
DK-1114 Copenhagen K  
Denmark

Phone: +45 - 1 - 11 15 60

**Contact:**

Kim S. Andreasen  
Beta-plan ApS  
Stengade 75  
DK-3000 Helsingør (Elsinore)  
Denmark

Phone: +45 - 2 - 21 48 48

**Mailbox:**

IPSA mailbox address: LARS

**General:**

The group was established in 1978 as a rather informal special interest group within the Danish Data Processing Association. Over the years, this group has arranged a number of highly successful events, including APL conferences with up to 200 delegates from throughout Scandinavia, well-attended afternoon seminars (called 'On-your-way-home'-meetings), an APL workshop, and APL exhibitions.

The events are most often held in Copenhagen and its surroundings, but we consider meetings in the provinces as well.

Our group benefits from having available the entire mailing list from the Danish Data Processing Association (with approx. 1600 members), as well as having access to the Inter-Scandinavian bi-monthly magazine *Data-nytt* (*Data-news*) published by the cooperating Scandinavian Data Processing Associations.

**Plans:**

Right now, the APL Group is planning at least 3 'On-your-way-home'-meetings for the season 85/86. As it has been the case earlier, we will try to make these meetings attractive to APL'ers as well as non-APL'ers, by presenting applications for which APL has shown to be a useful implementation language. In the announcements we will emphasize the applications, and not the APL language on its own.

The major event arranged by the APL group this season will be an exhibition, covering both APL-related hardware, software and services, and communications. The exhibition will be in Copenhagen, in late spring 1986.

At the same time, the APL group works on a book project, which we hope will result in a series of 5 valuable APL-handbooks (in Danish language). These 5 volumes will cover a rather broad spectrum of problems that APL programmers meet in their daily work. They will include an APL tutorial (2 volumes), an APL language reference, an APL Systems Design Handbook, and a book with useful programming tips and techniques. As we have been able to attract a number of very competent authors, we hope that these books will be used by APL programmers and teachers throughout Scandinavia.



## **H.M.W. PROGRAMMING CONSULTANTS LTD**

**Why not discover more about**

- ★ Consultancy/Support Service**
- ★ APL on the IBM PC**
- ★ VM/CMS Packages**

**Ring Ken Jackson at**

## **H.M.W. PROGRAMMING CONSULTANTS LTD**

142 FELTHAM HILL ROAD,  
ASHFORD, MIDDLESEX TW15 1HN  
Telephone: Ashford 41232

## British Computer Society news

*compiled by Philip Goacher*

As well as its geographical branches, the British Computer Society has a large number of special interest groups, which form the backbone of the Society's technical expertise over the whole computing field. In response to requests from several VECTOR readers for more information about these groups, our B.C.S. news section this issue gives details of the current list of groups.

The Specialist Groups Board, which meets four times a year, is the central co-ordinating body on which all groups are represented.

In the brief details which follow, affiliated groups are marked with an asterisk. Any enquiries should be addressed direct to the individual(s) named for each group. In case of difficulty, members are advised to contact the Liaison Executive (Specialist Groups) at B.C.S. Headquarters.

### **Association for Computing Machinery – British Chapter \***

The ACM British Chapter is a meeting point for the 70,000 members of the ACM worldwide. It acts as a forum for the exchange of international technical ideas and has at present some 200 members. In addition it is closely affiliated with three student associate groups at the Thames Polytechnic, Imperial College and the University College of Swansea.

#### Activities:

Social get-togethers; meetings; professional development seminars.

#### Enquiries to:

Liaison Executive (Specialist Groups),  
BCS Headquarters  
Tel. 01 637-0471

### **Advanced Programming**

This group was founded in 1959 and provides a forum for the consideration of new developments in software and their application to commercial data processing. The group took a lead in the early 1970s in spreading awareness of database systems. In the last two years, the group has concentrated on new programming paradigms, including expert systems languages and their place in information processing.

The group supports two working parties, on database administration and data dictionary systems.

#### Enquiries to:

Prof. J J Florentin, Dept. of Computer Science,  
Birkbeck College, Malet Street, London, WC1E 7HX.

**Auditing by Computer**

To provide a major forum where computer audit security and control matters are discussed by data processors, internal and external auditors from the public and private sectors, and end-users with a view to contributing to the development of secure, well-controlled computer systems to serve the needs of management and contribute to the development of cost-effective methods of audit for such systems.

**Activities:**

six meetings with speakers; annual conference; 1-day joint meeting with the Institute of Internal Auditors.

**Enquiries to:**

Mr. W List,  
KMG Thomson McLintock & Co.,  
70 Finsbury Pavement,  
London, EC2A 1SY.

**British APL Association**

Formed in 1976, it aims to promote the acceptance and widespread usage of APL. It provides a forum where both the novice and expert user can meet to discuss programming and design techniques, business applications, and new software and hardware tools available from the software houses and manufacturers.

**Activities:**

meetings; annual conference (in conjunction with other national APL user groups); quarterly journal, *VECTOR*.

**Enquiries to:**

Anthony Camacho (Secretary),  
2 Blenheim Road,  
St. Albans,  
Herts., AL1 4NR.

**British Computer Association for the Blind**

To promote the exchange of knowledge and experience among blind and partially-sighted

data processing professionals.

**Activities:**

2 courses p.a. on data processing user groups.

**Enquiries to:**

Mr F Ellis,  
9 Almond Tree Road, Cheadle Hulme,  
Cheadle, Cheshire, SK8 6HW.

**British Pattern Recognition Association**

To promote knowledge and application of pattern recognition.

**Activities:**

colloquia; international conferences; technical visits.

**Enquiries to:**

Prof. M J B Duff, Dept. of Physics & Astronomy,  
University College, London,  
Gower Street, London, WC1E 6BT.

**Business Information Systems**

To provide investigation and discussion of business-related aspects of data processing, so that computer practitioners, academics and non data-processing personnel can improve working systems and relationships.

**Activities:**

monthly meetings; newsletter; working groups; open events; committee representation.

**Enquiries to:**

Ms E Somegyl, Butler Cox & Partners,  
12 Bloomsbury Square,  
London, WC1A 2CC.

**Cobol**

Formed to study and improve the Cobol language.

**Activities:**

working groups on selected topics; no regular meetings.

**Enquiries to:**

Mr A Sale, 15 Northampton Road,  
Bromham, Beds., MK43 8QB.

**Computer Aided Design/Computer Aided Engineering**

To provide awareness of developments in CAD and CAE/M, and to provide a bridge between practical computer specialists and those industries which need to know about advances in computing technology with potential for applications in the sector.

**Activities:**

meetings, seminars.

**Enquiries to:**

Mr W Freidl, W Freidl Associates,  
33 Earls Court Gardens,  
London, SW5 0TR.

**Computer Arts.**

To encourage the use of computers in the arts and to act as a forum for the exchange of information in this field.

**Activities:**

10 meetings p.a.; exhibitions; bulletin.

**Enquiries to:**

Mr R J Lansdown, System Simulation  
Ltd., Southbank House,  
Black Prince Road, London, SE1 7SJ.

**Computer Education \***

To spread knowledge of computers and help with problems, e.g. establishing courses, setting up equipment, contacting specialist lecturers, industrial and other users.

**Activities:**

annual meeting; other group meetings; bulletin; formation of branches to organise local meetings and activities.

**Enquiries to:**

Dr. H L W Jackson,  
Head of Faculty, Maths, Statistics &  
Computing,  
North Staffordshire Polytechnic,  
Computer Centre,  
Blackheath Lane,  
Stafford, ST18 0AD.

**Computers and Employment**

To spread awareness of work on the impact of computers on the nature and content of work.

**Activities:**

meetings; seminars.

**Enquiries to:**

Ms L Antill,  
Office Technology Unit,  
Polytechnic of the South Bank,  
Manor House,  
Clapham Common Northside,  
London, SW4 9RZ.

**Computers in Psychology \***

The Mathematics, Statistics and Computing section of the *British Psychological Society* represents the Society's interest in mathematics, statistics, modelling cybernetics, systems theory and expert systems.

**Activities:**

workshops; specialist symposia; co-operation with other national and international societies to organise conferences.

**Enquiries to:**

Dr. A Elithorn,  
Dept. of Psychological Medicine,  
The Royal Free Hospital,  
Pond Street,  
Hampstead,  
London, NW3 2QG.

**Computer Security**

To develop awareness of computer security, to contribute to the development of security guidelines, standards, etc., and to monitor the state of the art of enabling tools, techniques and methods for protecting computer-based systems

**Activities:**

publications; seminars; discussions.

**Enquiries to:**

Liaison Executive (Specialist Groups),  
BCS Headquarters  
Tel. 01 637-0471

**Database**

To provide a forum for discussion of database and teleprocessing products and user problems.

**Activities:**

regular meetings; working parties; conference; Northern sub-group.

**Enquiries to:**

Mr. S Holloway,  
Applied Data Research,  
Portmill House,  
Portmill Lane,  
Hitchin,  
Herts., SG5 1DJ.

**Data Communication**

To further the exchange of knowledge and experience in all areas related to data communication.

**Activities:**

monthly meetings.

**Enquiries to:**

Dr. P Radford,  
Logica Ltd.,  
64 Newman Street,  
London, W1A 4SE.

**Developing Countries**

Formed in 1976, the group seeks to stimulate interest in the problems faced by developing countries in applying computers effectively for the benefits of their societies and national economy.

**Activities:**

seminars; conferences; publications, including a regular newsletter.

**Enquiries to:**

Mr. J Sherif,  
71 St Gabriel's Road,  
London, NW2 6BU.



**Disabled (North and South)**

Formed in 1975 to secure the increasing involvement of severely disabled computer practitioners, to study and publicise the special needs of severely disabled computer practitioners and to draw the attention of other severely disabled people to the possibilities offered by computers, especially to those who have not yet left school.

Activities:  
conference; 1-day afternoon meetings.

Enquiries to:  
Mr. Frank Ellis,  
CBG (Computer Centre) Roseneath,  
Carwood Road,  
Branhall, Stockport,  
Cheshire, SK7 3LS. (North)

Mr. Geoff Busby,  
3 Dowches Gardens,  
Kelvedon, nr. Colchester,  
Essex, CO5 9JP. (South)

**Display**

To study, examine, discuss and disseminate information on graphics, displays and associated technology, from detailed hardware design/construction through logical modules and software interfaces to higher-level application areas.

Activities:  
meetings; seminars; workshops; visits to installations.

Enquiries to:  
Dr. R A Earnshaw,  
University Computing Service,  
Leeds University,  
Leeds, LS2 9JT.

**Distributed Systems**

To further the understanding of distributed

systems, provide a forum for the exchange of ideas between interested parties, contribute to the development of relevant standards and methods.

Activities:  
meetings; seminars; publications.

Enquiries to:  
Mr. M Imber,  
10 Verulam Road,  
Greenford,  
Middlesex, UB6 9RH.

**Electronic Publishing**

To provide a forum for discussion for all involved in electronic publishing, to exchange ideas, identify common problems and try to devise better solutions.

Activities:  
meetings; seminars.

Enquiries to:  
Mrs. H Brown,  
Electronic Publishing Unit,  
Computer Laboratory,  
University of Kent,  
Canterbury, CT2 7NF.

**Eurographics Association \***

To promote graphics activities in Europe.

Activities:  
annual European 5-day conference plus tutorials, newsletter, local events in UK, including 2-day conference at Easter.

Enquiries to:  
Dr. R A Earnshaw,  
University Computing Service,  
Leeds University,  
Leeds, LS2 9JT.

**Expert Systems**

To foster research, development and application of expert systems.

**Activities:**

newsletter; 2 principal annual conferences (technical and tutorial).

**Enquiries to:**

Mr. M Ashill,  
The Grange Cottage,  
off Southborough Road,  
Surbiton,  
Surrey, KT6 6JN.

**Formal Aspects of Computing Science**

To facilitate the interchange of theoretical results and research and to make available the results of formal studies in computing science.

**Activities:**

2 or 3 conferences/ workshops p.a.;  
newsletter; occasional afternoon/evening seminars.

**Enquiries to:**

Dr. D J Cooke,  
Dept. of Computer Studies,  
The University of Technology,  
Loughborough,  
Leics., LE11 3TU.

**Fortran**

To provide a forum for discussion of Fortran and its applications. To monitor and seek to influence the development of the language through interaction with the various standards bodies.

**Activities:**

regular meetings; seminars; conferences.

**Enquiries to:**

Mr. J D Wilson,  
Computer Laboratory,  
Leicester University,  
Leicester, LE1 7RH.

**Computer Interest Group of the Society of Genealogists \***

Formed to report on progress in the application of computers in genealogy both for amateur genealogists and professionals needing to process records of genealogical interest, and to provide a forum for discussion of relevant matters.

**Activities:**

quarterly newsletter; conferences;  
evening meetings.

**Enquiries to:**

David Hagwood,  
26 Cloister Road,  
Acton,  
London, W3 0DE.

**Human Computer Interaction**

To forge links between HCI research and its implementation in industry, commerce and government, to provide communication between users, designers and researchers of interactive systems, to liaise with other professional bodies to promote greater awareness of the importance of computer human features.

**Activities:**

meetings; workshops; conferences;  
publication.

**Enquiries to:**

Mr. P Johnson, Dept. of Computer  
Science & Statistics, Queen Mary  
College, Mile End Road,  
London, E1 4NS.

**Information Retrieval**

To provide a forum for discussion of operational and research aspects of computerised information systems.

**Activities:**

meetings; publications; register of expertise and experience; working groups; newsletter; liaison with other specialist and national organisations, End User Systems Group.

**Enquiries to:**

Dr. N Teskey, Dept. of Computing Science, Brighton Polytechnic, Moulsecomb, Brighton, BN2 4AY.

**Law**

To study the ways in which computers can assist lawyers and the problems which computers raise for the law.

**Activities:**

speaker meetings; occasional advice on legal matters.

**Enquiries to:**

Prof. G B F Niblett, Dept. of Computer Science, University College of Swansea, Singleton Park, Swansea, SA2 8PP.

**Lincoln Microprocessor Group \***

To inform and educate members and the general public in and around Lincoln on techniques, applications and social implications of micro-electronics and computation.

**Activities:**

lectures; visits; exhibitions and forums; provision of computers for practical work by members; sub-group giving aid to the handicapped.

**Enquiries to:**

Dr. P Griffiths St Georges Hospital, Lincoln, LN1 1EF.

**Local Government**

To promote the interests of the Society and the computer profession within local authorities and associated bodies.

**Activities:**

specialist presentations; normal group activities.

**Enquiries to:**

Mr. D Griffiths, Assistant Director, LAMSAC, Vincent House, London, SW1P 2NB.

**Mathematical Programming**

To promote the dissemination of knowledge and discussions in the area of mathematical programming.

**Activities:**

10 lectures p.a.; periodic specialist studies.

**Enquiries to:**

Dr. G Mitra,  
5 Virginia Drive,  
Virginia Water,  
Wentworth, Surrey, GU25 4RX.

**Medical (London)**

To provide a forum for discussion of all aspects of health-care computing (with the exception of primary health care and nursing) in the South. Founder member of European Federation for Medical Informatics and member of International Medical Informatics Association.

**Activities:**

annual conference; meetings; journal (in association with outside agency).

**Enquiries to:**

Mr. S W Sargent,  
RSS Implementation Manager,  
Shaftesbury House,  
402 High Street,  
West Bromwich, B70 9NR.

**Medical (Northern)**

To provide a forum for discussion of all aspects of health-care computing (with the exception of primary health care and nursing) in the North.

**Activities:**

annual conference; meetings; journal (in association with outside agency).

**Enquiries to:**

Ms. J Roberts,  
PAS/IPS Central Team,  
Room 616,  
Gateway House,  
Piccadilly South,  
Manchester, M60 7LF.

**Medical (Scotland)**

To provide a forum in Scotland for those interested in all aspects of medical computing.

**Activities:**

meetings; two 1-day conferences p.a.

**Enquiries to:**

Mr. R B Marshall,  
Greater Glasgow Health Board,  
222 Bath Street,  
Glasgow, G2 5JT.

**Microcomputer**

To provide a forum for discussion on the impact of the "micro revolution", on both conventional computing and society as a whole.

**Activities:**

meetings; seminars.

**Enquiries to:**

Mr. R Phillips,  
19 Blackthorne Road,  
Great Bookham,  
Surrey, KT23 4BN.

**Microform**

To promote awareness throughout the computer and business community as to the breadth and depth of COM (computer output to microfilm) as a systems tool.

**Activities:**

seminars; discussion sessions; application/user visits.

**Enquiries to:**

Mr. T Hendley,  
17 Mobbsbury Way,  
Stevenage, Herts., SG2 0HC.

**Natural Language Translation**

To provide and develop the use of machine translation.

**Activities:**

meetings; seminars.

**Enquiries to:**

Mr. I D K Kelly,  
Bellows,  
Burrowhill Green,  
Chobham,  
Woking, Surrey, GU23 8QP.

**Nursing**

To promote and support the use and development of computers in nursing in providing patient care, nursing education, management and research; to promote computer literacy among nurses.

**Activities:**

meetings; seminars; conferences.

**Enquiries to:**

Mrs. C Ashton,  
45 Listowel Road,  
Kings Heath,  
Birmingham, B14 6HH.

**Office Automation**

To disseminate information on word processing and office automation.

**Activities:**

monthly meetings (September to May); half- or 1-day seminar each year; 3 regional subgroups (St Helens, Croydon, Strathclyde); newsletter.

**Enquiries to:**

Mrs. B Levine, Flat 5,  
16 Canfield Gardens, London, NW6 3LA

**Payroll**

To provide a forum for the discussion of matters of interest relating to the payment of salaries, wages and pensions using computers; the legislation which affects the processes; and new developments in the area.

**Activities:**

meetings (about 6 p.a.)

**Enquiries to:**

Mr. C A Cowland,  
PE/PS/4, Greater London Council,  
The County Hall,  
London, SE1 7PB.

**PL/1**

To promote interest in, and discussion of, PL/1 and its applications.

**Activities:**

meetings; bulletin.

**Enquiries to:**

Mr. A P Smith,  
7 Meriden Court,  
Northlands Drive,  
Winchester, Hants., SO23 9TH.

**Primary Health Care**

To promote all activities and interests in this field for GPs, nurses, dentists, ancillary staff and all computer professionals in the primary health care field.

**Activities:**

newsletter; meetings; symposia; committee activities.

**Enquiries to:**

Dr. G Clayton,  
1 Trinity Street,  
Norwich, NR2 2BQ.

**Process Control**

To create a forum for discussion of developments affecting process control, software packages and techniques; to foster closer contacts between users, suppliers and universities; and to promote the interchange of ideas and techniques originating from the use of micros, minis and mainframes.

**Activities:**

conferences; symposia; courses.

**Enquiries to:**

Dr. P A Bennett,  
Byways, 2 Brigg Road,  
Broughton, Nr. Brigg,  
South Humberside, DN20 0TW.

**Robotics**

The group provides a major forum for the exchange and discussion of knowledge in all areas related to the use of computing in robotics. It is also designed to foster research and application of such technology with attention to its social consequences, and to promote professional practice through education and training.

**Activities:**

seminars; meetings.

**Enquiries to:**

Peter B Scott,  
The Centre for Robotics,  
Imperial College,  
Exhibition Road, London, SW7 2BX.

**Software Engineering Specialist Group**

To consider the total process of developing software from initial idea to final system through maintenance and enhancement to ultimate replacement. Methods, tools to support them, management processes to back them up, and theories to underpin them are all covered.

**Activities:**

day meetings (11am to 4pm) on a single theme once a quarter usually in Central London; some special events; working groups (currently on testing, with quality assurance and cost estimating being established).

**Enquiries to:**

Ann Petric (Secretary),  
Computing Laboratory,  
University of Newcastle,  
Newcastle-upon-Tyne, NE1 7RU.

Patrick Hall (Chairman),  
Cirrus Computers Ltd.,  
Waterside Gardens,  
Fareham, Hants., PO16 7AI.

**Study Group on Computers in Survey Analysis**

Formed in 1971 to provide a forum for exchange of information on the use of computers for the processing and analysis of survey data so as to reduce the duplication of programs. Publishes biennial software register which gives details of packages/programs for use in survey and statistical analysis and administration.

**Activities:**

conferences in June & November; prize for the year's best published paper on the use of computers in survey or statistical processing.

**Enquiries to:**

Terry Orchard,  
Cabinet Office,  
Central Statistical Office,  
Great George Street,  
London, SW1P 3AQ.

### Technology of Software Protection

Started in 1981 to act as a forum to update members on existing and new techniques for the protection of software by technical methods. The scope of the group is intended to cover all non-legal methods of protection. Recent meetings have covered disk protection, intelligent dongles, cryptography, software security and new inventions in hardware.

**Activities:**

evening meetings; 1-day conferences.

**Enquiries to:**

Mr. Len Jones,  
377 Nore Road,  
Portishead, Bristol, BS20 8EX.

### UK Simulation Council


To promote the advancement of simulation and allied computer technology. Full members are also members of the USA-based Society for Computer Simulation and receive the journal Simulation.

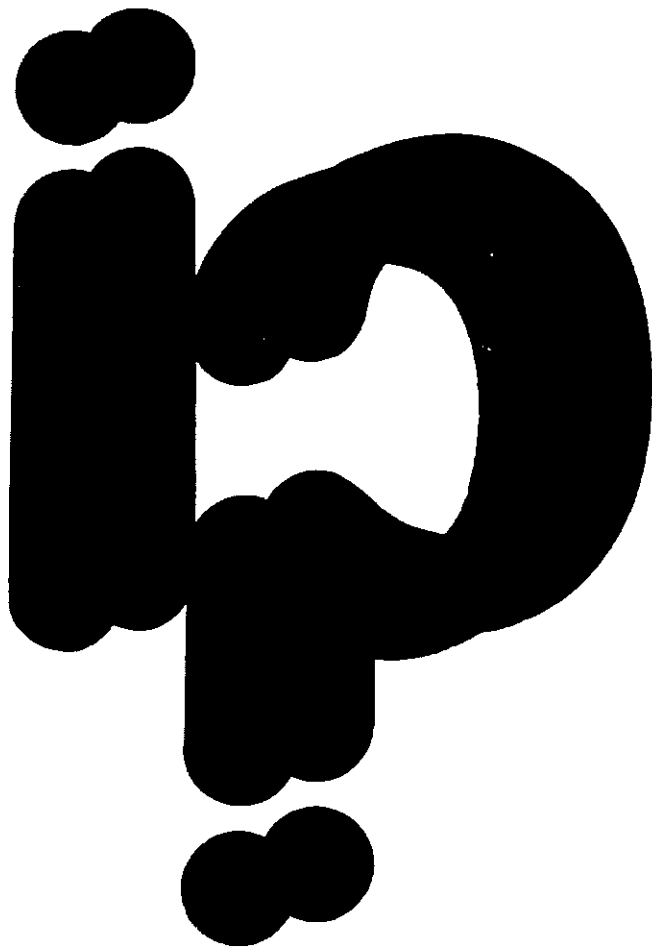
**Activities:**

three or four 1- or 2-day meetings p.a.; conferences triannually.

**Enquiries to:**

Dr. P Luker,  
Computing Laboratory,  
University of Bradford,  
Bradford, BD7 1DP.

	<p>APL People Ltd.</p>	<p>17 Barton Street, Bath, Avon, BA1 1HQ Tel: 0225 62601</p>
<p><b>APPOINTMENTS</b></p>		
<p>A placement service solely for the APL community. Use our extensive knowledge, contacts and experience in the APL world. We have vacancies NOW, both permanent and contract ..... and our confidential advice on career moves is always freely available.</p>		
<p><b>CONSULTANCY</b></p>		
<p>A consultancy service designed to solve your APL problems. Consultant's Toolkit and Supervision available.</p>		
<p>Current projects include:-</p> <ul style="list-style-type: none"> <li>* graphics</li> <li>* project management</li> <li>* databases</li> <li>* system design</li> <li>* system 'rescue'</li> <li>* documentation</li> <li>* VSAPL under VSPC, CMS, TSO</li> <li>APL*Plus/PC, Sharp APL</li> </ul>		
<p>To discuss your requirements: — Call Valerie Lusmore on 0225-62601 Registered employment agency licence SE6440</p>		



**THE SOLUTION TO  
YOUR APL QUESTIONS**  
*on IBM PC or Mainframe.*  
**INNER PRODUCT**

For further information, contact JAMES MANNING, Marketing Manager, Inner Product, Eagle House, 73 Clapham Common Southside, London SW4 9DG, or telephone 01-673-3354.



## APL Product Guide

*Compiled by Steve Lyus*

VECTOR's exclusive APL 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 do depend on the alacrity of suppliers to keep us informed about their products so that we can update the Guide for each issue of VECTOR. Any suppliers who find they are no longer included, have not bothered to reply to my two warnings. If they would like to be re-instated, would they please contact me to confirm their entry. Any suppliers who are not included in the Guide should contact me to get their free entry – see address below.

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. Where no UK distributor has yet been appointed, the vendor should indicate whether this is imminent or whether approaches for representation by existing companies are welcomed.

For convenience to readers, the product list has been divided into the following groups:

- ★ Complete APL Systems (Hardware & Software)
- ★ APL Timesharing Services
- ★ APL Interpreters
- ★ APL Visual Display Units
- ★ APL character set printers
- ★ APL-based packages
- ★ APL Consultancy
- ★ Other services
- ★ Vendor addresses

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. The Guide has now been entered on our Wordprocessor, so there might be some this time!

*Note:* 'poa' indicates 'price on application'

All contributions to the APL Product Guide should be sent to:

Steve Lyus  
Group Management Services  
Imperial Group plc  
East Street  
Bedminster  
Bristol BS99 7JR

**COMPLETE APL SYSTEMS**

COMPANY	PRODUCT	PRICES £	DETAILS
Analogic	The APL Machine	\$60,000+	AP500 array processor, 4 Mb data memory, 80 Mb disk drive.
APL Software Technology	Ericsson PC	2,000 -5,000	Authorised Ericsson dealer supplying complete APL systems based on Ericsson PC - fully IBM compatible including hardware add-on boards.
Cooking/Drury	MicroAPL SPECTRUM SAGE II SAGE IV	6,000 -35,000	Supplied as part of a turnkey system. See MicroAPL entry.
Gen. Software	Myriade	poa	TI computer + APL & APL operating system
Inner Product	IBM PC	2,000 -6,000	IBM PCs supplied for turnkey applications
M.B.T.	MBT Series 10 TORCH	poa poa	UNIX/68010 based multi-user APL system 68000/Z80 multiprocessor
MetaTechnics	—	3,000+	Details on application
MicroAPL	Aurora	23500	Multi-user APL computer using 68020 CPU. Std. configuration 2Mb RAM, 16 RS232 ports, 68 Mb hard disc, 720K diskette
	Sinclair QL with QL/APL SPECTRUM	300+ 11,000 -15,000	Fully expandable APL system with colour graphics. Expandable multi-user APL computer using Motorola 68000. Std. configuration 1 Mb RAM, 12/36 Mb disc, 12 ports.
	STRIDE 440	8,500	Multi-user APL computer, 1 Mb RAM, 12/18 Mb disc.

**APL TIMESHARING SERVICES**

COMPANY	PRODUCT	PRICES £	DETAILS
Boeing	Mainstream APL	poa	Enhanced IBM VS APL (CMS)
Mercia	APL*PLUS	poa	STSC's Mainframe Service - MAILBOX etc.
I.P. Sharp	SHARP APL	poa	International Network application systems and public databases.

**APL INTERPRETERS**

COMPANY	PRODUCT	PRICES £	DETAILS
APL Software Technology	APL*PLUS/PC	600 -695	See Cocking & Drury entry. Release 5 now available. Discounts available on multiple copies.
Cocking/Drury	APL*PLUS/PC Release 5 Release 4 update Release 3 update Run-time	695 150 110 poa	STSC's full-feature APL for IBM PC, PC/AT, Compaq, Olivetti, Wang, Apricot, Ericsson etc Extension upgrade from release 4 Extension upgrade from release 3 Closed version of APL*PLUS PC which prevents user exposure to APL.
	APL*PLUS UNX	poa	STSC's 2nd generation APL for IBM PC/AT, DEC, AT&T and other Unix computers.
Dyadic	Dyalog APL	1,000 -8,000	2nd gen. portable APL for UNIX systems, e.g. IBM PC/AT, SUN, MC68000, HP9000, NCR, VAX, Gould, Amdahl.
Gen. Software	APL*MYRIADE	poa	Runs on Texas Instruments T1990 range.
IBM UK Product Sales	IBM PC APL	poa	Event-handling & APs for full-screen I/O disks, diskettes, asynch. comms.
Inner Product	VIZ::APL	250 -350	8-bit Zilog Z-80 CP/M
	APL*PLUS/PC	600	See under Cocking & Drury
M.B.T.	Dyalog APL MBTAPL VIZ::APL	poa poa poa	See Dyadic Systems entry Enhanced Dyalog APL for MBT hardware. Customized for TORCH hardware
Mercia	APL*plus/PC  Release 5.0 upgrade to 5.0 APL*PLUS/UNIX	695  150 poa	STSC's full-feature APL for IBM PC, PC/AT Compaq, Olivetti, Wang, Apricot, Ericsson etc  Interpreter for UNIX systems: WICAT, CADMUS, CALLAN, FORTUNE 32:16, HP, 9000/500, OLIVETTI 3B2
MetaTechnics	APL*PLUS Release 4	695	Discount on quantity.
MicroAPL	APL.68000  QL/APL (keyword) QL/APL (APL chars) MAC/APL AMIGA/APL ATARI/APL	1,000+  87 87 257 326 131	Full implementation with component files, error trapping etc. for SPECTRUM, SAGE & other MC68000-based computers. Full keyword APL for QL with many extra features. VSAPL compatible APL for QL with many extra features.
Portable	PortAPL	\$195 \$275 \$2,995	IBM PC Software Mackintosh DEC VAX
I.P. Sharp	Sharp APL/PCX  Sharp APL/PC	2,575 1,000+ 325	For IBM XT/AT For IBM mainframes For IBM PC or PC/XT

## APL VISUAL DISPLAY UNITS

COMPANY	PRODUCT	PRICES £	DETAILS
Farnell	Tandberg TDV 2221	995	Ergonomic design APL terminal, 50-19200 baud, 15" anti-reflex screen, low profile keyboard
Gen. Software	Tandberg TDV 2271	1,195	Combined APL/ANSI ergonomic terminal as above. Second-hand
	Mellordata Elite 3045A	400	
M.B.T.	various		Contact MBT for details
MicroAPL	Insight VDT-1	795	Inexpensive APL VDU
	Insight GDT-1	1,450	With monochrome graphics
	Tektronix 4105	3,545	High resolution colour graphics supporting APL character set on MicroAPL hardware.
	Concept 201	1415	APL VDU with 8 page memory
	Concept 201G	1935	Graphics VDU
Shandell	AVT-APL+ GVT-APL+	1,200+	ANSI 3.64; DEC compatible; full overstrike chars; 4/8 pages; 2/3 comms. ports; 80/132 col/s; windowing; 12" screen; 46 PF keys; Tek 4013 graphics compatible.
	HDS 200	1,200+	As above plus 15" screen; viewports; smooth scroll; NVM storage; 55 PF keys; Tek 4014, VT640/DQ640 & Visual 500 graphics compatible.
Textronix	4114B	13,500+	19" D.V.S.T.: Graphics: 3120 x 4096 displayable; Intelligent: up to 800K memory; APL keyboard (option 4E)
	4125	21,550+	19" 2D colour graphics; Workstation (1280 x 1024); Intelligent: up to 800K memory; APL keyboard (mod AP)
	4128	26,822+	As 4125 plus 3D wireframe

## APL PRINTERS

COMPANY	PRODUCT	PRICES £	DETAILS
Datatrade	Datasouth DS180+	1,295	180 cps matrix printer with 4K buffer, 9 x 7 dot matrix and APL option. Letter quality; graphics capability, APL option (both available with IBM Twinex or Coax interface).
	Datasouth DS220	1,695	
Inner Product	Epson FX80	500	Soft char. set, 160 cps, 80 column
	Anadex 9620	1,150	200 cps., 132 col., tractor feed
	Siemens PT88	620	180 cps., 80 col., silent
	TGC Starwriter	1,180	40 cps., letter quality
M.B.T.	Facit 4565	poa	40 cps letter-quality
	Facit 4510/11/12	poa	Matrix printers
MetaTechnics	Quen-data	295	Low-cost APL Daisy-wheel printer
MicroAPL	Datasouth DS180+	1,295	See Datatrade entry
	Philips GP300	1,924	Matrix printer with letter & draft quality and APL.
	Qume Letterpro20	549	APL/ASCII Daisy-wheel printer

## APL PACKAGES

COMPANY	PRODUCT	PRICES £	DETAILS
APL ◊ 385	FSM 385	PC: 50	Screen development
	DRAW 385		Screen design
	DB 385 Mainframe:	125	Relational W.S.
	GEN 385		Utilities
APL Software Technology	POWERTOOLS/PC	295	Assembler written replacement function for commonly used CPU-consuming APL functions. Includes extended full-screen Forms Processor.

Boeing Beta-plan	TABAPL BETA-FONT	poa poa	Hierarchical Planning System Multiple font PC character generator. Dealers required for non-Scandinavian countries.	
Cocking/Drury	<i>Mainframe</i> STSC's SHAREFILE	poa	Component files, quad- functions & nested arrays for IBM VSAPL under VM/CMS & MVS/TSO	
	SHAREFILE AP	5,000	STSC's sharefile for APL2	
	EMMA	poa	STSC's database package.	
	CALL/AP	3,000	Non-APL program execution from VS-APL or APL2 under VM/CMS & MVS/TSO	
	AFM	10,000	High-performance shared files for VS-APL.	
	FORMAT	2,250	Enhanced report formatting.	
	WSAIDS	795	Workspace doc. & development aids.	
	<i>Microcomputer</i> STATGRAPHICS PC	995	Statistics & Graphics for IBM PC, AT, etc.	
	APL*PLUS PC Tools			
		VOL 1	325	Incl. 327x IRMA support, RAM disk, full screen data entry, menu input, report generation, games.
		VOL 2	125	Incl. file documentor, screen editor, exception handler.
	APL*PLUS PC Fin & Stat. Library	350	Financial & statistical routines	
	APL/DIF	145	Interface between APL*PLUS PC and DIF file supporting spreadsheets.	
	MetaScreen	245	See MetaTechnics entry.	
MetaPack	95	See MetaTechnics entry.		
POWERTOOLS	295	See APL Software Technology entry.		
E&S	PROTOPAK consisting of:		Packages for prototyping management information systems - PC & mainframe	
	RMS	Modules 250+	Relational databases.	
	AMS		Multi-dimensional arrays	
	RAMS		Combined RMS & AMS.	
	BMS		Dynamic financial modelling & forecasting	
	FMS		Full-screen handler for APL*PLUS/PC. (AP 124-based)	
	CMS		Communications package.	
	SOS		Scheduled ordering and stock control.	
	poa			
Gen. Software	PROPS	500+	Spreadsheet system for Product and/or Project Planning.	
H.M.W.	INPUT	poa	Matrix manipulation package for data entry & report generation	
	PRINTPAK	poa	Block printing for VM/CMS	
	VIEWPAK	poa	AP124 Protocol emulator for IBM/PC	
Holtech	CASH	3,500	Accounting package & hotel management system on MicroAPL SPECTRUM & SAGE CPUs.	
		-10,000		
Inner Product	Viewcom	150	Control Viewdata from APL	
	APL/dBASE II	150	Interface APL with dBASE II	
	APL/dBASE III	150	Interface APL with dBASE III	
	APL/LOTUS	150	Interface APL with Lotus	
	APL/WORDSTAR	150	Interface APL with Wordstar	
	APL/MULTIPLAN	150	Interface APL with spreadsheet	
	CEMAS	3,500	EEC monetary and agrimonetary analysis.	
M.B.T.	RHOMBUS	poa	Integrated Office System	
	HASLEMERE	poa	Hotel Accounting System	

Mercia	STATGRAPHICS	995	Integrated stat. graphic system for PCs.	
	APL*PLUS tools			
		VOL 1	325	IBM PC Utilities: IRMA3270 comms, full screen, RAM Disk report generator
		VOL 2	125	File documentation, screen editing. Exception handling
			275	Financial and Statistical analysis
	FINANCIAL AND STATISTICALLIB. INFO CENTRE	2,000	Full-screen entry, display & multi-dimensional analysis. Interfaces to other I.C. products. Runs under VM VSAPL on IBM mainframes.	
	EXECUCALC	-20,000	Mainframe Spreadsheet with VisiCalc and Lotus 1-2-3 functionality requires VSAPL under TSO or VM.	
EXECUPLLOT	4,000	Mainframe Graphics display system with VisiPlot functionality requires VSAPL under TSO or VM and GDDM.		
EXEC*U*STAT	3,200	Easy to use Statistics for management.		
MetaTechnics	MetaScreen	245	Full-screen handler for APL*PLUS/PC, based on VSAPL AP124	
	MetaPack	495	Comprehensive utilities package for APL*PLUS/PC. Includes MetaScreen, MetaWS, Browse, Toolbox, Numeric Editor	
	ADAPTA DLS	poa	Production & purchasing scheduling for process manufacturing.	
	ADAPTA MSP	poa	Job-shop loading & scheduling for multi-stage production.	
MicroAPL	MicroTASK	250	Product development aids	
	MicroFILE	250	File utilities and database	
	MicroPLOT	250	Graphics for HP plotters etc	
	MicroLINK	250	General device communications	
	MicroEDIT	250	Full screen APL editor	
	MicroFORM	250	Full screen forms design	
	MicroSPAN	250	Comprehensive APL tutor	
	MicroGRID	poa	Ethernet & other networking	
	APL CALC	400	APL spreadsheet system	
	MicroPLOT/PC	250	For APL*PLUS/PC product	
MicroSPAN/PC	250	For APL*PLUS/PC product		
Parallax	ExecuCalc	\$5,000	Mainframe-based electronic spreadsheet for VM/CMS & MVS/TSO with links to micro products.	
	ExecuPlot	\$5,000	Mainframe-based colour graphics with micro links.	
I.P. Sharp	ACT	poa	Actuarial system	
	APS	poa	Financial Modelling	
	BOXJENKINS	poa	Forecasting technique	
	CONSOL	poa	Financial Consolidation	
	COURSE	poa	APL Instruction	
	EASY	poa	Econometric Modelling	
	FASTNET	poa	Project Management	
	GLOBAL LIMITS	poa	Exposure management for banks	
	MABRA	poa	Record maintenance/reporting	
	MAGIC	poa	Time series analysis/reporting	
	MAGICSTORE	poa	N-dimensional database system	
	MAILBOX	poa	Electronic Mail	
	MICROCOM	poa	Mainframe to micro link	
	SAGA	poa	General graphics, most devices	
	SIFT	poa	Forecasting system	
	SNAP	poa	Project management	
SUPERPLOT	poa	Business graphics		
VIEWPOINT	poa	4GL - Info centre product		
XTABS	poa	Survey Analysis		

**APL CONSULTANCY**

(prices quoted are per day unless otherwise marked)

COMPANY	PRODUCT	PRICES £	DETAILS
APL People	Consultancy	poa	All levels, most APL systems
APL Software Technology	Consultancy	poa	Technical & business systems, micros, networking & communications a speciality.
Boeing	Consultancy	poa	
Camacho	Consultancy	poa	Specialising in programming & manual writing.
Cocking/Drury	Consultancy	120-150 140-200 185-300 275-400	Junior consultant Consultant Senior consultant Managing consultant
	Courses	375 330 595 poa poa	3 day Fundamentals 3 day Advanced 5 day System Design Introduction to APL2 APL2 in depth
Delphi	Consultancy	poa	Specialising in management reporting systems and APL on microcomputers.
Dyadic	Consultancy	poa	APL system design, consultancy, programming & training for Dyalog APL, VSAPL, APL*PLUS, IPSA APL etc.
E & S	Consultancy	150 -250	System prototyping: all types of information system.
FASTCODE	Consultancy	poa	Specialise in improving performance of APL applications on micros & mainframes.
Gen. Software	Consultancy	100+	
H.M.W.	Consultancy	100-250	System design consultancy, programming.
Inner Product	Consultancy & Training	200	On-site micro-mainframe APL, PC/DOS & Assembler
M.B.T.	Consultancy & Courses	poa	
Mercia	Consultancy & Training	poa	APL*PLUS & VSAPL consultancy. Onsite & Offsite APL courses.
MetaTechnics	Consultancy	poa	Management Information & Production. Management Systems & Process Control.
	MetaSys	poa	Prototyping APL-Hybrid Custom Programming Service.
MicroAPL	Consultancy	poa	Technical & applications consultancy.
Parallax	Consultancy & Training	\$750 poa	Introductory APL, APL for End-user & Advanced Topics in APL
QB On-Line	Consultancy	200	Specialising in Banking, Financial & Planning Systems.
I.P. Sharp	Consultancy	poa	Consultancy & support service world-wide.

**OTHER PRODUCTS**

COMPANY	PRODUCT	PRICES £	DETAILS
APL People	Employment Agency	poa	Permanent employees placed at all levels. Contractors supplied for short/long-term projects, supervised.
Mine of Information	APL Book Service		See booklet
I.P. Sharp	Productivity Tools	poa	Utilities for systems, operations, administration & analysts; auxiliary processors, comms software, international network.
	Databases	poa	Financial, aviation, energy and socioeconomic.

## VENDOR ADDRESSES

COMPANY	CONTACT	ADDRESS & TELEPHONE No.
Analogic Corporation	Denise Favorat	8 Centennial Drive, Centennial Industrial Park, Peabody, Mass. U.S.A. 01961 ☎ 617-246-0300
APL 385	Adrian Smith	Brook House, Gilling East, York. ☎ 04393-385
APL People	Valerie Lusmore	17 Barton Street, Bath, Avon. ☎ 0225-62601
APL Software Technology	John Hagger or Per Hultin	14 Rosewood Avenue, (UK) Ltd. Alveston, Bristol BS12 2PP. ☎ 0454-415737 46 Vicarage Hill South Benfleet, Essex SS7 1PB. ☎ 03745-60501
Beta-plan APS	Kim Andreasen	Stengrade 75, DK-3000 Helsingor, Denmark. ☎ 45 221 48 48
Boeing Computer	Anne Harding	19 Fitzroy Street, Services (Europe) Ltd., London W1P 5AB. ☎ 01-631 0808
Anthony Camacho		2 Blenheim Road, St. Albans, Herts AL1 4NR. ☎ St. Albans 60130
Cocking & Drury Ltd.	Romilly Cocking Brian Drury	16 Berkeley Street, London W1X 5AE. ☎ 01-493 6172 155 Friar Street Reading RG1 1HE. ☎ 0734-588835
DataTrade Ltd.	Tony Checksfield	38 Billing Road, Northampton, NN1 5DQ. ☎ 0604-22289
Delphi Consultation Ltd.	David Crossley	Church Green House, Stanford-in-the-Vale, Oxon SN7 8LQ. ☎ 03677-384
Dyadic Systems Ltd.	Peter Donnelly	Park House, The High Street, Alton, Hampshire. ☎ 0420-87024
E & S Associates	Frank Evans	19 Homesdale Road, Orpington, Kent BR5 1JS. ☎ 0689-24741
Farnell International Instruments Ltd.	R. Fairbairn or Roger Attard	Jubilee House, Sandbeck Way, Wetherby, W. Yorks. ☎ 0937-61961 Davenport House, Bowers Way, Harpenden, Herts. ☎ 05827-69071
FASTCODE	Andrew Dickey	P.O. Box 281, Croton-on-Hudson, New York 10520, U.S.A. ☎ (914) 271-3200
General Software Ltd.	M.E. Martin	22 Russell Road, Northolt, Middx. UB5 4QS. ☎ 01-864 9537
H.M.W. Programming Consultants Ltd.	Ken Jackson	142 Feltham Hill Rd, Ashford, Middx. TW15 1HN. ☎ 07842-41232
Holtech Ltd.	Jan Bateman	'O' Block 4th Floor, Metropolitan Wharf, Wapping Wall, London E1 9SS. ☎ 01-481 3207
IBM UK Ltd	Chris Sell	PO Box 32, Alencon Link, Basingstoke, Hants. RG21 1EJ. ☎ 0256-56144
Inner Product Ltd.	Dominic Murphy	Eagle House, 73 Clapham Common Southside, London SW4 9DG. ☎ 01-673 3354
Mercia Software Ltd.	Gareth Brentnall Barrie Webster	Aston Science Park, Love Lane, Birmingham B7 4BJ. ☎ 021-359 5096
MetaTechnics Systems Ltd	John Stenbridge David Toop	Unit 216, 62 Tritton Road, London, SE21 8DE. ☎ 01-670 7959
MicroAPL Ltd.	Jonathan Matthews Paul Thornton	Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London SW8 5BP. 01-622 0395
Mine of Information	Richard Ross-Langley	PO Box 1000, St. Albans, Herts AL3 6NE. ☎ 0727 52801
Modern Business Technology Ltd. (MBT)	Michael Branson	P.O. Box 87, Guildford, Surrey GU4 8BB ☎ 04868-23956
Parallax Systems Inc.	Kevin Weaver	60 West 9th Street, New York, New York 10011, U.S.A. ☎ 212-475-4001
Portable Software	Richard Smith	60 Aberdeen Ave, Cambridge, Mass. U.S.A. 02138. ☎ 617-547-2918
QB On-Line Systems	Philip Bulmer	5 Surrey House, Portsmouth Rd Camberley, Surrey, GU15 1LB. ☎ 0276-20789
Shandell Systems Ltd.	Maurice Shanahan	12 High Street, Chalfont St. Giles, Bucks HP8 4QA. ☎ 02407-2027
I.P. Sharp Associates Ltd.	David Weatherby	10 Dean Farrar Street, London SW1. ☎ 01-222 7033
Tektronix UK Ltd.	Paul Morgan	Fourth Avenue, Globe Park, Marlow, Bucks SL7 1YD. ☎ 06284-6000



**APL BOOKLIST****List of books stocked and on sale**

Title	Price	UK Post & Pack (£)
An Interactive Approach, Gilman & Rose	24.35	2.65
APL: An Introduction, H. Peele	15.00	2.25
Introducing APL to Teachers, K. Iverson	3.00	.40
Introduction to APL for Scientists & Engineers	3.00	.40
An APL Notebook, Barrie Wetherill	1.90	.50
A Source Book in APL, A. Falkoff & K. Iverson	8.75	1.80
APL: Design Handbook for Commercial Systems, Smith	13.10	1.50
APL and Insight, P. Berry and G. Bartoli	4.50	.55
APL in Practice, STSC	40.00	2.50
APL: The Language and its Usage, R. Polivka & S. Pakin	36.00	2.70
APL Quote-quad the Early Years	31.50	2.70
APL Business Technology '83 Proceedings	11.20	2.70
APL SV Reference Card	.50	.50
Designing APL Systems, Toronto Press Procs '82 Vol 2	8.50	1.80
FinnAPL Idiom Library	11.20	1.30
Sharp APL Reference Book	1.50	.50
Sharp APL Reference Manual, P. Berry	10.50	2.70
Yale University Idiom List	2.00	.50
APL in Exposition, K. Iverson	3.00	.50
Algebra, K. Iverson	12.00	2.30
Solutions to Algebra, J. Iverson	3.95	.50
Elementary Analysis, K. Iverson	13.50	1.50
Resistive Circuit Theory, R. Spence	20.00	2.80
Star Map, P. Berry & J. Thorstetsen	6.00	.50
Reliable Software through Composite Design, Myers	15.00	1.30
APL Lapel Pin with gripping backplate	2.00	.50
Whizzbangs, Volume II, R. Sykes	10.00	1.80
Whizzbangs, Volume I, R. Sykes	10.00	.40
Information Centres Toronto Procs '84	8.00	1.80
Application Systems in APL, Gibson Levine Metger	30.00	2.70

***Please order direct from Mine of Information:***

APL Book Service, PO Box 1000, St. Albans, AL3 6NE, U.K. ☎ 0727-52801

This list replaces all previous lists

Prices are in pounds sterling and are as at 1st January 1986

Prices are subject to change without notice.

Terms: Access & Visa accepted, or sterling cheque with order;

Outside UK add an *extra* amount per book –

£1 to Europe, £2.50 Africa & Middle East, £5 elsewhere (all sent airmail).

# APL CONSULTANTS

LONDON & READING

---

**Account Managers** (6 years+) to **23K**

---

**Senior Consultants** (4-6 years) to **19K**

---

**Consultants** (2-4 years) to **16K**

---

**Junior Consultants** (1-2 years) to **11K**

---

Are your APL skills and potential being recognised and rewarded?

Cocking & Drury consultants have been implementing successful decision support applications for 10 years, with clients who appreciate the productivity benefits of APL.

In our professional team you will experience a range of APL environments - APL\*Plus, VSAPL, APL2 and Unix, on both mainframes and micros. You will also be developing systems which, increasingly, need to interface with non-APL Information Centre products.

Of course as the leading APL consultancy, in a rapidly expanding market, we offer a rewarding career with first class benefits - profit sharing, free health insurance, and a non contributory pension.



For further details call Brian Drury on 01 493 6172

**COCKING & DRURY LTD.**

16 Berkeley St, London W1X 5AE

# Software Review: MicroSPAN

*by Anne Wilson and Steve Phillips*

## Introduction

MicroSPAN is a self-teaching system for computer-aided instruction in APL, written by MicroAPL. The version we used was MicroSPAN/PC. The package is held on three disks, containing the workspace and two files. It requires a PC with an APL chip, at least one floppy disk drive and 256 bytes of RAM. Apart from the package itself you need APL\* PLUS/PC Release 2.0 or above.

MicroSPAN is meant for a variety of people ranging from the computer illiterate to those who know almost everything – MicroSPAN has set itself a difficult task if it is to be simple enough for the ignorant without being simplistic for the educated. Unlike practically every package I have come across it does succeed; I think this is because it behaves like a courteous Englishman rather than a matey American, so that you do not mind being told things you already know because it is so polite and does not patronise you! MicroSPAN is fun to use, mainly because it is written with a sense of humour. Reassuring words lead the student through seemingly obscure matrix operations, and a generally light-hearted attitude (often at the expense of APL programmers) helps them to feel like carrying on, no matter how bemused they are by the latest set of squiggles and dashes.

## Presentation

The course is presented in the form of an electronic book, with chapters and sections; there is a good index which enables you to jump straight into the section in which you are interested. I have not worked out how it achieves the effect, but it seems to present all the topics of APL in such a sensible logical order that it all seems very simple; perhaps this merely reflects my training which was a selection of bits of courses plus Gilman and Rose?

As well as clear explanations of the operations, you are presented with examples which execute before your very eyes: a test at the end of each chapter is mainly composed of multiple choice questions. There are good facilities for going backwards, forwards and jumping around the book, so that the student may revise any topic he wishes. On completion of the course our programmers are fully competent to produce their own functions and programs.

## Why use a PC?

Our requirement was for a package for use by experienced programmers with no knowledge of APL, or of PCs, which would enable them to write and amend APL programs on an IBM mainframe. Secondary requirements were that it should be efficient in teaching a comprehensive subset of APL; have a reasonable number of examples; not take too long to complete; and need minimal effort on the part of the training advisor. We were not looking particularly at PC packages, but this one seemed to meet all our requirements.

Although our students were eventually to use APL on a mainframe we considered the advantages to be gained from using a PC would compensate for the disadvantages. Because PC time is not chargeable the student feels free to experiment and revise; in addition there are actual cost savings. Once in APL there is no noticeable difference between PC and mainframe. We do, however, handicap our students by giving them a keyboard with red stickies which are not all on the correct keys ...but they never complain.

The last two chapters in MicroSPAN deal with PC enhancements and PC files, but none of the earlier chapters are PC specific.

### **Student Performance**

Most of us find it difficult to remember how long a course takes, but one of our programmers has it accurately recorded! He took 710 minutes to do the first eight chapters, and found that sessions of an hour a day suited him best.

My personal benchmark for any text or package on APL concerns the way in which it describes inner products; in order to pass my test it must introduce matrix multiplication with its row-into-column rule – MicroSPAN passed with flying colours. Without being overtly mathematical it conveys the flavour of APL as a language for expressing mathematical concepts. I looked for, but could not find, a single mistake.

My criticisms all lie in the field of presentation; although these do not detract from the usefulness of the course they do distract the student, and should respond to fairly trivial amendments. Firstly, although we used a colour screen I had a strong suspicion that some parts of the course had been originally in mono and the colour was added later, and not very skilfully. Secondly, instructions to mount a new disk do not always say which disk; changing disks is enough of a nuisance without having to think which one goes next! Thirdly, in common with most training packages it makes noises, and while there are instructions for switching off the menu bleep, there are no such instructions for preventing the electronic fanfare or raspberry which follows each of the student's answers (although it can be done). In a professional, open-plan programming office it can best be considered as an embarrassing (noise) nuisance, and at worst deters the student from doing the questions at all!

### **Summary**

MicroSPAN is one of the best automated training packages I have come across, both in presentation and in its scope. For anyone with access to a PC it should certainly be considered as a possible APL tutor and as a reference manual.

# APL In Action

Manchester  
England

7-11 July  
1986



# APL 86



The British  
Computer  
Society



On behalf of the Organising Committee I am pleased to bring you the latest information on APL86. As our preparations become finalised we are pleased to see an attractive and informative programme taking shape — this is definitely the conference you cannot afford to miss!

Our theme — APL in Action — indicates that both the presentors of contributed papers and exhibitors will be providing more demonstrations than ever before.

Whether you are a system developer or a user, a beginner or an expert, there is much to interest you.

Bookings have surprised us by beginning earlier and in greater quantity than in previous years. If this trend continues we may sell out! Book early to be sure of your place.

I look forward to meeting you in Manchester.

**Dick Bowman**  
APL86 Chairman

### **Organising Committee**

<b>Chairman</b>	<b>Dick Bowman</b> , Central Electricity Generating Board
<b>Programme</b>	<b>Anthony Camacho</b> , Consultant
<b>Promotion</b>	<b>Philip Goacher</b> , British Computer Society
<b>Technical</b>	<b>Dave Ziemann</b> , Cocking and Drury
<b>Treasurer</b>	<b>Mel Chapman</b> , North Staffs Polytechnic
<b>Exhibition</b>	<b>Christine McCree</b> , Beechams
<b>Administration</b>	<b>Cathy Dargue</b> , Tesco
<b>Secretariat</b>	<b>Josy Roberts</b> , BISL

## Plans for APL 86

Plans for APL86 are now so well advanced that we can offer you abstracts of the contributed papers, a timetable and a description of tutorials and some invited talks.

There are a few things we can't tell you yet - we are still finalising arrangements for invited speakers, prizes, competitions and what I suppose we should call side shows. And of course we cannot tell what new products will be launched at the exhibition.

What is APL86 going to be like? We can offer the following facts:

- 1 You will be able to learn more at APL86 than on any week's professional development seminar or course. There are three full streams of activities to choose between and one stream consists entirely of tutorials.
- 2 You can choose what you learn. The whole conference is timed in 45 minute sessions: the lecture theatres open on to a common concourse and you can change streams at every break.
- 3 There is something for you in every subject that APL is used for, from stock control to typesetting, from graphics to music, from Laura Ashley to Nuclear Physics, and there are more varieties of machine and of APL than ever before.
- 4 You can see and hear better - the theatres have raked seating and are equipped with the latest visual aids. More of the papers will be accompanied by demonstrations with these aids than ever before; so many indeed that we have equipped two theatres with large screen BARCO colour VDU projectors.
- 5 All the foremost vendors of APL hardware and software are exhibiting: you will be able to see and question them for the whole week and by Friday 11 July, if you wish, you will be really well informed about what the market has to offer.
- 6 In addition to the Conference Proceedings (46 contributed papers out of 166 abstracts offered to us) which are full of valuable material, we intend to publish a second volume of Tutorials and other contributions. These will contain extended tutorial material and such useful things as Alan Graham's APL2 Idiom List.
- 7 For our software exchange we already have offers of a row of micro-computers. Bring or send your software for any implementation of APL and a six by four inch (15 by 10 cm) advertisement for us to put on the wall and let other delegates see what things you can do with APL. Blank disks will be on sale and we will allow commercial software samples if they are free.
8. The bookstall will be run by a professional bookseller: they will handle all the APL books they have heard of and a good many rarities will also be on sale (back numbers of APL magazines, Proceedings of little-known APL conferences, private venture publications and so on). Anyone with APL books for sale please contact us.

## INVITED SPEAKERS, PLENARY SESSIONS AND OTHER ATTRACTIONS

Our list of invited speakers is not yet final. We are still adding to it.

The conference will be opened by Dr. K.E. Iverson. His subject is "The Split in APL". Most APLers would like to understand the differences between the enhanced versions of APL and few of us have the opportunity to find out for themselves. Dr. Iverson's introduction will suit people who have only ever used a simple APL and those who have some experience of enhanced versions. His aim is to clarify matters so that listeners can appreciate both the theoretical basis and the practical effect of the differences (and of the different states of implementation.)

On Tuesday morning we have a panel session on "The APL Entrepreneur". Robert deCloss and Timo Seppala are among the members of this panel which is being arranged by Romilly Cocking.

Other panel sessions are on the ISO standard, on Graphics (organised by Theo Formanek with Giorgio Faconti's help), on Typesetting APL and on New Implementation.

On Thursday 10 July from 2 pm to 3 pm Povl Skifter, the Financial Director of Bang & Olufsen will tell us about the Bang and Olufsen Data Workshop. The APL community is sometimes a bit too inward looking. Mr. Skifter will bring us a view from the outside and tell us what benefit the Data Workshop brings to his company and why it was done in APL. His talk will be followed, for those who are interested in more of the detail, by a paper from Gert L. Moeller, who worked on the project.

There are two debates: in one Neil Mitchison will suggest that APL is not yet suitable for AI; in the other Richard Nabavi will argue that the enhancements to APL are doing nothing to help persuade more people to use it. These debates are formal arguments; they begin with a speech in favour from the proposer, and one against from the opposer. Then a seconder on each side makes a short speech at which stage the chairman will invite contributions from "the floor of the house", in other words, the audience. The last few minutes are taken up by a summing up: first the opposer and finally the proposer, and at the end there is a vote to see whether the "Motion before the House" is agreed or not.

Our closing plenary session will be run under the chairmanship of Mr. Roy Sykes, of Whizbang fame (now Managing Director of Optimization). We are sure of a lively close to the conference under his guidance.

### A TUTORIAL VOLUME TOO!

The people giving tutorials and invited speakers have all been asked to contribute some suitable text to the tutorial volume. This will be a companion to the Proceedings, but whereas the proceedings should be papers of original work, the tutorial volume can include material which is only partly new or which might have been given out at the papers presentation as a handout. We can also allow items of greater length in the tutorial volume (such as Alan Graham's APL2 idiom list, which was too long for the proceedings).

Both the tutorial volume and the proceedings will be produced in the same format as previous proceedings.



# APL86 TIMETABLE

Sunday 6 July

Education Day

**A special event for those interested in  
 APL and Education**  
**A day of presentation and hands on experience  
 on micro-computer versions of APL**  
**Designed to introduce  
 people in education  
 to APL and APL users  
 and persuade them that it  
 is better than other languages both  
 to teach and to use in teaching other subjects.**  
**Details of speakers and timetable  
 available on application.**

Monday 7 July

	Stream 1	Stream 2	Stream 3
09.00	Registration Begins		
10.30	Exhibition Opens		
11.30	Opening Plenary Session Dr. K.E. Iverson "The Split in APL" Other speakers to be announced		
13.00	Lunch		
14.00	Intro to APL2	Berry/Pesch	Kee/Post
14.45	Polivka	Pesch	
15.30	Refreshment Break		
16.00	Intro to APL2	Landaeta	Dakin
16.45	Graham		Barnaud
17.30	Exhibition Closes		
18.00	Civic Reception		
18.00 - 19.15	Civic Reception		
19.30 - 20.30	Dinner		

Notes on the tutorials (stream 1) and abstracts for the contributed papers (streams 2/3) follow the timetable and are listed in alphabetical order of author/speaker.

## Tuesday 8 July

	Stream 1	Stream 2	Stream 3
09.30	APL2 & the IC	Thomson	Bastian
10.15	Ostling/Helgason	Dorrer	McKee
11.00	Refreshment Break		
11.30	Generalised Linear Models Sykes	"The APL Entrepreneur"	Gustafsson
12.15			Groner/Cook
13.00	Lunch		
14.00	APL & Assembler on the PC Dickey	APL enhancements don't help spread of APL - Debate	Rudd
14.45			Soop
15.30	Refreshment Break		
16.00	APL/PC A P s Alfonseca/Selby	ISO Report and Panel Discussion	Searle
16.45			Sullivan/Fordyce
17.30	Product Forums — The exhibition closes at 19.15 Dinner		
17.45 - 19.15			
19.30 - 20.30	Real Ale Evening ( <i>For connoisseurs of the hop</i> )		
20.30 -			

## Wednesday 9 July

09.30	A 2nd course in APL Jaffe	Weigang	McGowan/Parker
10.15		Christopher/Ching	Butterworth
11.00	Refreshment Break		
11.30	APL2 & Comp Fns Latterman Idioms - Graham	Jernigan/Eisner	Naugle
12.15			Wiedmann
13.00	Lunch		
14.00	APL2 & AI Brown, Eusebi and Guerreiro		Stengl/Gfeller
14.45			Benkard
15.30	Refreshment Break		
16.00	APL2 & AI Continued	Livingstone	
16.45	Product Forums — The Exhibition closes at 19.15 Dinner		
17.45 - 19.15			
19.30 - 20.30			

## Thursday 10 July

	Stream 1	Stream 2	Stream 3
09.30	Teaching Stats with APL Hawkes/students	Bernecky	Pardee/Weintraub
10.15		Yoshino	Mayhew
11.00	Refreshment Break		
11.30	Formanek & Graphics Panel Discussion	APL is not suitable for AI – Debate	Chua/Winton
12.15			Brown
13.00	Lunch		
14.00	Plenary Session: Bang & Olufsen's Financial Director – "Bang & Olufsen's DATA WORKSHOP – benefits from APL"		
15.00	APL for Thoughts & Dreams: Smith	Moeller	Alfonseca
15.40	Refreshment Break		
16.00	Smith continued	Sandi	Brown
16.45		McDonnell	Eusebi
17.30			
17.45 - 19.15	Product Forums – The Exhibition closes at 19.15		
20.00 - 22.00	Conference Banquet		

## Friday 11 July

09.30	Howland/Feldberg & Typesetting Panel Discussion	Goldsmith	Udo
10.15		Holmes APL Calculator	New Implementations
11.00	Refreshment Break		
11.30	Closing Plenary Session and Award Giving Roy Sykes and a round up of APL86 Other speakers to be announced		
13.00	Lunch		
14.00	Farewell – Exhibition Closed		
14.30	APL86, EuroAPL, SigAPL and other committee meetings by prior arrangement		

## TUTORIALS

Stream 1 of APL86 consists entirely of tutorials. All are given by experts in their field. Below is a summary of arrangements as they stand at the time of writing (end of March 1986).

- An Introduction to APL2 — Principles and Concepts* Mon. Stream 1 1400  
Ray Polivka aims to suit the needs of people relatively new to APL. Anyone who knows a bit of standard APL should find this a splendid introduction to APL enhancements.
- APL2 by Example* Mon. Stream 1 1600  
Alan Graham will pick up the thread where Ray left it and will present the power and elegance of APL2 in example after example.
- APL2 and the Information Centre* Tue. Stream 1 0930  
Bjorn Helgason and Kenneth Ostling will review the features of APL2 which help in the Information Centre — component files (object library), graphic images and the use of the SQL workspace and the PC with IC/1.
- Generalised Linear Models* Tue. Stream 1 1130  
Dr. Alan Sykes will demonstrate and explain the advanced software in use at the Department of Management Science and Statistics at University College of Swansea.
- APL and Assembler on the PC* Tue. Stream 1 1400  
Andrew K. Dickey will explain why you might want to call PC assembler routines from APL and how to go about doing it — with examples in APL\*PLUS/PC.
- APL PC Auxiliary Processors* Tue. Stream 1 1600  
Manuel Alfonseca and David Selby will explain how IBM PC APL can use Auxiliary Processors to do work the interpreter cannot handle and will deal with the uses of eleven of those currently available, including interfaces to non-APL programs, graphics, a screen handler, DOS files, the ports (and RS232 and GPIB/IEEE) and even a music interface.
- A Second Course in APL* Wed. Stream 1 0930  
Stephen Jaffe submitted this as a paper: we liked it so much we asked him to give a sample rather than a paper about it. He will give guidance for non-experts in the use of multidimensional arrays and the less obvious uses of grade and scan.
- APL2 and Complementary Functions* Wed. Stream 1 1130  
Dieter Latterman will explain how the power of APL2 makes many standard utilities obsolete. He will teach you how to write those utilities you still need as defined functions which behave like and can be used just like primitives, and how nested arrays are for many purposes more suitable and natural to handle than simple matrices.

*APL2 Idioms*

Wed. Stream 1 1215

Alan Graham's second tutorial was submitted and accepted as a paper but the full list of wonderful idioms was so long that we have had to put it into the Tutorial volume. In this session he will explain some of the most useful and interesting idioms on the list.

*Artificial Intelligence Using APL2*

Wed. Stream 1 1400

The two tutorials will be given by Ramiro Guerreiro, Ed Eusebi and Dr. James Brown (IBM's Chief Architect of APL2). The first part is the Fundamentals of AI (for newcomers to it) such as propositional and predicate logic, rules of inference, resolution and search strategies. Part two is about the implementation of these ideas in APL2 using nested arrays and recursive function sets. The results will be compared with PROLOG and LISP and you can have listings of the APL2 functions.

*Teaching Statistics with APL*

Thur. Stream 1 0930

Professor Alan Hawkes and students will teach you some statistics and show how easy it is to convert standard formulae into APL. Some student projects may also be demonstrated if there is time.

*Graphics with APL — a tutorial and panel discussion*

Thur. Stream 1 1130

Theodor Formanek will show how to do simple graphics with APL — the geometry needed and the data structures that are suitable. The lecture part will be suitable for people who have not previously used APL for graphics. This will be followed by a panel session with contributions (and we hope some demonstrations) from a number of people who have done spectacular work with APL Graphics. Come and ask them questions!

*APL A Language for Thoughts and Dreams*

Thur. Stream 1 1500

Phil Smith aims to startle people out of their complacency to overcome old habits of thinking and to reveal the magic of APL to newcomers. Come and see why his sessions have such a wonderful reputation.

*How to Typeset APL (and other things)*

Fri. Stream 1 0930

This is more than a tutorial: it is two contributed papers (from John Howland and Ian Feldberg) with quite different approaches to producing mixed APL and Roman text, followed by a panel discussion in which everyone concerned with printing or publishing APL will want a say — not to mention the authors.

## Alphabetical List by Author Name of All Contributed Papers to APL86

- M. Alfonseca and M.J. Tobar** Thur. Stream 3 1500  
*Extending APL to Logic Programming*  
Abstract: The limitations that make IBM's PC APL slow at logic programming were analysed. The best option for speeding it up was to give it an auxiliary processor with suitable logic functions. It can now compete with PROLOG. The paper also describes experience in use.
- A.M. Barnaud, B. Mailhol & Y. Le Borgne** Mon. Stream 3 1645  
*Development of a Portable Software Product for Library Automation.*  
Abstract: The paper describes the use of APL alongside assembler and C to produce a system to meet the needs of French public libraries. It handles cataloguing, bibliographical enquiries and follow-ups on lending and acquisition. It runs on IBM mainframes and Unix machines.
- C. Bastian** Tue. Stream 3 0930  
*The Scope of APL in Nuclear Measurements*  
Abstract: The whole range of data acquisition and analysis in experimental nuclear physics can be covered with a few APL workspaces. Typical steps in the treatment of statistical data are highlighted and compared with solutions in other languages. There is advice for APLers wishing to compete against FORTRAN.
- J. Philip Benkard** Wed. Stream 3 1445  
*Analysis of Function Application in Deep Arrays*  
Abstract: The proposed extension suggests that arrays of functions as well as arrays of data are possible in APL2, although they are more complex. The paper examines the case of mixed function arrays and draws a parallel with the Sharp concept of function rank.
- Robert Bernecky** Thur. Stream 2 0930  
*APL: A Prototyping Language*  
Abstract: A thorough analysis of prototyping and APL's advantages over other languages for this task. Hierarchical and hybrid approaches to modelling systems of various degrees of complexity are illustrated with examples to show that APL makes for A Pleasant Life!
- Michael J.A. Berry and Roland H. Pesch** Mon. Stream 2 1400  
*Style and Literacy in APL*  
Abstract: The shibboleth that APL instruction lines should be short is amusingly contrasted with the practice of many skilled APL writers. Then it is suggested that the genius of the language calls for line length to be appropriate for the task in hand and many examples are given of 'four lines good two lines better' (or even one line best).
- Dr. James A. Brown et al** Thur. Stream 3 1600  
*Logic Programming in APL2*  
Abstract: This paper explores the representation of Logic using the data structures of APL2. Some LISP and PROLOG features are translated into APL2 and developed to create a complete PROLOG-like programming environment, with an inference engine and knowledge base. Search strategies are also discussed.

**Dr. James A. Brown**

Thur. Stream 3 1215

*Writing Multi-user Applications in APL2*

Abstract: This paper presents a general scheme for rapid implementation of multi-user applications showing the special features of APL2 that make it suitable for this.

**Richard W. Butterworth and Robert A. Stephan**

Wed. Stream 2 1215

*DELIS: A Decision Support System Generator for Frequency Data*

Abstract: DELIS displays frequency data as tables or graphs on many types of display and output device. It provides query and display functions, a method of passing queries to be processed and functions for automatic analysis and response to the queries. The APL is plain VSAPL. Frequency data is extracted by PL/1 and passed to the system in APL file format.

**Wai-Mee Ching**

Wed. Stream 2 1030

*An APL/370 Compiler and Some Performance Comparisons with APL*

Abstract: An experimental 370 compiler works with a subset of APL chosen for engineers and scientists. The compiled code runs several times faster than the interpreter and is of about the same quality as is produced from FORTRAN. The paper explains how the front end analyses the APL dataflow to produce parse trees, graphs and tables and how such efficient 370 code is generated.

**Thomas W. Christopher and Ralph W. Wallace**

Wed. Stream 2 1000

*Compiling Optimized Array Operations at Run-Time*

Abstract: Some APL interpreters hold arrays as descriptors which point to a block of memory holding the data sequentially. Scalar functions therefore can be translated into machine code which has fewer loops than the number of dimensions to the array. This paper describes an analyser which constructs optimised array operations at run time from the descriptors of the arrays. The performance and results from experiments are analysed.

**Y.S. Chua and Charles N. Winton**

Thur. Stream 3 1130

*Automated Application Programming Environment*

Abstract: Small businesses can justify the low cost of tailor-made applications in APL if ways can be found to reduce the programming cost of the user and file interfaces. The paper offers file and screen utility functions to help solve this problem and illustrates their value and portability with an example.

**Adam Dakin**

Mon. Stream 3 1600

*LAPLAN - A Corporate Planning Model for Laura Ashley*

Abstract: A really complex business problem was handled by a group of APL novices and a successful system developed on a microcomputer in a remarkably short time. The detailed steps are explained. The whole is an excellent example of the strengths in APL and its value in prototyping.

**Prof. Dr. Ing. Egon Dorrer**

Tue. Stream 2 1015

*From Tensor and Suffix Notation to APL*

Abstract: Tensor and suffix notation reveals the simplicity and symmetry of some otherwise complicated and obscure multivariate formulae. Matrix algebra and APL normally would hide the structures of interest but the paper shows how to represent the notation in APL and the advantages and disadvantages of APL for this work are made clear.

Ed Eusebi and Dr. James A. Brown

Thur. Stream 3 1645

*APL2 and AI: A Study of Search*

Abstract: To demonstrate the suitability of APL2 for AI some paradigm AI problems are solved. They show logical inference, state space representations and other AI methods. A set of functions for searching a database of rules and facts will be explained and demonstrated. Enhancements to cater for heuristic, bidirectional and parallel search are shown.

Ian Feldberg

Fri. Stream 1 1015

*TEXT: Publication-Quality Characters Come to APL Graphics*

Abstract: TEXT is an APL workspace that creates and manipulates all kinds of characters on the screen or a plotter. It can be used with several plotting packages. Characters can be scaled, rotated justified etc. The workspace will be demonstrated.

Martin Gfeller and Monika Stengl

Wed. Stream 3 1400

*Open Fullscreen Systems*

Abstract: An open fullscreen system combines two advantages a) presenting the user with data and prompts a whole screen at a time, b) providing access to the power of APL and all the functions in the workspace. Systems can be designed which allow any command to be given whatever the current state of the system. Non-APL users can be given access to some of the same power through cover functions which become a command language like keyword APL.

Leslie H. Goldsmith et al.

Fri. Stream 2 0930

*LOGOS: an APL programming environment*

Abstract: Logos provides support to the user, developer and manager of application systems. It leads to increased programmer productivity and more robust software by providing a centralised uniform interface. Logos resides in the development workspace and looks like a small holdall containing a huge variety of tools and facilities shared in a controlled way with other programmers.

Alan Graham

Wed. Stream 1 1215

*The Usage of APL2 – Common Idioms and Problem-solving Techniques*

Abstract: This collection – a must for anyone using an enhanced version of APL – contains many examples which are shown in several forms from the fully general to those which rely on the shapes of their arguments. There is a discussion of techniques and a well annotated list of idioms.

L.H. Groner and J.H. Cook

Tue. Stream 3 1215

*Modelling the Arithmetic of Statistical Distributions*

Abstract: The paper reviews various kinds of extended arithmetic and their advantages. Statistical modelling can be approached as an extended arithmetic of distributions. An APL prototype leads to a suggestion for the result of the Quad-CR of a primitive function.

Stig Gustafsson

Tue. Stream 3 1130

*Quaternions and Homogeneous Co-ordinates*

Abstract: Real numbers, complex numbers and mathematical vectors are all special cases in the number scheme of quaternions (which are ordered quadruples of real numbers). Quaternions provide an attractive alternative to



matrix methods for transformations of geometric figures — especially rotations about arbitrary axes (such as are needed by robots). APL allows easy implementation of quaternions but a hybrid between quaternions and homogeneous matrices gives the best speed of transformation.

**John E. Howland**

Fri. Stream 1 0930

*Typesetting APL using a Macintosh.*

Abstract: The approach described uses the Macintosh and LaserWriter to produce commercial quality printing. The PostScript notation used by the LaserWriter is explained and examples of both bit mapped and analytic characters are given.

**Stephen B. Jaffe**

Wed. Stream 1 0930

*Topics for a Second Course in APL*

Abstract: After the introductory course and a little experience, what do newcomers to APL need? Here is part of a syllabus for the second course. It helps with multidimensional arrays, the less obvious applications of grade and scan and the creation of better user interfaces including the process of closing an open workspace and error trapping. The presentation will be a sample of the course.

**Robert Jernigan and Arie Eisner**

Wed. Stream 2 1130

*A Generalised Data Base Interface for APL*

Abstract: By the provision of a function through which all database calls are made it becomes possible to link APL with several different proprietary databases. Only subfunctions invisible to the APL programmer vary according to the database in use. The generalisation that made this possible is described. The system has been in use for over two years at John Hopkins University.

**H. Clark Kee and Roy L. Post**

Mon. Stream 3 1400

*Manufacturing Resource Planning on a PC Local Area Network*

Abstract: Eight work years and \$150,000 of hardware (12 PCs and a network with a 137 megabyte file server) make an astonishingly cheap and efficient modular construction for this system — its specification looks like a job for a large mainframe. Many micros make light work — APL is quite fast enough — the most heavily loaded unit is the file server. A comprehensive account full of lessons for people wanting similar cost effective results.

**David Landaeta**

Mon. Stream 2 1600

*A Notation for Manipulating Arrays of Operations*

Abstract: How should arrays of operations be incorporated into APL? This paper combines and builds on earlier suggestions and proposes to extend strand notation, to allow 'orders' of operation with ways of stepping between order levels and to extend the use of jot to allow niladic operator calls.

**D. Livingstone**

Wed. Stream 2 1600

*The Integration of Relational Database Algebra into APL*

Abstract: Instead of linking databases to APL (which involves inconsistencies and loss of generality), this paper proposes that APL is enhanced by the addition of relational database primitives. A relation would become a special case of generalised arrays. The paper concludes by proposing ways to improve machine efficiency in the use of such extensions.

**E.E. McDonnell**

Thur. Stream 2 1645

*A Perfect Square Root Routine*

Abstract: This is an example of the use of APL as a prototyping language to design and test algorithms for use in other languages. Here it is used on the method now incorporated into Sharp APL for calculating square roots. The problems of binary representation and how to minimise the inaccuracy in the result are explored. Some fascinating problems arose and were resolved in the testing of the method ("perfect" because the method is as accurate as the representation allows).

**J.E. McGowan and D.J. Parker**

Wed. Stream 3 0930

*Medium Term Forecasts of Half-Hourly Power Demand*

Abstract: A new forecasting methodology was developed and implemented in APL to provide these forecasts from records of demand by category, date, time, outside temperature etc. The system can estimate the effect of changing one or more of the variables and can display the results graphically.

**Terence J. McKee**

Tue. Stream 3 1015

*APL in Quality Business Management*

Abstract: The paper amusingly shows how people are accidentally misled by graphs and statistics and gives advice on how to avoid such pitfalls. The data may be accurate but care needs to be taken (guidelines provided) if the decision to take remedial action is to be correctly made.

**Loren B. Mayhew**

Thur. Stream 3 1015

*Increasing Productivity with ISPF/APL2*

Abstract: Interactive System Productivity Facility provides a consistent way for users to edit, browse, create and delete datasets. It deals with things a screen or 'panel' at a time. The paper explains how to use ISPF from APL2, so giving users the environment they are used to and greatly shortening the process of providing the user interface. Live examples will be demonstrated.

**Gert L. Moeller**

Thur. Stream 2 1500

*A Logic Programming Tool for Qualitative System Design*

Abstract: This paper describes the boolean geometrical representation of rules for expert systems. The method is used to simulate the reactions of complex television and audio products to user commands. Ideas for new features can readily be tried out and designs refined to improve ease of use and eliminate snags. The techniques have been used to develop Bang and Olufsen products.

**Richard L. Naugle**

Wed. Stream 3 1130

*APL Compilation and Interpretation by Translating to F83VEC*

Abstract: A way of overcoming APL's slowness or excessive memory usage is to translate the appropriate functions into another language. This paper considers one such method using a version (F83VEC) of reverse polish FORTH extended to use the APL character set and demonstrates the improvement in speed dramatically.

**Professor Otway O'M. Pardee**

Thur. Stream 3 0930

*Checkpoint - Restart for APL Applications*

Abstract: APL programs can be made proof against crashes arising from external causes such as communications or equipment failure, if there is a reliable file system. The method is called 'checkpoint and restart' and can

easily be applied to any APL code and tuned for the best balance between protection and cost.

**Roland H. Pesch**

Mon. Stream 2 1445

*Empty Frames in Sharp APL*

Abstract: The rules for deriving the shape of the result from the shape of the arguments (and the function to be performed) are not reliable in the "empty frame" case when the shape of an argument contains a zero. Some functions return results programmers don't expect and dislike. Ways of overcoming the problem are reviewed and assessed.

**Jack Rudd**

Tues. Stream 3 1400

*APL Graphics Representation and Analysis of Space-based Observations*

Abstract: APL can process three dimensional data and produce views from space of the earth against a background of stars showing the line between day and night. Shuttle flights and 3D terrain maps are also shown. The paper describes techniques for deriving 3D models from multiple 2D satellite observations. Examples will be displayed in colour.

**Claudio Sandi**

Thur. Stream 2 1600

*Generalised Arrays and Shortest Path Problems*

Abstract: The paper presents an analysis of the classical "shortest path" problem using three different methods implemented both in VSAPL and APL2. APL2 is shown to provide a substantial improvement in run time for this type of problem, and particularly for one class of algorithm.

**John R. Searle**

Tue. Stream 3 1600

*Iota3: the Integration of Lotus Spreadsheets and APL*

Abstract: The paper describes a software system which integrates Lotus 123 and Symphony with APL. It allows spreadsheet users access to some of the power of APL and APL users to hold a spreadsheet as an extension of the workspace. APL \*PLUS/PC and IBM APL PC experience is described.

**Karl Soop**

Tue. Stream 3 1445

*Bringing Graphic Dialogues to APL*

Abstract: This paper proposes a canonical representation for pictorial data and a picture processor. These make picture processing easy to incorporate into the interaction between the user and the machine even for complex graphics. Experience with an APL2 prototype is described.

**Gerald A. Sullivan and Kenneth J. Fordyce**

Tue. Stream 3 1645

*Extending Worksheet Modelling Software with Expert System Technology*

Abstract: This is a description of a tool called Automated Ledger Book, which is a worksheet model generator with an equation manipulating expert system and a natural language interface. It can answer 'what if' questions and give justifications for its responses. The system will be demonstrated.

**N.D. Thomson**

Tue. Stream 2 0930

*APL2 - A Mathematician's Delight*

Abstract: APL has always claimed to be suitable for mathematics and to help the understanding of mathematical ideas. The APL2 enhancements have increased the affinities between APL and mathematics in many delightful ways, some of which are outlined in this paper.

**M. Udo et al.**

Fri. Stream 3 0930

*Japanese APL Language System on IBM Multistation 5550*

Abstract: APL's problems with an extended character set are insignificant compared with those of languages that need a range of characters that cannot be represented within a byte per character. This paper describes how the 10,000 Kanji characters were incorporated into IBM PC APL. The method could be adapted for other languages with large character sets.

**Jim Weigang**

Wed. Stream 2 0930

*Machine Oriented Languages in the APL environment*

Abstract: To enable APL to compete in efficiency with other languages we need a way to specify the machine operations in detail. The functions that use most time can then be optimised. APL itself is not ideal for this, as the paper explains. It then shows what is needed to make compiled functions available in a workspace and describes experience with several methods, including a language specially designed for the job (TABL).

**David M. Weintraub**

Thur. Stream 3 0950

*Assertive Comments in APL Programming*

Abstract: This proposal suggests that instead of describing what code is intended to do, comments should assert the current state of relevant variables at critical points in the code. This is shown to match the design method that many APLers use, which works from desired result to functions.

**Clark Wiedmann**

Thur. Stream 3 1215

*Field Results with an APL Compiler*

Abstract: This is a report on users' experience with an APL compiler, as gained from case studies. It deals with how long it takes to learn to use the compiler, how much faster processes run, how much extra development effort it takes, what parts of the workspace should be compiled and how the cost of compile time compares with the savings in run time.

**Matsuki Yoshino**

Thur. Stream 2 1600

*APL as a Prototyping Language: Case Study of Compiler Development Project*

Abstract: APL's advantages for such work are its conciseness, its excellent debugging and modelling facilities, the great power of enclosed arrays and the simple interface between the workspace and outer environment provided by shared variables. The strategy used is to write prototype functions in APL and by several steps to convert from high level APL to low level Pascal. The benefits of this approach are evaluated.

## THE SOFTWARE EXCHANGE

At APL86 there will be a forum where delegates can meet to exchange software. For each item of software offered one advertisement may be displayed in the Exchange. Whilst there is space available we will accept advertisements for samplers or commercial software as well as privately owned software. The only condition is that there is to be no charge.

A number of microcomputers (mainly IBM PCs) will be available and these may be booked in advance. There will be a full time attendant to make bookings and to assist delegates.

The Software Exchange will be open during exhibition hours (except when required for any demonstrations).

We aim to have enough machines to allow every delegate participating in the Exchange sufficient time to copy the software they want. We will have a supply of very reasonably priced floppy disks, in case you need extra.

There will be no restriction on what is offered, except that it should be useful to APLers. Bring along a single, elegant function or a raft of graphics pictures, public domain music or a complete stock control workspace. The more you bring the better the Exchange will be – and what might you find to take back with you? Here is your chance to show the world what you can do with APL.

Please Note: The Exchange is not available for copying software for sale.

Piracy and selling code are prohibited; persons caught attempting either will be debarred.

## THE EXHIBITION

The exhibition area is on two floors: on the concourse between the lecture theatres and on the floor below, next to the bar. The two floors are connected by a broad open staircase.

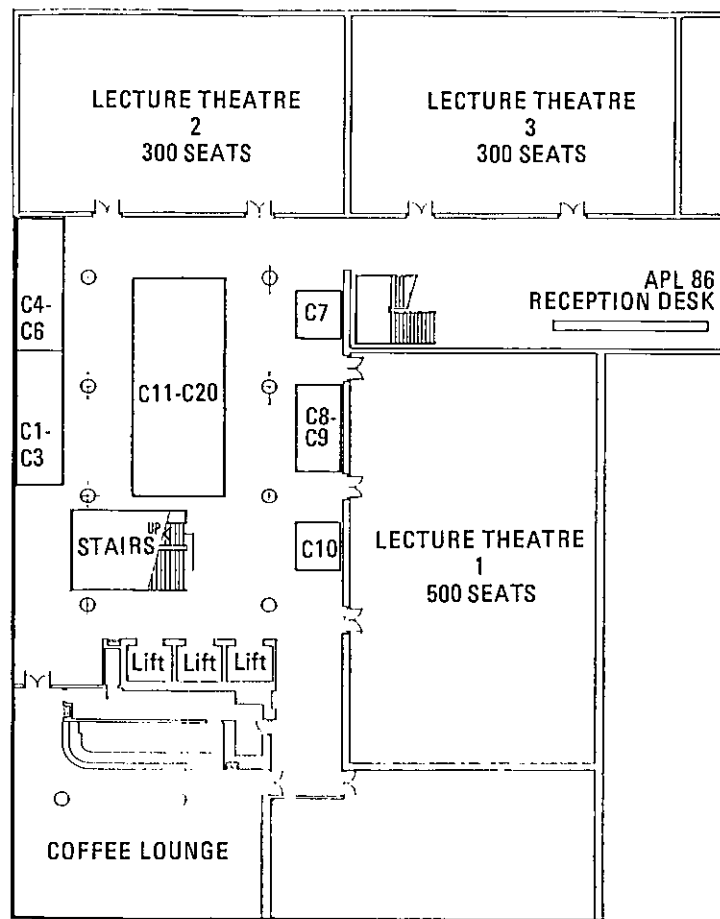
We have extended the exhibition area and already have more than thirty stands booked – the exhibitors include all the major APL vendors.

	<b>Stands</b>	<b>Exhibitors</b>
Upper Floor	C1 - C3	MicroAPL
	C4 - C6	Cocking & Drury
	C7	Mercia Software
	C8 - C9	Dyadic Systems
	C10	Analogic
	C11 - C20	STSC
Lower Floor	B1	Bookstall
	B2	Ampere
	B3 - B4	I P Sharp
	B5	BCS – British APL Association
	B6	ACM
	B7	APL87
	B8 - B11	IBM
	B12	APL People

You can be sure that all the exhibitors will be doing their best to interest you and show you their latest and best products.

The exhibition will be open the whole time during the conference, so there will be plenty of time to look at everything that takes your fancy. Anyone who attends APL86 can come away well informed about all the latest products from the main vendors.

For detailed expositions, and particularly for new and enhanced product launches, there are Product Forums. These are half hour presentations by vendors in the lecture theatres from 17.45 on Tuesday, Wednesday and Thursday. The schedule of Product Forums will be announced at the conference.



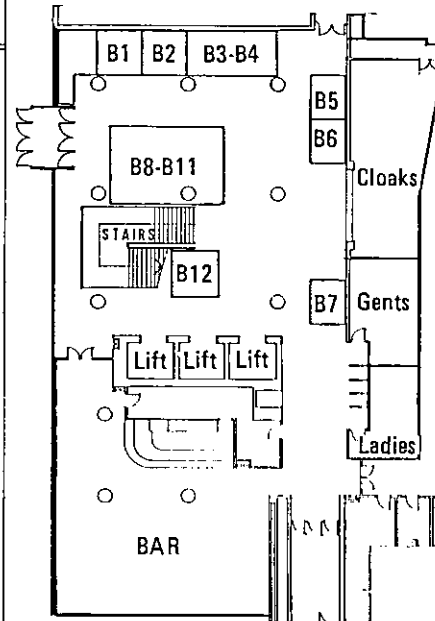
FIRST FLOOR 'C'

APL EXHIBITION  
7 - 11 JULY 1986

UNIVERSITY OF MANCHESTER  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
RENOLD BUILDING

Organised by  
THE BRITISH APL ASSOCIATION  
01 637 0471

revised layout 27.3.86.



GROUND FLOOR 'B'

## How to get to APL86

Manchester lies at the heart of the United Kingdom. From it radiates a vast network of air and surface routes, both national and international.

**By Air:** Manchester's international airport is the principal airport in the UK outside London. International flights are available to many parts of the world on the services of the large number of international airlines operating from the airport.

For APL86 delegates travelling from Europe, direct flights link Manchester with Belgium, France, West Germany, Italy, Ireland, The Netherlands, Scandinavia and Switzerland. Direct flights are also available to Canada and the USA.

Regular shuttle flights run between London Heathrow and Manchester to extend considerably the number of destinations that can be easily reached.

Buses linking the airport to the air terminal in central Manchester stop quite near the UMIST campus.

**By Rail:** With two Inter-City rail terminals, Manchester connects with all parts of the UK. For example, London is only two and a half hours away.

Discount fares are available to all delegates travelling to APL86 by train, and application forms will be sent to those who request them.

By taking advantage of cheap, daily return fares, accompanying persons can use the Inter-City rail network to travel to many interesting towns and cities – even to shop in London's West End.

**By Road:** Motorways give fast and speedy access to Manchester from almost any part of the country. There are direct connections to Scotland, the North, London and the South East with links to the East and West coasts, including the main channel ports.

The motorway spur linking Manchester to the main M6 motorway will bring APL delegates very close to the UMIST campus. Local maps will be supplied to those wishing to drive to the conference.

Please indicate on the booking form which form of transport you will be using so that we can send the appropriate information.

**British Airways:** The official airline for APL86 can help with group travel and fly/drive schemes, allowing delegates to spend a few days sightseeing before or after the conference. Contact British Airways for further information. A list of local British Airways offices is given on the following page.



The following are British Airways Offices that will be able to help to arrange your travel to APL86. Please call or write to the manager of your nearest office to see what he can offer you.

#### AUSTRALIA

33 King William Street  
Adelaide (Tel 2117566)

243 Edward Street  
Brisbane (Tel 2211599)

330 Collins Street  
Melbourne (Tel 6023000)

140 St Georges Terrace  
Perth (Tel 3225011)

64 Castlereagh Street  
Sydney (Tel 2335566)

#### AUSTRIA

Mr Strelka, Karntner Ring 10  
Vienna (Tel 657691)

#### BELGIUM

Mr Van Wezer  
Centre International Rogier  
9th Floor, Brussels  
(Tel 02/2194220)

#### BRAZIL

21st Floor, Avenue Rio Branco 108  
Brazil (Tel 021 800 6926  
or 242 6020 local)

#### CANADA

1021 De Maisonneuve West  
Montreal (Tel 2879282)

60 Bloor Street West  
Toronto (Tel 5952500)

1012 West Georgia Street  
Vancouver (Tel 112 800 2688090)

#### DENMARK

Vesterbrogade 2  
Copenhagen (Tel 01146000)

#### FINLAND

Mr Raymond, Keskuskatu 7, 4th Floor  
Helsinki (Tel 650677)

#### FRANCE

91 Avenue des Champ Elysees  
Paris (Tel 7781414)

#### WEST GERMANY

Mr Keil, Friedenstrasse 1  
Frankfurt (Tel 250121)

#### GREECE

10 Othnos Street  
Athens (Tel 105 57)

#### HONG KONG

Jardine Airways, Alexander House  
Chater Road, Kowloon (Tel (5) 775023)

#### IRELAND

112 Grafton Street  
Dublin 2 (Tel 686666)

#### ISRAEL

59 Ben Yehuda Street  
Tel Aviv (Tel (03) 229251)

#### ITALY

Mr. Guglietmetti, Via Bissolati 48  
Rome (Tel 479991)

#### JAPAN

Mr Terashima, Sanshin Building  
Yurakucho, Chiyoda  
Tokyo (Tel 03 214 4161)

#### MEXICO

Mr Sanchez, Edificio Alber  
Paseo de la Reforma 322  
Mexico City (Tel 525 9133)

#### NETHERLANDS

Mr Hage, British Travel Centre  
Hirsch Building, Leidseplein  
Amsterdam (Tel 229333)

#### PORTUGAL

Avenida da Liberdade 23/37  
Lisbon (Tel 360931/4)

#### SOUTH AFRICA

Anglo American Life Centre  
2 St George's Street  
Cape Town (Tel 252970)

NBS Building, 300 Smith Street  
Durban (Tel 3044741)

#### Nedbank Mall

145 Commissioner Street  
Johannesburg (Tel 331-0011)

#### SPAIN

Gran Via 68, Madrid (Tel 2475300)

#### SWEDEN

Mr Shaw, Norrmalmstorg 1  
Stockholm (Tel 08/2339000)

#### SWITZERLAND

Mr Stevens, Talacker 42  
Zurich (Tel 2114090)

#### USA

530 5th Avenue  
New York (Tel 212 687 1600)

## Accommodation

**UMIST:** Accommodation in the halls of residence on the campus at UMIST has been included in the delegate and accompanying person fees. Rooms are available for the duration of the conference only (6-10 July inclusive) and consist of single study bedrooms with washbasin. Shared bath/shower and WC facilities are available on each floor – usually serving four rooms.



Also included in the fees (both delegate and accompanying person) are all meals from the evening of Sunday 6 July to lunch on Friday 11 July. A cold buffet will be available for those arriving late on Sunday 6 July. All residents have access to kitchen facilities (tea/coffee provided) and to communal TV lounges.

A limited number of twin rooms are available (conference rate per person) and these will be allocated on a first come, first served basis. A full guide to the campus will be provided with confirmation of booking.

**Hotel Supplement:** Delegates wishing to stay at one of the nearby four-star hotels will be charged a supplement in addition to the conference fee. Please note that this charge is to supplement the amount allocated for accommodation in the conference fee. The rates per room, per night are £50 for a double/twin room and £36 for a single room. This price includes breakfast, private bath, telephone, colour TV, tea and coffee making facilities and VAT at 15%. Extras, such as drinks are settled with the hotel directly. These rates also apply to accompanying persons.

Hotel bookings must be made before 7 May, after which time delegates must make their own hotel arrangements.

Please indicate on the booking form if you wish to book a hotel.

**Facilities:** All delegates and accompanying persons will be entitled to use the UMIST sports facilities during the week of the conference. Automatic membership of the campus bars also applies.

## Registration Information and Terms

**Registration:** Delegates are requested to register as early as possible using the enclosed Registration Form. A separate form is required for each delegate — please photocopy the form as required. Completed forms together with payment in full should be sent as soon as possible to:

Conference Department (APL86)  
The British Computer Society  
13 Mansfield Street  
London W1M 0BP, England

Telephone 01-637 0471

**Fees:** Reduced fees are available to delegates holding membership of any of the following: British APL Association (BAA), British Computer Society (BCS), SigAPL and the Association of Computing Machinery (ACM). The rates in pounds sterling are:

	Before 6 July	On-Site
Delegate — Member	345.00	373.75
— Non-Member	373.75	406.50
Accompanying Person	181.25	210.00
Banquet (Accompanying Person)	20.00	20.00

Delegate and accompanying person fees include accommodation at UMIST and all meals from the evening of Sunday 6 July to lunch on Friday 11 July. In addition, the delegate fee includes attendance at conference sessions, access to the exhibition areas, conference documentation and attendance at the conference banquet. All fees are inclusive of Value Added Tax at 15%.

**Payment:** All fees are due with the application and must be in £ sterling. Cheques should be drawn on a British clearing bank and be made payable to the 'British Informatics Society Limited'. Confirmation of the booking will be sent to each delegate along with relevant maps and instructions. Early payment discounts will apply according to the date of the postmark on the envelope in which the application has been sent.

Payments still outstanding after 11 July will be subject to a 5% surcharge.

**Substitution:** A registered delegate may nominate a substitute providing that the conference office is informed no later than 4 July 1986. The amount of fees then due is according to the eligibility of the substitute, who will be invoiced for any difference owing.

**Cancellations:** Cancellations will be accepted only if received in writing not later than 6 June 1986. A 10% handling charge will be deducted from any refund.

**Programme:** The Organisers reserve the right to make changes to the advertised programme.

---


We can fly  
more people to  
more conferences  
than any  
other airline.

---



British Airways is the official airline for this conference. We are the world's largest international airline and in the best position to offer you the most comprehensive range of services from Concorde through to Economy travel. For further details, contact your local British Airways office or appointed Travel Agent.

**BRITISH AIRWAYS**

The world's favourite airline. 

---

## RECENT MEETINGS

This section of VECTOR is intended to document the seminars delivered at recent meetings of the Association, particularly for the benefit of those members based away from London who often find it hard to find the time to attend. It also covers other selected events which are likely to be of interest to the wider APL community.

We are dependent on the willingness of speakers to provide us with a written version of their talk, and we would remind them the "a picture's worth a thousand words". Copies of slides and transparencies will enhance their articles.

The Activities Officer (see inside back cover) will respond enthusiastically to offers from individuals who wish to contribute seminars and supporting papers.

This issue's coverage concentrates on the outside world with reviews of two major computing conferences held last Autumn - Computer Graphics '85 and Expert Systems '85. The January and March Association meetings will be covered in the next issue.

---

---

---

# APL ON YOUR IBM OR COMPATIBLE PC?

MERCIA SOFTWARE provide APL★PLUS/PC the best APL available for personal computers, together with a comprehensive range of services:

## SOFTWARE

APL★PLUS/PC Release 5	£695.00
APL★PLUS/PC Release 4 to Release 5 Upgrade	£150.00
APL★PLUS/PC Tools Volume 1	£325.00
APL★PLUS/PC Tools Volume 2	£125.00
MICROSPAN (APL Self-Teach Course)	£250.00

## TRAINING

Public and In-House APL★PLUS courses are our speciality

### INTRODUCTION TO APL★PLUS/PC

May 13, 14, 15. August 5, 6, 7. October 7, 8, 9. £350.00

### APL★PLUS/PC ENHANCEMENTS

June 11, 12. August 19,20. October 21, 22. £240.00

### SYSTEM DESIGN WITH APL★PLUS

June 24, 25, 26. September 24, 25, 26. £375.00

## CONSULTANCY

We undertake APL★PLUS and VSAPL assignments and will be pleased to discuss your requirements.

For further information on any of the above please call us on 021-359 5096 or see us at APL '86.

---

---

## MERCIA SOFTWARE LIMITED

Aston Science Park, Love Lane, Birmingham B7 4BJ.  
Telephone: 021-359 5096

---

---

## Computer Graphics '85

*Reviewed by Bruce Hollamby & Katie Williamson*

### Introduction

Way back in October we were both able to find time to attend the 1985 Conference for Business & Information Graphics. This was of potential interest to us as we use graphics to develop decision support systems at the Head Office of the Imperial Group.

The following is an account of our 2 days (1 day each!) at the Wembley Conference Centre. The conference had an excellent chairman – Alan Paller. The presentations were of varying interest and relevance and so the account is from a personal viewpoint.

### International Status Report on Business Graphics

*by Alan Paller, President, AUI Data Graphics/ISSCO, USA*

The speaker summarised the current state of affairs both from a software and a hardware point of view. Looking towards the future, we can expect a machine that will provide personal computing power and high resolution graphics, combined with the capability of acting as a terminal into a mainframe computer.

At the moment, there are 3 main functions for PC software:

1. Immediate analysis (e.g. Lotus)
2. Presentation (e.g. Lotus)
3. Drawings for memos (e.g. Macprint).

Surprisingly enough, the third function (i.e. integrated text and graphics) is the growth area. It was predicted that this will become the typewriter of the future.

However, PCs are no good for million dollar applications i.e. the kind that make computers and computer graphics competitive and strategic tools. An overview was given of 12 very high payoff applications which were chosen on the basis that their contribution can be measured in millions of dollars and that they have stood the test of time.

### Charting a Middle Course for Information Graphics

*by Stephen Silver, Consultant, Touche Ross*

This was an interesting talk but was geared mainly towards the 'graphic designers' of the audience. "In the development of computer graphics the information graphic designer has been ignored. The computer needs to become an everyday tool to assist him in solving design problems without restricting his creativity or style."

It was considered how the information graphic designer works. There are 5 stages in the design process:

1. Accept brief
2. Visualisation
3. Rough Sketch
4. Draft
5. Final production

The computer can really help in stages 3 and 5. The facilities required by designers can be assessed by Ease of Use, Looks, Standard Routines, Reliability, Output, Growth and Costs.

The designers have 2 roles to play:

1. Use the computer graphics products and by their demand guide the computer industry in the right direction.
2. Help the computer industry to develop products for use in the business market i.e. packages which incorporate design principles.

Until the design community is prepared to take on these roles and the computer industry recognises that the roles are necessary, computer graphics will never realise its potential for business information graphics.

In summary:

"Designers design – computers compute"



## Graphics in the Boardroom

*by David Preedy, Development Manager, Metapraxix*

The subject was of great personal interest, having recently been involved in implementing a Boardroom graphics system for Imperial.

“Despite the advances of the electronic revolution, computers have still to make a direct impact on the majority of corporate Boardrooms. Considerable research has been carried out into the software and hardware requirements of Board-level systems and a new type of software, called a corporate control system, is now emerging which addresses the specific problems of Board directors.”

The talk covered a series of key questions about the introduction and use of computer graphics in the Boardrooms of large organisations. Both hardware and software issues were considered.

There are four ways in which a Board director is likely to use numeric information:

1. Browsing
2. Analysis
3. Reference
4. Presentation

A large organisation is highly complex. From a director's point of view the top 3 levels of the organisation are the most important i.e. Group, Divisions and Operating Companies. It may also be necessary to go to lower levels to research the nature of specific problems. Accessing all the key information can only be achieved in an acceptable timescale by using a computer-based system. It is accepted that people find graphics help them to identify key trends and factors faster than text or tables of numbers.

The basic hardware requirements are a high-resolution colour graphics screen and a facility for top-quality paper copies. The projection system in the boardroom should be designed so that the light level does not have to be reduced – effective meetings cannot be held in the dark – and it should also handle the traditional media e.g. 35mm.

Several UK companies have already implemented computer graphics in the Boardroom. Such systems should lead to:

1. Improved understanding
2. Better decisions
3. Quicker diagnosis
4. Better meetings.

### **Graphics for Large Volume Data**

*by Janusz Adamson, Senior Consultant, Scicon*

The paper presented covered work that is being done for the research programmes of the Royal Armament Research and Development Establishment. Here, they have the problem of effectively processing and absorbing large volumes of data – calculations may take the best part of a day. The answer has been to introduce a graphics package to monitor calculations as they are taking place. The colour video recordings have played an important role in understanding many aspects of explosive technology.

### **An Integrated Approach to Technical Publishing**

*by Nigel Nobes, European Product Manager, Intergraph*

The speaker would have been of interest to those involved with CAD/CAM systems. He described (very well) the system Intergraph has developed to solve the integrated solution problem i.e. merging text and graphics

“Technical documentation has been estimated to be growing at a rate of almost 500,000 pages a minute.”

The system is designed for capturing, editing and composing text, and creating technical illustrations, merging both and producing high and low resolution output.

### **A New Image for Corporate Publishing**

*by John Marsh, Marketing Director, Miles 33*

“Corporate publishing has become a focus for intense interest within companies. This is a recognition not of the direct costs incurred but of the losses associated with poor corporate documents.”

Examples of corporate publishing include Sales Brochures, Financial Reports, Technical documentation and Presentation Material. The requirements of corporate publishers are quite distinct from those of commercial publishers:

1. Low Volume
2. Continuous Revision
3. High Graphics Content
4. Text on Word processors
5. Multi-authoring

The speaker predicted that in the future companies such as IBM, Kodak and Rank Xerox will dominate the market in corporate publishing. The individual products are available now. The current need is the integration of these units into a total corporate publishing system to meet the needs of the market cost-effectively.

As an aside, the speaker was helped by an excellent presentation system set up by Showcase. The system was run on an Apple and sent out to a TV screen. It really looked good!

### **Improved image using visual information systems**

*Paul Barrett, ISSCO*

This talk appears to have been largely a selling forum for the ISSCO Tell-a-Graf software package. The major software attributes were described with examples of:

1. Good user interfaces
2. Comprehensive capabilities and flexibility
3. Access to data
4. Publication quality output e.g. colour, font, line thickness
5. Layout intelligence (to give compatibility between devices)
6. Machine and device independence.

Cost in the region of £5,000 – £130,000! . . . but the product was impressive.

### **Enhancing point of display with graphics**

*by Graham Young, Headline Advertising and Design*

This talk was launched by a reasonably lengthy video produced by Positive video which demonstrated the power of the Quantel Paintbox system. This neatly led into an account of how point of display and even product packaging could be “simulated” without the need to produce dummy packaging/product. The medium allows for changes in design due to client requirements without major cost penalties being incurred. The tremendous scope for use of this technique by the advertising/design agencies was quite apparent.

### Designing for the corporate image – a new launch for graphics

by R. Markell, *English Markell Pockett*  
and P. Bernais-Price, *Spectrum Communications*

A two man slide show presenting the work which had been done in assisting large corporate entities to develop a corporate image, combined with a discussion on the need for a clearer understanding of what makes for better communication. The examples shown included those for ICL, G.E. Plastics, National Westminster Bank – Markell had also had a hand in work on BT, BMid, Renault and Channel 4 (of tumbling blocks fame).

The area of corporate image making is a new market for the new generation of computer animation techniques where music as well as action is important. The impact of the use of music was dramatically stressed by showing the Natwest Logo video without the soundtrack.

The other message which seems to be important was that for a successful corporate image a single corporate identity was required throughout company activities (e.g. Blue Circle good; BBC too varied throughout the range).

### Making the right choice

by R. Thompson, *Eidographics*

According to the speaker, 37% of business graphics are produced by computer. The slides market in the US is \$4bn; £350-£375m in the UK. From which one can deduce that either (a) a lot of slides are being produced and/or (b) the cost per slide is pretty high.

Eidographics offers a “high resolution graphics” slide service bureau at £6.50 per 35mm slide – the cost of having an ‘in-house’ facility of sufficient quality is usually prohibitive unless you produce a vast number of slides within the organisation.

In discussing the 5 software ‘front runners’, he mentioned Palette (c£1395), Starburst (c£55000); Autographics/Genographics; Dicomed (high quality service compatible with Imaginator at £180K or Presenter PC at £995 from Eidographics). The first four did not rate highly on the speaker’s score sheet unless in the case of Starburst you were a designer.

### **Training the user to be computer graphics literate**

*by P. Brown, Middlesex Polytechnic*

The speaker dealt with the subject title in the first 5 minutes of his talk not least with the startling statement that only about 30 out of 1100 students attending Computer Graphics courses came out competent (Graphic designers are born not trained?). The remainder of the talk by this "guru" of graphic design (as became apparent from the questions after the talk) was taken up with dealing with some of the more esoteric aspects of graphic design – undoubtedly good stuff for the graphic designer but of limited interest to all others, one suspects.

### **What price business graphics?**

*by Technical Director, Micro Design*

The only talk of the day without graphics, but this was not a setback to this sometime impassioned talk on micro-based graphics. A brief history of the progress to date from monochrome to colour 32-bit micros lead up to the viewpoints concerning the problems faced today when using colour graphics on micros:

1. The mainframe/mini approach has been foisted onto micros without regard
2. This generally leads to poor performance of the software
3. Conflict of standards e.g. textual prompts are OK on minis, but icons are preferable on micros.
4. Conflict between closed and open software i.e. tailor made applications of multi-attribute systems. It is unlikely that both will score high marks on all aspects so the inevitable tradeoff (or so it seems in micros) must be made.

The future may be rosier with improvement in memory/storage but economic arguments will need to be balanced in terms of what price performance and output quality.

The limited availability of 3D graphics on 16-bit micros was touched upon, though, possibly only to pre-announce a Micro Design software package which will address the problem.

### Ten trends shaping the future

*by A. Paller, AUI Data Graphics/ISSCO*

The best was saved till last, Alan Paller delivered his talk at breakneck pace on his perceived ten trends for computer graphics. Highlights of which were:

- Future software will aim to provide solutions rather than tools
- PCs should be networked so as to open up the availability of shared central resources (which would otherwise be prohibitively expensive).
- The linkage of databases to graphic packages so that by integration the best attributes of each could be achieved.
- A warning on the dilemma of standards (GKS versus CORE) and the need for device drivers.
- Intelligent interactivity from the graphics software e.g. the facility for slide content to alter according to the device output rather than re-creation.
- The problems with chart books of standard charts, with the useful suggestion of posters rather than books.
- The introduction of a standard facility which allowed for varying background colours within the slide (e.g. rather than the dreaded single blue background).

### Summary

The following points sum up the overall view of the conference:

1. The organisers of the show don't seem to have acknowledged that there were two schools interested in colour graphics namely 'hands on' users and graphic designers. This leads to polarity on the talks given and unsatisfactory appeal to both, one suspects. Surely splitting the talks into two streams is not beyond the bounds of their ingenuity.
2. Many of the talks seemed to fall into two categories either esoteric, academic dissertations or thinly veiled promotions. Surely the latter should have been dealt with at the accompanying side shows.
3. The use of video in presentations was excellent, less successful was the use of 35 mm slides – lamentable given that this was a colour graphics show!
4. Being in favour of the promotion of the use of graphics, one was dismayed that the overall impression of the show was that of a 'closed shop' – anyone interested in finding out about graphics and their usage would have struggled to have got anything substantial from the conference. Hardly an encouraging thought.

## Expert Systems '85

*reviewed by Peter S. Davies*

"Expert Systems 85" was the somewhat unoriginal name given to the 5th Technical Conference of the BCS Expert Systems Specialist Group. Frankly, it wasn't much different from any of the previous conferences except that it was bigger – over 900 eager beavers attended.

Assembled gurus included all the old familiar figures, though – Alex d'Agapayeff late of CAP and now of Expertech, Martin Merry late of GEC and now of Hewlett-Packard, Aaron Sloman of Sussex, Father Aloysius Hacker of Hacker Enterprises, Austin Tate of almost everywhere, and so on down the list of experts about expert systems.

As well as the 900 conference attenders, many of the day trippers who set off to visit the accompanying exhibition, also at the University of Warwick, actually arrived. (Coventry is 10 miles from Warwick, but "University of Coventry" doesn't sound as good as "University of Warwick"). The throng having been thus swelled by survivors of the cross-country trip, the exhibition was pretty sticky for much of the time, and difficult to gather information (though I did witness the theft of a complete knowledge base by a daring young man. He had enquired about the quality of documentation, settled down to examine a proffered listing and then, when the salesman's back was turned, coolly walked off with it!). Portability of extremely valuable knowledge is one of the hazards of expert systems.

Everything was for sale from a £20 demonstration expert system, through videotapes, software, consultancy and robots to hardware systems costing £250,000 or so (every desk should have one??)

The conference proper followed the modern fashion for two or more parallel sessions. As usual this resulted in the simultaneous coverage of interesting topics in two different places, followed by the simultaneous availability of two equally depressing longueurs. Each paper in a session was supposed to finish at the same time as the corresponding one in its parallel 'stream' but they always over-ran or under-ran by several minutes, so that the closing stages of one, or the essential explanatory moments of another, lecture were constantly interrupted by multiple noisy, out-of-breath arrivals and departures. Corridors and stairs between the theatres would be filled by surging crowds, moving in opposite directions, having missed the dénouement of one lecture in their haste to be able to understand what the next man was talking about. And some unfortunates haunted the passageways, desperately shuttling between alternate attractions like the donkey between two piles of straw, unable to decide which to enjoy.

Enjoyment is perhaps the wrong word to use of papers such as “Symbolic uncertain inference: a study of possible modalities”, though a certain innocent amusement arose from time to time – as when one chairman, by giving an eloquent introduction to a talk on the application of casual knowledge to real-time process control, revealed that he had failed to do the most basic homework of reading the paper (which was actually about causal knowledge!). This latter paper, castigated by some as indecently non-academic, actually showed a serious application of deeper, causal, knowledge as being necessary to the proper handling of information overload situations such as North Sea oilfields (or, in retrospect, the Three Mile Island disaster, where 500 warning lights went off or on in the first minute alone!).

Whilst many papers should never have been let out of their ivory towers, the most pragmatically useful were not, as might have been expected, on techniques and applications, but tended to be about the user interface – for example users quite often don’t just need a diagnosis but need remedies (or even explanations as to why what the user did, as a remedy, actually worked). In an extremely entertaining lecture on an unusually readable paper, a team from Bath reported on expert systems as a medium for knowledge transfer (rather like d’Agapeyeff’s “sharing know-how”) and how human knowledge is a lot more than just the information and heuristics they are supposed to be by traditional expert systems and knowledge elicitation processes. They showed convincingly that humans use tacit knowledge without ever realising it, and that skilled accomplishments may need analysing in order to make sense of the formal knowledge currently being collected.

One of the examples of tacit knowledge they used was this step from a recipe: “Beat the egg whites until stiff and then fold in” – just think about explaining that to a computer.... which bit of an egg is white? (not the egg white, which is transparent!), which bit is an egg white? how do you get it out? what does “beat” mean, or “stiff” or “fold in”? Although this is an extreme example of human-speech being difficult to formalise, I’m currently finding out the hard way that there’s an awful lot of tacit knowledge about everyday information sources which it’s difficult to disentangle and formalise, and so am realising that Expert Systems ’85 wasn’t quite as academic as it seemed at the time. Overall rating for people with a slight interest in expert systems? . . . worth a detour – perhaps not worth a special trip.



---

## GENERAL ARTICLES

This section of VECTOR is oriented towards readers who may neither know APL, nor be interested in learning it. However we hope you are curious about how, under the right conditions, such impressive results can emerge so quickly from APL programmers.

Both our general articles this issue concentrate on impressive APL applications. Christine Brewster (of Whitbread) describes how they used APL to develop their Sales Analysis system – producing the reports other languages couldn't handle.

Steve Lyus describes the reasons behind Imperial Group's choice of APL for financial modelling, culminating in the development of a complex 5-year planning system.

---

# MetaPack... Everything you need for APL on the PC.

**MetaPack** is the utilities package that has been designed especially for users of APL★PLUS/PC.

**MetaPack** comprises a number of programs, each designed to help you build better systems in less time.

**MetaPack** will increase your productivity as an APL systems builder, and enhance the function and appearance of your systems.

The **MetaPack** programs are:

- MetaScreen** — the popular AP124 based full-screen management system. The speed-critical modules have been written in assembler, making **MetaScreen** ideal for running mainframe applications on the PC.
- NI** — for full-screen input and editing of APL numeric variables.
- Browse** — an attractive and sophisticated method of viewing reports or text files on the screen by horizontal or vertical scrolling.
- ToolBox** — an invaluable collection of APL utilities, including table-lookup and vector-matrix conversion functions written in assembler.
- MetaWS** — the program filing system for easy management and documentation of large APL systems.

**MetaPack** programs have been written in a combination of APL, C and Assembler. This means that your systems can benefit from **MetaPack's** advanced features yet still run with realistic execution times.

The programs may be used either as stand-alone utilities or as an integrated part of your systems.

Whether you're a professional APL systems builder or an end user who has discovered the power of APL, you will find **MetaPack** an indispensable aid to building better systems.

**MetaPack** — £495    **MetaScreen** (available separately) — £245  
For IBM PC XT AT, COMPAQ & Compatibles

APL★PLUS/PC is a registered trademark of STSC, Inc.

# MTS

## MetaTechnics Systems Ltd

Unit 216V, 62 Tritton Road, London SE21 8DE, 01-670 7959

The Systems Builders

## Case Study

*edited by Adrian Smith*

### Acknowledgements

Many thanks are due to Christine Brewster of the Group Management Services Department of Whitbread & Company. I had previously heard of 'The Whitbread System' by repute, but to see it in action was a wholly different thing. From the perspective of my hitherto sheltered existence the prospect of using APL to gulp massive amounts of data in megabyte workspaces, indeed to beat COBOL comprehensively at its own game, was highly attractive, and impossibly remote.

Well, I know better now; it really can be done, and done well. In fact I would have to say that the way APL is being used at Whitbreads was one of the most impressive syntheses of old and new that I have ever seen. I hope the flavour of this comes out clearly in Christine's notes; I'm sure she will forgive the occasional editorial interjection from time to time!

Incidentally, readers may be interested to know that the excellent pub chosen for lunch did not in fact offer for sale her company's product. Well, I have (on rare occasions) been seen eating Mars Bars, so perhaps she may be forgiven for this momentary lapse!!

## APL as a System Development Language

*by Christine Brewster (Whitbread & Co)*

### Introduction

APL had been in use at Whitbread for around 5 years, covering all the normally accepted areas of personal computing. However in the system about to be described it was used much more as a 'conventional' systems development tool. As it happened APL turned out to be very much more effective than conventional data-processing methods; it was quicker to implement, gave a better end-product, and was cheaper to run.

### Background

The brewing and wholesale interests of Whitbread are today complemented by two other high-profile areas: retailing and Wines & Spirits. It was from the National Sales division (responsible for 'take-home' sales) that the pressure for a responsive sales analysis system first came. Typically, take-home sales involve cans and bottles (of both beer and wine) through shops, off-licences, wholesalers and cash-and-carry outlets.

The system was deemed URGENT because it was necessary to have both recent and accurate information on the business, in order to know the effect of sales promotions, advertising campaigns and the status of new products in a changing and very competitive market.

It was known to be fairly COMPLEX because it needed to be able to report on whichever areas of the market were of interest at any time. It was difficult to predict which these areas would be.

It was going to be LARGE in terms of data quantities – 5 to 12 megabytes of sales history data being input each week, building up over two years to a database of about a gigabyte of data. The major data keys are outlet, product and depot and there are some 720 million possible key combinations.

It was also going to be LARGE in terms of print volumes – the month end reports were expected to produce between 10,000 and 40,000 pages of print – between 5 and 20 boxes of paper.

It was also going to pose an unusual processing problem in that the users felt they would need up to 300 reports a month, each report covering between 10 and 300 megabytes of data.

There was a huge area of UNCERTAINTY:

- ★ How fast would the database grow? This growth is determined by how many different products each customer orders. The average would increase at Christmas and other peak sales periods, but by how much?
- ★ The effect of promotions – products sold on promotion take on new characteristics and need to be reported on separately. A product sold on promotion effectively becomes a new product and adds to the size of the database. It was uncertain how many transactions were going to be under promotion and what their overall effect would be.
- ★ Reports – the number and content of the reports requested each day was unpredictable and depended on the marketing interest at the time.
- ★ Interpretation of data – it was suspected that there were items of the source data that were ambiguous. There was also additional data such as budgets and targets whose source, quantity and interpretation were unknown.
- ★ Organisation of reports – 11 different types of reports had been defined. It was uncertain how many different subtotalling systems were needed, or how complex these were. This subtotalling might reflect the structure of the customer's company, as well as of NSD's different sales forces and management reporting structures.

In summary there were a lot of problems!

### **Where to Start?**

It was of course natural to use the conventional and proven 'Whitbread Approach' to the problem, especially because of the size of the system and its importance to the business. In due course a proposal was developed to meet the requirement; it involved holding a VSAM database of all outlet/product/depot occurrences with all subtotals necessary for reports. This made reports and on-line enquiries cheap, but the weekly update would be expensive.

The subtotalling systems were to be represented by structured codes, the management of these to be left to the user. Any changes to the subtotalling system would necessitate a recalculation of the database.

The method was found to be unacceptable because:

- ★ The run times of the jobs which updated the database were too long. It was estimated that the weekly run would be in excess of 40 hours, due to the huge number of disk I/Os involved.

- ★ The database would be too big. This created problems of cost, reruns and security.
- ★ There was too much uncertainty about the subtotalling structures. Changes in customer companies might mean amendments to COBOL programs.
- ★ As there were so many uncertainties it was obviously very risky to commit programming resources to producing a system which was not guaranteed to meet the users' requirements.
- ★ The timescales were too long. The users would not start getting results from the system for at least three months. (?! Ed)

Essentially the problem was a classic chicken-and-egg situation. We needed information to design the system, but that information could only be provided by the system we did not yet have.

### **The Search for a Stop-gap**

Because our conventional solution was unacceptable, we looked for a stop-gap measure based on the Business Systems Design and requirements reports already produced.

The minimum criteria were:

- Acceptable run times
- Flexible reporting
- Quick, phased implementation
- Low profile

We looked for a stop-gap measure in order to:

- ★ provide the users with as much information as possible with which to run their business as soon as possible.
- ★ generate information which would provide a clearer understanding of what was needed so that a 'proper' system could be specified and the necessary processing capability could be afforded.

Firstly, an external bureau was approached. They proposed that we use a relational database to set up and update the data, a query language to extract the subsets, and a general reporting language to produce the final reports (with infinite variations).

This approach was not feasible because:

- ★ The size of the problem was too great;
- ★ The complexity of the system could not be handled elegantly by general languages;

- ★ The CPU and I/O time estimates were again enormous.

APL was then adopted as a stop-gap measure. The APL solution involved storing all the data *at the lowest level only*, processing it offline using SORT (with SORT exits to reduce the number of I/Os) and doing all the calculations at report time.

Using the APL approach we could tackle the problem by making only a small number of basic assumptions, and build up towards a complete solution. Since all the reports require data for one month, with year to date, compared to data for the same month last year, we decided to use a database that was segmented by month.

The APL solution was accepted because:

- ★ It was CHEAP. The implementation was estimated to require one man-year of effort – a team of four for 3 months. We did not need to buy extra software.
- ★ It was QUICK to implement. Any problems with the system would appear soon, before too much money had been spent. As all development was done using live data, the development process would produce information useful to the business.
- ★ It was LOW PROFILE. The small team would not use programming or other resources required by other systems, and would not be highly visible. The team was made up of three APL contractors provided by a specialist APL bureau, in addition to our own APL specialist.
- ★ It was almost CERTAIN to work. The team had already produced results for the users while in the initial feasibility stages.
- ★ The system was seen as an EXPERIMENTAL stop-gap measure and was not intended as the final solution. If the user was not satisfied he knew that he would have another chance later.

### Experiences of Implementation

Our first venture into this type of development was more successful than we could have imagined. This was because:

- ★ We used PROTOTYPING – we effectively wrote the system specification in APL. It could convey infinitely more information than a specification in words; the user could log on to the system and try it out. It also avoided all the ambiguities of a written specification.
- ★ Users quickly became involved in the development process, and could provide valuable FEEDBACK.

- ★ Changes to the system could be made while sitting at the terminal with the user. Problems could be sorted out IN FLIGHT. Writing a system in APL is like driving a car with a mechanic sitting in the back seat. If the car breaks down the mechanic fixes it and you continue your journey, without having to have your program towed back to the garage.
- ★ The system was designed using an EVOLUTIONARY process. It was used to generate the clearer understanding of the problem that was needed to refine the system design. The initial system was the egg which hatched into information which laid the next version, in turn hatching into more information and so on.
- ★ The system design was GENERALISED. A small number of basic assumptions were made to start the process. These were acknowledged as guesses and refined as necessary with hindsight. There were as few limits hard-coded into the system as possible.
- ★ The system designed was OPEN-ENDED. No time was wasted on areas of uncertainty – these were omitted until we had a clearer understanding. We split the large area of uncertainty into small areas that could be tackled separately.
- ★ The system was developed by a SMALL TEAM, comprising four APLers and a user representative. This meant that there was GOOD COMMUNICATION: within the team; with the users; with other technical experts.
- ★ Each team member could see (and understand) the whole system. This eliminated problems with consistency and ensured that components interfaced. It also eliminated re-inventing the wheel, and enabled problems to be identified SOON.
- ★ Our team had LOW OVERHEADS. Because our APLers had a high concentration of skills, each team member combining abilities in: analysis (business); systems design; APL programming; technical (JCL etc.) – there was no need for the systems analyst to explain the user's requirements to the programmer because the analyst was the same person as the programmer.

### The Solution

The APL solution introduced the idea of trees. These are hierarchies which are subject to strict rules:

- ★ Each tree can be based on either outlets, products or depots.
- ★ All products must appear in each product tree.
- ★ Each product can appear only once in a tree.
- ★ The number of levels in a tree is defined by the user when the tree is set up.
- ★ The products are all at the bottom level of the tree.
- ★ Each level in a tree has a name.
- ★ Each node in a tree has a code and a description. The nodes are sorted by code.



The trees are used throughout the Sales Analysis system to specify the level of subtotalling, data selection and ordering in reports. Trees can even be stacked on top of one another! They are maintained by the user with a full-screen editing system, where . . .

1. The user gets a screen of all trees and can select the one he wants to edit.
2. The next screen is a list of all nodes at the top level of the tree, and he can select one by moving the cursor and pressing a PF key.
3. He is then shown a screen of nodes at the next level of the branch he has chosen and can select one for further investigation.
4. The process continues down to the bottom of the tree where the screen shown has the products belonging to that branch.

The user can insert or delete nodes, change their codes or descriptions, or GET branches from the same or another tree. He can also move products from one branch of the tree to another. Alternatively, he can edit a special tree called PRODUCTS and see where one particular product fits into all the product trees.

The trees are also a fundamental part of report requests. These are again generated from a full-screen system, based on menus, which allows the user to select from data known to the system:

- ★ Which report from the list of standard types.
- ★ Which trees to use to structure the report. Where the report requires a product tree a menu of all available product trees is shown.
- ★ How far down the tree to do subtotalling. The user can select from a menu of level names in the tree.
- ★ At which level to do data selection. The user can select from the levels down to the subtotalling level.
- ★ Which branches of the tree to select data. The menu shows all nodes at the level chosen.
- ★ Which level of the tree to do page-throws. The user can choose from the levels covered by the report.
- ★ Which month's data. A menu shows all the months that make up the database.

Once a report has been defined, it is 'posted' away for offline processing in overnight batch. This is where things really start to happen.

### **The Offline Reports System**

Each report in the stream of requests generates a sequence of job steps, such that:

- ★ Firstly, an APL step reads data from tape, tags on the necessary sort keys (tree information), and writes the result to a temporary dataset.
- ★ A standard JCL SORT is then used to sort the data.
- ★ A final APL step reads the sorted data and produces the report.

JCL is written from APL which examines all the pending requests and decides on order of reports, which tapes to use, which data fields to extract from the database, how many and which sort keys to use, if a sort is necessary, and the logical record length of data to be sent to the intermediate dataset. The JCL generator also writes a schedule sheet for use by operations staff, and a profile sheet listing the expected CPU time and the tapes to be ready for each job step.

The appendix shows part of a typical job stream. The DTL step puts sort keys on to the data. Each SRT step sorts the data for reports. Each REP step produces a report. The number of steps in each job is determined by the number of reports that have been requested. The users often request up to 35 reports at a time.

*(Ed. This all sounds very simple; to make it work in practice, with very large data volumes, is altogether another matter. Some of the 'tricks of the trade' used in the Whitbread system are detailed below.)*

The structure of the database is clearly all-important:

- ★ The sales history data is held throughout the Sales Analysis system in APL format.

APL sees the data as matrices of integers. The number of columns (the logical record length) is determined by the system and depends on the number of data fields and the number of sort keys that are in use at the time. The number of rows is calculated so that each matrix will fit into a half track of a 3380 disk (23476 bytes).

SORT sees the data as single record blocks which are decoded by an E15 exit. The E15 uses the APL header and trailer information to determine the actual length of the logical records and to pass these separately to the sort process. The E35 exit reverses the process on the sorted records. This method reduces the number of I/Os performed by the system to a minimum, while preserving the APL format of the data.

- ★ Sales history data is held on tape – each tape holds a month's figures plus year to date. The database for a year is thus made up of 12 tapes. The tape files grow as the year progresses because of the accumulation of year to date figures.

Each report requires the tape for the relevant month this year and possibly the tape for the same month last year.

Some of the software enhancements needed to optimise the running times were:

- ★ SORT uses E15 and E35 exits in Assembler to unblock and reblock data which is in internal APL format.

- ★ APLPRINT is used to output reports a page at a time.
- ★ Table lookups were speeded up using a fast dyadic iota routine (written in Assembler) which takes advantage of the fact that the table and the data are both in ascending numerical order. This is interfaced using 'CALL/AP' from Interprocess Systems.

*(Ed. So much for all the Whizzbangs . . . how well does all this really perform? The answer is interesting; in APL terms it still manages to crunch massive amounts of CPU, but of course this is mostly off-line and overnight, and therefore 'free'. In terms of the same system written in conventional batch terms, the run-times look much less frightening. Strange how one's perspective changes; I used to think very little of FORTRAN jobs which ran for 1,200 sec of CPU.)*

### System Performance

The system produces approximately 80 standard and 150 ad hoc reports per month. It prints on average 15,000 pages on paper per month (mostly multi-part paper) used by over 100 people. Some 20,000 pages are also sent monthly to microfiche (for archiving).

The average CPU per report is about 6 mins on a 3083J; an estimate because the system has never run for a whole year between optimisations. Typically, 110 megabytes of data are processed for each report; to manage this efficiently, large amounts of memory are essential:

- ★ A 6.2 megabyte address space is needed for batch steps. This enables APL to load large quantities of data into memory to do calculations.
- ★ A 7.5 megabyte workspace size for on-line sessions. This is necessary to hold the APL variables which make up the trees.

How much APL code is involved? Measured simply in lines of code:

- ★ 40,000 lines (in 10 workspaces);
- ★ Approximately 2 megabytes of code (compared to 7,500 lines in 0.4 megabytes for ADRS; Sales Analysis 5 to 6 times more complex than ADRS).

Estimates for re-writing current system in COBOL – no volunteers.

### What do the Users Think of it All?

We know that the system is successful because the users continue to identify new areas that the system could cover. These new requirements emerge as a result of information that the system provides, as well as from the users' increasing experience. These requests are easy to implement because our team has expertise in the whole system.

- ★ Data extract – Sales Analysis can produce data extracts from the database in machine readable form. These can be used in Personal Computing (usually APLDI) for the users to do their own analysis and manipulation.
- ★ Drop size analysis – a parallel database with data keyed by delivery date and delivery number to give an analysis of the size of deliveries. The reports produced by this development were difficult to verify because there was no existing system for reconciliation of the results.
- ★ Report generator – this allows users to change the format of reports by specifying column widths, headings, which data in which columns, ad hoc calculations, and so on. These changes previously had to be made by the development team.
- ★ Distribution analysis – a type of report not previously planned for, measuring market penetration of products.
- ★ Budgets and targets – the users can use ADRS to set up budget and target figures and to report on them. Sales Analysis now has a facility to access this data.
- ★ Tree prints – since the idea of trees has become so popular with the users, they asked for a facility to print them out in graphic form. Every manager who is anyone in NSD has a tree print on his office wall.

### Conclusion

It is our opinion that the evolutionary development process was the most sensible approach because we were faced with so many uncertainties over a system that was undoubtedly going to handle large quantities of data. APL provided us with the best answer for prototyping. Also APL was suitable because of the power of the language to handle large quantities of data in a way that was completely user-driven.

The methodology used in APL enabled a system to be designed to suit the user's requirements even though he couldn't specify those requirements at the time.

Among other things, the undoubted success of the system is demonstrated by:

- ★ Its continuing use. In spite of the implementation of data extract, which allows users to do their own reports, usage is higher than expected. The users anticipated a maximum of five outlet trees and five product trees. They now have 14 outlet trees, 12 product trees and three depot trees. There is a thriving user group.
- ★ The ease with which it is being applied to new areas, and the requirement to do so. Development has been continuous since the beginning of the APL project.

The final accolade is surely this; the scrapping of the original plan to produce a permanent solution to replace the stop-gap APL experiment.

## Appendix. Typical job stream

```

/*****
/* GO INTO APL AND DO STEP: DTL5
/*****
/*
//DTL5      EXEC PGM=IJKFT01,REGION=6200k,DYNAMNBR=5
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
PROF PREFIX(NSSA)
CALL 'SYS1.TFAX.LOADLIB(TFAXPGMS)' 'SYSTS6G'
VSAPLNEW SHR(128K) FREE(512K) CODE(1) SMAPL(OFF) AP(CALLB02)
/*
//CONTINUE DD DUMMY
//CLIB     DD DSN=PNSSA.LOAD.LIB,DISP=SHR
//APLPRINT DD SYSOUT=*,CHARS=APL,DCB=LRECL=133
//APLIN    DD *
)LOAD NEWREP
DETAILS
/*
//THISYR   DD DSN=NSSA.PERIOD.D1985070,DISP=OLD
//LASTYR   DD DSN=NSSA.PERIOD.D1984070,DISP=OLD,UNIT=AF=THISYR
//FULLYR   DD DSN=NSSA.DUMMY.EMPTY,DISP=SHR
//TEMP     DD DSN=NSSA.REPTEMP,DISP=OLD
//FIRST    DD DSN=NSSA.CONTROL.FILE,DISP=OLD
//BUD      DD DSN=NSSA.BUDGET.INTERIMU,DISP=SHR
//LAY      DD DSN=NSSA.BUDGET.LAYOUT,DISP=SHR
//REQS     DD DSN=NSSA.REPORT.REQ(N5),DISP=SHR
/*
/*****
/*          SORT FOR REPORT: SRT5
/*****
/*
//SRT5     EXEC PGM=SORT,REGION=6200K,PARM='SIZE=6000K,DIAG'
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SORTLIB  DD DISP=SHR,DSN=SYS1.SORTLIB
//USERLIB  DD DISP=SHR,DSN=PNSSA.LOAD.LIB
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(75,20))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(75,20))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(75,20))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(75,20))
//CONTROL  DD DSN=NSSA.CONTROL.FILE,DISP=SHR
//SORTIN   DD DSN=NSSA.REPTEMP,DISP=OLD
//SORTOUT  DD DSN=NSSA.REPTEMP,DISP=OLD
//SYSIN    DD *
          SORT FIELDS=(17,4,BI,A,21,4,BI,A,25,8,BI,A,5,4,BI,A,1,4,BI,A)
          RECORD TYPE=F,LENGTH=(23476,48,23476)
          MODS E15(NSDE15,5000,USERLIB,N),E35=(NSDE35,40000,USERLIB,N)
/*

```

```

/**
/*****
/** GO INTO APL AND DO STEP: REP5
/*****
/**
//REPS      EXEC PGM=IJKEFT01,REGION=6200K,DYNAMNBR=5
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROF PREFIX(NSSA)
CALL 'SYS1.TFAX.LOADLIB(TFAXPGM5)' 'SYSTEM66'
VSAPLNEW SHR(128k) FREE(512K) CODE(1) SMAPL(OFF) AP(CALLB02)
/*
//CONTINUE DD DUMMY
//CLIB     DD DSN=PNSSA.LOAD.LIB,DISP=SHR
//APLPRINT DD SYSOUT=*,CHARS=APL,DCB=LRECL=133
//APLIN    DD *
)LOAD NEWREP
REPORT
/*
//REF      DD DSN=NSSA.REPTMF,DISP=OLD
//HSAPLOUT DD SYSOUT=(A,,LUN),FREE=CLOSE,
//          DCB=(RECFM=FBA,LRECL=133,BLT SIZE=188B6)
//CNCL     DD DSN=NSSA.REPORT.CNCL,DISP=OLD
//REQS     DD DSN=NSSA.REPORT.REQ(NS),DISP=SHR
//PDIR     DD DSN=NSSA.EXTRACT.DIR,DISP=SHR
/**

```

## Imperial Group 5 Year Planning Model

*by Steve Lyus*

### Introduction

Imperial as a group has been using computers for financial modelling for at least 15 years, under various guises. For the last 4 years the APL language has been very successfully used at the centre, at first on a single-user 8-bit micro, and more recently on a multi-user 16-bit Supermicro. This paper gives a non-technical review of the history of the development, together with an overview of the current state of the model.

### The Historical Perspective

Prior to 1981, Imperial as a group had had considerable success with financial modelling using the various versions of Prosper on ICL mainframes. However this had always been at a level in the organisation well below the consolidated Group accounts.

Around 1980 the Operations Research section became convinced that APL was the preferred modelling language (supplanting Fortran) for ad-hoc projects. The advent of APL on a cheap, portable 8-bit micro encouraged us to review our options for financial modelling. In March 1981 an internal report was produced, comparing VDU Prosper, FCS (on two bureaux), EIS (on Boeing Computer Services) and TIS-APL on a Superbrain. Several typical financial modelling benchmarks were produced using each of these languages. We wrote the Prosper and APL models and the bureaux provided their versions. Our conclusions were that we would continue to use VDU Prosper where this was appropriate (e.g. end-user running the model), but TIS-APL would be used on models where the O.R. section was the user.

Late in 1981, with the advent of a new Chairman, came our chance to demonstrate the use of financial modelling at the Group level, and our choice naturally fell on the Superbrain, as the London end of Group Head Office had no computing facilities. The demonstration went well and we were commissioned to work with the Chief Accountant's Department to produce a 1-year, or Budget, model at the Group consolidated accounts level. This was initially squeezed onto the Superbrain and was very well received. The ability to do more than the 1 or 2 "what-ifs" the manual calculations allowed, opened up a whole new world of possibilities.

The purchase of two multi-user MicroAPL Spectrums for Head Office in July 1982, one for the Chief Executive's Information System (q.v. VECTOR Vol 1, No 2) and the other for the Chief Accountant's Department to do the routine consolidation of the monthly accounts, meant that APL.68000 was now available. The Budget model was very easily transported over and work began almost immediately on a 3-year model.

This meant an order of magnitude change in the complexity, mainly connected with time lags inherent in the calculation and payment of Corporation tax. In January 1983, the model was expanded again to a 5-year one, a much less traumatic change. The complexity was increased however, in April 1983, by the desire to input U.S. data in dollars and have the model deal with the U.S. tax regime, borrowing in the U.S., and exchange rate conversion.

The model had become by then an integral part of the Strategic Planning process, and was at times in use 12 hours a day, and at weekends. The ability to investigate the numerous strategic options being considered, without being constrained by the time it takes to "turn the handle" was exploited to the full.

Since mid-83 the pace of development has slowed considerably, though not the usage. Further enhancements included the ability to acquire and/or dispose of companies. It was extensively used to explore the possibilities available before deciding to dispose of most of the Howard Johnson Company in 1984, and before deciding to approach United Biscuits with a view to a merger late in 1985.

The latest change to the model has effectively removed the limit on the number of years in the planning horizon.

### **Important Characteristics**

#### **1. Evolutionary Development**

The development was always undertaken in close contact with the accountants, in fact most of the important code was written whilst actually sitting at the terminal with the accountant responsible for the model. In this way the modularity available to a writer in APL was used to the full. Functions were often tested in isolation with all possible (and impossible?) combinations of input. This was especially important when trying to model the U.K. tax structure. The ability, in APL, to stop the function wherever necessary, investigate variables, and then re-start was also well used. This was most fruitful in areas where misunderstandings arose over the interpretation of the words used to specify a section of logic.

#### **2. User Driven**

The method used to develop the model led naturally to the accountant involved taking over the running of system. He was totally familiar with what was going on, and indeed the words used in the dialogue are mainly his. Once it was accepted as working, 3 other accountants (including the Chief Accountant and his deputy) started to use it with no obvious difficulty. One of the reasons for the success of the model is that it is now "owned" by that department.



### 3. Top-Level Model

As mentioned above, the aim of the program is to model the Group consolidation process. It works in a bottom-up fashion, data being input at the Divisional level. A diagrammatic representation of the model structure is shown in figure 1.

In terms of the complexity of the calculations, it is very definitely at the detailed, as opposed to the "broad-brush", end of the spectrum. Some parts of the model allow for two modes of input, (e.g. depreciation) either "accurate" calculations that are carried out off-line and the results fed in, or additional data given to a sub-model that produces a good approximation. The trade-off between the volume of input data required to achieve an end result and the ease with which it can be derived from other sources is an important consideration.

### 4. Calculations undertaken

As mentioned above, the UK & US tax structures are modelled to a high degree of accuracy. Another important feature is the treatment of interest on short term borrowings, this is calculated correctly with a simultaneous equation to allow for "interest on interest". We heard recently that a reputable merchant bank did not bother with this in their "corporate model". Lease rental payments (capital & interest) are also calculated, at the given interest rates, for new leases input in the planning period.

### 5. File Structures

Only input data are stored, in the overlay form available in APL.68000, one component of the file for each different scenario. There is also an administrative component, holding the text used to describe each run, the date & time of creation, as well as the name of the creator. In the early days, frequent major changes to the structure meant that no file became too large (100 runs was typical). However, recently this has become a problem, both in terms of an enormous list of model names being presented whenever the "read-file" option was chosen, and, more seriously, the limit on physical storage on the hard disk was reached (6.5 megabytes for a logical disk surface).

The latest change has been to create a meta-file structure (i.e. file of files). This allows much more flexibility, both in terms of the physical location of the files, and also in terms of the actual usage of the model (eg. one file would be for all the runs connected with the proposed United Biscuits merger, another with the 85/86 Budget).

### 6. Security

The data in the various runs (even the titles!) are obviously very sensitive. There is the usual multi-level passworded access to the model. There is, however, an additional level, whereby the creator of an input file has to specify who can access (read only or read/overwrite or not at all) that file. Thus the user, when choosing his input file, only has available those files he is allowed to know about.

## 7. Full-screen data input

From our VDU-Prosper days we were convinced of the utility of this approach, and were amazed, back in 1981 when we conducted our survey of financial planning languages, that the developers of these languages were not taking any interest in this facility that was available now that the VDU had superceded the teletype to become the preferred input device. One of the big plus-points in TIS-APL's favour was that we could design our own data input screens.

The functions currently available on our data input screens include:

- the execution of a specified function at the press of a key (mainly used to re-calculate sub-totals, although it could be used for validation).
- a printer dump of the screen as displayed.

## 8. Data Input Comparison

There have always been a large number of input variables (118 at present). It became apparent, even in the early days of the model, that a facility to print-off a report of the differences between two files would be exceedingly useful. It started life as a "special", but is now one of the options on the main menu. It produces an exception report (i.e. only those tables where there is a difference), using the same format as that used to input the data on the screen.

## 9. Output of Model Results

One of the first utility workspaces we developed was an extremely flexible report-generator. This is used to produce the output. The individual tables are only produced if requested. Some tables are intended for direct inclusion in reports, and are in the same format as that used internally for all financial reporting (e.g. monthly accounts), whilst others are much more detailed 'working papers' that allow the accountants to reconcile the results of several runs.

The following options are available at the output stage (figure 2 is a copy of the main menu presented at this stage) :

- view any table on the VDU screen.
- redirect any table to the printer (high-speed matrix or quality daisy-wheel).
- store a subset of the results into a special database file (see later).
- send most tables to an IBM PC (for 'Importing' into Lotus 1-2-3 for further analyses – see later)

### Current Statistics

1. There are currently available some 490 different runs of the model, 174 in the latest version of the model.
2. Each run of the model requires about 35k of storage.
3. There are 118 input variables, ranging from scalars to three-dimensional arrays.
4. The data are entered using 49 different input screens.
5. The model takes about 1 minute to calculate the 5 years of a plan, in an empty machine. It is a multi-user system, so it is often longer than this. It will run in a 500k workspace.
6. There are 25 different result tables available. A print of the whole set runs to 36 pages.

### Links to other systems

#### 1. Database Link

We make considerable use of our own relational database language (Mistral) in the work connected with the Chief Executive's Information System. The idea of using the many facilities available for re-formatting results, producing graphs, and displaying comparisons between the results of different runs of the model seemed very attractive. The output format is by its nature less easy to specify as a fixed format such that it will cope with all the various reasons for a particular run of the model.

In practice, re-formatting has been undertaken using a word-processor (at the moment it all has to be re-typed, an electronic link has been tested but has not been considered worth the effort required to make it work). The database was used in the early days for producing the graphs, but this is no longer the case due to two main reasons. Firstly, the volume of data necessary to draw a graph is usually very small. The time taken to exit from the 5-year planning model, load the database and specify the graph format was too long when compared with typing in a few numbers into our general graph drawing package.

Secondly, the required layout of each graph was always unique, and usually differed in important ways from the default settings easily available in the graphics part of the database. However, considerable use has been made of the facilities for certain specific exercises. For example, in order to understand the implied synergy in an acquisition, runs can be made of Imperial on its own, Company X on its own, and then Imperial plus Company X. The database then easily allows the user to table the "synergy" in the Balance Sheet, Profit & Loss or even the tax payments. Manually this would not only be very time consuming, but also error prone.

## 2. PC Link

All of the main reports can be sent to a PC. The PC is connected to the Spectrum as another peripheral (a printer!) and the Spectrum completely controls the transactions. The PC has to be configured to receive the data, which we do using a package called "Crosstalk". On defining which tables are to be sent, the Spectrum mimics the typing of PC-DOS commands at the keyboard, and a file is created on the PC for each output table requested. The format of the file is such that it can be imported into Lotus 1-2-3 (i.e. a PRN file). The complete table, including page and column headings as well as row titles is reproduced. This facility has not been heavily used as yet, and we are considering several alternatives to make the process simpler and quicker, including having the PC as the Spectrum terminal with Crosstalk (it is also a terminal emulator) running concurrently with 1-2-3.

## Conclusion

The model has become an essential part of Imperial's strategic planning process. The power available from a computerised financial model cannot be undersold. As to the decision to write the model in APL as opposed to one of the specialist financial planning languages, we have no regrets at all. It is true to say that we have not been held back by the computer/language from providing the user with any facility he required, and I am sure that would not have been true if we had not decided on APL. If we were making the decision today, I know that the competition is a lot closer, but I would be very interested to hear if any of them can provide exactly what our users have available today.

*Acknowledgements: The author wishes to thank Ian Henson and Peter Carrington for providing the substance, around which the model has been built. The views expressed in this paper are those of the author and are not necessarily shared by Imperial Group.*

Figure 1: Schematic of model

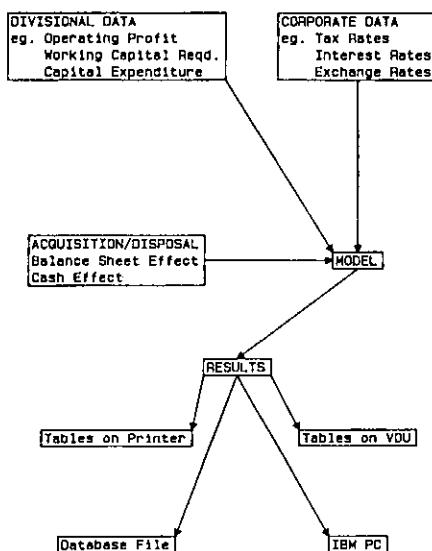


Figure 2. Main Output Options

1. Divisional Operating Profit
  2. Group Profit before Tax
  3. Group Consolidated Profits
  4. Group Balance Sheets (Assets)
  5. Group Balance Sheets (Financed by)
  6. Group Cash Flow
  7. Extraordinary Items
  8. Group Funds Statement
  9. Interest Charges Analysis
- 
- |            |                                   |
|------------|-----------------------------------|
| Print :    | 10. Hard Copy of the Main Results |
| Menu :     | 11. Output (Detailed) Options     |
| Database : | 12. Set up a File                 |
| IBM PC :   | 13. Transfer Output to an IBM PC  |
| Return :   | 14. Main Option Menu              |

# ANNOUNCING THE

# **dyalog** APL

## **COPROCESSOR FOR THE IBM PC**

The Dyalog APL coprocessor allows large scale second-generation APL applications to be developed and run on an IBM PC or plug-compatible computer.

*The Dyalog APL coprocessor consists of:*

- a 32-bit coprocessor board based on the NS 32000 chip set, hardware floating-point, up to 4Mb of on-board RAM, and 16Mb of virtual memory space.
- a complete implementation of UNIX\* System V including C, f77, Fortran and Pascal.
- a complete implementation of Dyalog APL.
- software that integrates the Dyalog APL coprocessor subsystem into the PC-DOS environment.
- a Dyalog APL character generator and set of keycap stickers.

The Dyalog APL coprocessor is easily installed in an IBM PC or plug-compatible system with a standard BIOS and a hard disk. The board simply slides into a single slot in the PC chassis, and the software is installed via an interactive menu.

The Dyalog APL coprocessor provides both DOS and UNIX System V environments without compromise. Data can be shared between the two environments, and a simple keyboard command switches between operating systems.

DOS commands and files can be accessed directly from Dyalog APL, and Dyalog APL can be run directly from DOS.

The Dyalog APL coprocessor supports very large APL objects and workspaces, to a maximum of nearly 16Mb; and floating-point performance is up to 3 times faster than an IBM PC AT.

Prices, including a multi-user Dyalog APL licence, start from around £3,000. A complete ready-to-run system with a fast 40 mb disk and tape back-up is also available for under £7,000.

*For further details and ordering information, please contact Peter Donnelly  
Dyadic systems Ltd., Park House, The High Street, Alton,  
Hampshire GU34 1EN, United Kingdom  
Telephone: Alton (0420) 87024 (10 lines) Telex 858811*

---

## 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.

Besides the contributed articles, this issue of VECTOR includes several regular features – “Surely there must be a better way”, APL Trivia, competition result and a new £50 competition.

---

## Technical Editorial: Utility functions

*by Jonathan Barman and Dave Ziemann*

Having just lost a couple of utility functions, and having spent an extremely aggravating couple of hours recreating them, the question arose of how to keep track of utility functions so that they are available when needed.

Ideally, there should be a book of every function that anyone could ever want, but although there are several good books published, they can never cover the full field. Everyone has slightly different requirements, and one person's utility function may be useless to another; particularly true where different APLs are used under different operating systems.

How should one keep utility functions? Probably the most common method on a mainframe computer is to keep them in public workspaces; each workspace containing the functions which will help solve a particular class of problems. New utility functions are usually created while developing a system, but in the general rush and panic of meeting a deadline the small job of adding the functions to the appropriate public workspace can be forgotten. Hence the "lost" functions, which were somewhere around but not in the proper utility workspace, or if they were, did not have a name that was very meaningful.

Another popular method of managing utilities is to store them on file, in their `□CR` or `□VR` formats. The application does not need to keep the functions in the stored workspace, but merely defines them when they are needed at run time. This has the advantage that each utility is stored centrally, but may be accessed by many users. In this way any given function can be revised by a supervisor and simultaneously made available to all the applications that use it. Such a system was described by Maurice Jordan of British Airways at a BAA meeting last year.

Part of the success of the FINNAPL idiom library is that it is well documented and has very simple code that is generally applicable. Several possible solutions are given for a problem, and although there is no attempt to say which method is best it is often rewarding to consider each of them.

"Quote-Quad" has a regular column of algorithms, but has not as yet published a compendium, so all the past issues have to be purchased, which is rather awkward for newcomers to APL.

Has anyone got a documented set of utility functions that could be published? Do you feel there is a need for such a publication, or should we all write our own utilities? Let's hear your views.



## Technical Correspondence

*This column features letters which are unlikely to interest readers who do not already know APL. Writers are requested to observe the requirements for the inclusion of APL code in VECTOR. All correspondence should be addressed to the Editor, who will forward technical letters to the technical editorial team.*

From Claude Henriod

19th December 1985

Sir: The article "Steps toward a better APL" in VECTOR 2.2 page 91, has especially retain my attention. Each programming language has its own characteristics, beauty and lack compared with the other languages.

Why disfigure the APL with a simulation of Basic?

In short:

- the APL presented code contain some missusage
- some functions are not in a good APL (NEXT and THEN)
- the simulator is a bad Basic

Consequent, the two languages are diminished.

Please maintain APL as it is!!

I have not sufficient vocabulary to give my disappointment in English, accordingly you will find an annex with more technical details in French, and APL.

I suggest to you to transmit the document at Ivor Kenson.

Claude Henriod  
Manteiller CH - 1965 SAVIESE

*Ed: It seems that some of Ivor Kenson's Anglo-Saxon humour has failed to come across in translation. Claude's technical details appear later in this issue.*

From: Eduardo Vesperoni

7th January 1986

Sir: As an affectionate reader of VECTOR, I am glad of submit an entry for your competition on range union, just for gratify you (I hope) with an entry also from Italy.

Yours,

Eduardo Vesperoni  
Vefreria Parmense Bormioli Rocco s.p.a.  
Casella Postale n.354 Parma

From J Sullivan

11th December 1985

Sir: I enclose my competition entry, which was written in IBM APL2 release 1.0 and run on an IBM 3084. However, the functions also works on our IBM PC running APL\*Plus Release 4.0.

This function does not work if the shape of the input matrix is  $0 \times 2$ ; however the competition specifies that input will always be a 2-column integer matrix (which implies it is never null) so special checks will not be necessary (so I haven't included any for the null-input condition).

You might like to know that about a week after developing this function I suddenly found a use for it; so your competitions do not only while away the time but are useful as well.

Yours faithfully,

J Sullivan  
Research Officer  
Statistics & Market Research Section  
National Westminster Bank PLC  
41 Lothbury  
London EC2P 2BP

*Ed: Thanks for pointing out the usefulness of the competitions! We think that an integer matrix includes a 0 row matrix. It also says much for the portability of APL that you can run your code not only on such different sized machines but using versions of APL from different suppliers.*

## Prize Competition result: "Range Union"

by Jonathan Barman

Competitors had to provide a function that combined overlapping and duplicate ranges in a set of ranges, and returned the 'union' of the set. For example:

```

      A
1   4
12  12
8   11
8   10
2   2
      RANGEUNION A
1   4
8   12

```

Twelve entries were received which exhibited almost as many different methods of solving the problem. The fastest, and the most elegant, method was submitted by Paul Chapman and also Morten Kromberg, who used the ingenious method of taking the maximum scan of the sorted matrix of ranges to get the set of overall start and end ranges. It is easiest to understand the method with a worked example.

The first step is to sort the matrix so that the range starts are in sequence:

```

      A[A[:,0]I0];]
1   4
2   2
8   11
8   10
12  12

```

and then do the maximum scan:

```

      [A[A[:,0]I0];]
1   4
2   4
8   11
8   11
12  12

```

The maximum scan has the effect of forcing the end of the ranges to be the biggest so far encountered. One competitor missed out this vital step, but sorted both columns of the matrix, which only produces an erroneous result if one range is wholly included inside another range. In the example the range 2 to 2 is wholly included inside the range 1 to 4.

The ranges required are those where the start of each range is greater than the end of the previous range. Rotating one column of the matrix makes the the comparison between the starts and previous ends easy:

```

      0 1e[ \A[ \AA[ ; \IO]; ]
1 12
2 4
8 4
8 11
12 11

```

At this point both Paul and Morten noticed that a temporary variable could be formed which could be used later when generating the result, so the first line of the function is:

```
R+0 1e[ \A[ \AA[ ; \IO]; ]
```

The first row is always required, as it contains the lowest start which is correct for the first range, and the highest end which is correct for the last range. The remaining rows are needed if start is larger than the previous end:

```

      1,1+1<-/R
1 0 1 0 0

```

and this is used to select the rows of the temporary result:

```

      (1,1+1<-/R)≠R
1 12
8 4

```

Finally, the second column is so rotated so that the previous range ends become the current range ends:

```

      0 1e(1,1+1<-/R)≠R
1 4
8 12

```

So the final function becomes:

```

v R+RANGEUNION A
[1] A <R> is union of ranges in two column matrix <A>
[2] R+ 0 1 e[ \A[ \AA[ ; \IO]; ]
[3] R+ 0 1 e(1,1+1<-/R)≠R
v

```

Many other methods were used, including a looping solution. Several solutions used the outer product to compare the start and end points with the full set of possible numbers; which results in WS FULL if the disparity between ranges is large. As in previous competitions, competitors did not always test their code in index origin zero or with zero row matrices.

The problem stated that the right argument could be assumed to be a 2-column integer matrix, but it was not stated that the ranges would always be given with the starts in the first column. Only two solutions included code to reverse the columns, and Mark Longstaff's method was best:

```
(>/A)ΦA
```

As it was stated that no checking of the matrix was necessary, it was decided not to make this check a requirement for a successful entry.

The CPU times varied enormously, from 0.44 to 11.59 seconds (for a matrix of 100 ranges, on an IBM PC XT running APL\*PLUS APL). Morten Kromberg's solution was the fastest. He transposed the matrix at the beginning and again at the end so that the rotates could be applied along the last axis.

Paul Chapman gave a very good analysis of how he came to save the temporary result, which is worth reproducing. In his example R is the sorted right argument:

By comparing each max-scanned range end with the following range start, we can establish where the breaks occur:

$$x \leftarrow 1 < 1 + - / 0 \quad \bar{1} \quad \ominus \Gamma \lambda R$$

This gives us a 1 between each pair of ranges which are not connected, so that there is one less 1 than rows in the desired result. Now the result range starts are:

$$(1, x) / R [ ; \bar{1} 0 ]$$

and the result range ends are:

$$(x, 1) / \Gamma \lambda R [ ; \bar{1} 0 + 1 ]$$

which also happens to be:

$$1 \phi (1, x) / \bar{1} \phi \Gamma \lambda R [ ; \bar{1} 0 + 1 ]$$

so the two operations can be combined:

$$0 \quad 1 \ominus (1, x) / 0 \quad \bar{1} \quad \ominus \Gamma \lambda R$$

But we have already calculated

$$0 \quad \bar{1} \quad \ominus \Gamma \lambda R \quad .$$

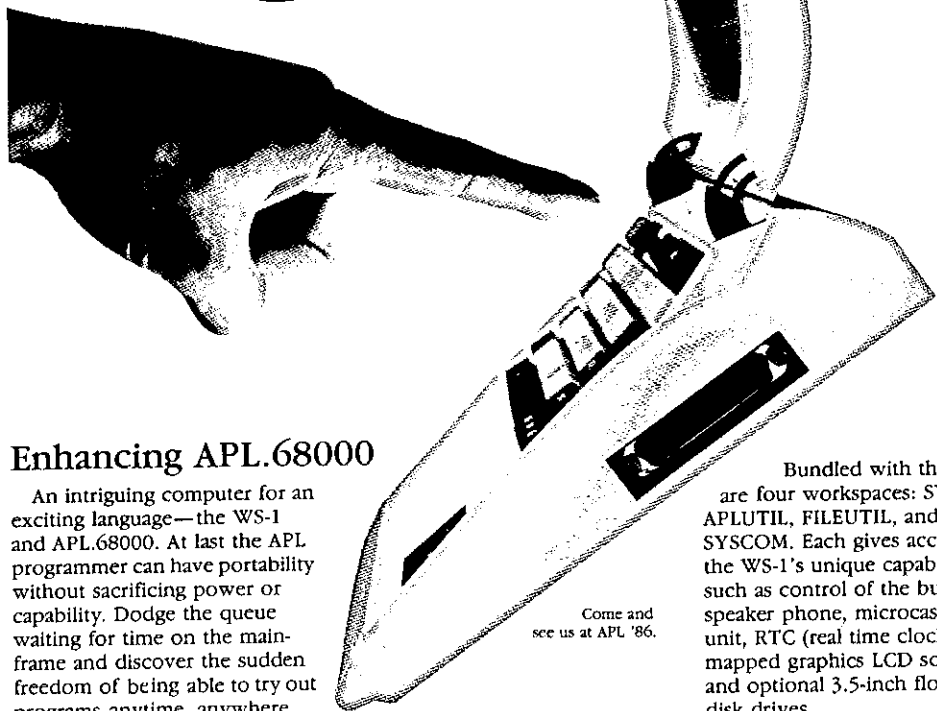
so we can simplify:

$$\begin{aligned} z &\leftarrow 0 \quad \bar{1} \quad \ominus \Gamma \lambda R \\ x &\leftarrow 1 < 1 + - / z \\ R &\leftarrow 0 \quad 1 \quad \ominus (1, x) / z \end{aligned}$$

Paul Chapman and Morten Kromberg are the joint winners. Paul included code to sort the second column of the matrix, but this is not strictly necessary as the maximum scan takes care of items being out of sequence within each set of starts. Morten did not use the minus reduction in order to compare the break points, so his code looked a little less elegant. Third prize goes to Mike Day who gave a fast solution using a slightly different method.

We were again pleased to see entries from overseas, this time they included correspondents from Italy, Sweden and Switzerland. It was also good to see a solution developed on the QL.

# POWER



## Enhancing APL.68000

An intriguing computer for an exciting language—the WS-1 and APL.68000. At last the APL programmer can have portability without sacrificing power or capability. Dodge the queue waiting for time on the main-frame and discover the sudden freedom of being able to try out programs anytime, anywhere.

The APL.68000 interpreter is implemented in 86KB of ROM, running under a multi-user, multi-tasking operating system called BIG. DOS. Speed is the essence of APL programming, and now the WS-1 makes development even faster.

APL.68000 on the WS-1 has attractive enhancements such as a powerful component file system, QUAD. FMT function for alpha report formatting, QUAD. CC function for full-screen control, and extended error trapping facilities.

Come and see us at APL '86.

Bundled with the WS-1 are four workspaces: SYSFNS, APLUTIL, FILEUTIL, and SYSCOM. Each gives access to the WS-1's unique capabilities such as control of the built-in speaker phone, microcassette unit, RTC (real time clock), bit-mapped graphics LCD screen, and optional 3.5-inch floppy disk drives.

Compress these capabilities into a sleek footprint measuring less than 13 inches by 11 inches, and you have the ultimate definition of power.

## ampère

FOR DISTRIBUTORSHIP INFORMATION AND PRODUCT DETAILS PLEASE CONTACT:

**ampère**  
INCORPORATED

Asahi Bldg., 7-5-20 Nishi-Shinjuku, Shinjuku-ku, Tokyo, Japan. Phone: 03-365-0825.  
Telefax: 03-365-0999. Telex: J33101 AMPERE. IP Sharp Mail Box Code AMP (Group Code APLWS).

## Prize Competition: Watch your step

by David Ziemann

It's a common requirement when writing APL to want to generate a set of indices from 1 up to some integer N. We can do this simply by using monadic iota, the index generator function, in origin-1:

```
      ⍵10
1 2 3 4 5 6 7 8 9 10
```

Sometimes we need to start counting from a number other than 1 however, or even to count backwards from a high number to a lower one. In such cases the function <TO> can be used:

```
      3 TO 10
3 4 5 6 7 8 9 10
      8 TO 2
8 7 6 5 4 3 2
      -3 TO 5
-3 -2 -1 0 1 2 3 4 5
```

By the way, <TO> can also be used to generate multiple range vectors, for example:

```
      3 TO 5 4 TO -1 5 TO 2
3 4 5 4 3 2 1 0 -1 5 4 3 2
```

Now the function <TO> is very useful in APL systems, but occasionally a little more is needed. In particular, what we want is a function which will produce a matrix of range vectors from an input matrix of ranges. To make things even more exciting we want to be able to specify the step in each case. Here's an example of a function called <STEP> which demonstrates the required behaviour:

```
      ⍵+M1+5 3p10 27 5,80 100 20,113 100 3,-8 -3 2,-1 -3 1
10 27 5
80 100 20
113 100 3
-8 -3 2
-1 -3 1
      0 STEP M1
10 15 20 25 0
80 100 0 0 0
113 110 107 104 101
-8 -6 -4 0 0
-1 -2 -3 0 0
```

Each row in the right argument specifies a start-point number, an end-point number and a step. The start and end-points may be positive or negative, but the step must be positive.

The shorter range vectors in the rows of the result matrix are padded with trailing zeros. This leads to the possibility of confusion of course, if a range vector should itself contain a zero:

```

      □+M2+3 3p8  3 1,-6 6 2,-2 2 1
      8  -3  1
      -6  6  2
      -2  2  1
      0 STEP M2
      8  7  6  5  4  3  2  1  0  -1  -2  -3
      -6 -4 -2  0  2  4  6  0  0  0  0  0
      -2 -1  0  1  2  0  0  0  0  0  0  0
  
```

That's where the left argument comes in – it's used as the trailing pad number so that you can choose a pad number you know cannot occur in your particular set of ranges. So, changing the left argument:

```

      99 STEP M2
      8  7  6  5  4  3  2  1  0  -1  -2  -3
      -6 -4 -2  0  2  4  6  99 99 99 99 99
      -2 -1  0  1  2  99 99 99 99 99 99
  
```

The competition this issue requires you to write the function <STEP> as shown and described above. As usual FIFTY POUNDS in prize money will be shared among the winners – those who produce the best answers, in the opinion of the judges. The closing date for entries is 30th July 1986.

### Competition Rules

- Entries must be in legible English or APL as appropriate and should preferably be machine produced.
- Entrants must declare the type of computer and the version and release level of the APL interpreter on which their functions were written.
- The date and your full name and address should appear on each sheet of your entry.
- Entries should be physically separate from other contributions such as letters, and should be clearly marked 'Competition Entry'.
- All submissions should be sent to the editor.
- Those on the committee, activities group or journal group of the British APL Association are ineligible.
- DOS format diskettes containing APL \*PLUS, IBM or Sharp APL workspaces are acceptable; diskettes will be returned.
- Unless otherwise stated, entrants should submit only one entry. We encourage submission of alternative approaches, but the entrant must indicate clearly which one of his answers is his entry in the competition.
- Non-members are encouraged to enter the competition, with the proviso that, should they win, part of their prize will comprise free B.A.A. membership for the current year.
- Late competition entries may be accepted if the competition has not yet been judged.



## Surely there must be a better way

compiled by David Ziemann

### Fuzzy Look-Ups

Simon Barker from Catford, London has sent us a function which he uses for doing what he calls a fuzzy table lookup. Here is Simon to explain his problem and solution:

"In several applications used in my place of work there is a need to take into account mis-spellings of names used for selecting records from a file, the name being the key to an individual record. As an example, the names 'PETER' and 'PEATER' should both match a record with a key of 'PETER'. In effect a sort of fuzzy version of the commonplace expression

```
MAT^.=VEC
```

"is required, where <MAT> contains a matrix of names and <VEC> contains a name that is to be matched against <MAT>. The fuzzy version of the above expression might look like this:

```
□→MAT+4 5p'CATS DOGS .....A.P.L'
badchrs+'AEIOU .'
MAT MATCH 'PL.'
```

```
0 0 0 1
```

"The variable <badchrs> contains characters that are to be ignored in the matching process. My solution is as follows:

```
V R+SET MATCH ITEM;SH;M;DF;Z
[1] A DOES EQUIVALENT OF <SET> ^.= <ITEM>
[2] A <ITEM> IS ANY CHAR. VECTOR, <SET> IS ANY CHAR. MATRIX
[3] A MATCHING IGNORES ANY CHARS. ASSIGNED TO GLOBAL VAR <badchrs>
[4] A
[5] A IF GLOBAL VARIABLE HAS NOT BEEN ASSIGNED, SET IT TO DEFAULT
[6] A
[7] A (0=□NC 'badchrs')/'badchrs+'AEIOUY ..;/:-+'
[8] A
[9] SH+pSET
[10] M+~SETεbadchrs
[11] DF+(Z+[+/+M)-L+/+M
[12] M+.M.(Z-+/+M).ε≥1DF
[13] SET+M/,(SH+0,DF)+SET
[14] SET+(SH[1],Z)pSET
[15] R+SET^.=Z+(~ITEMεbadchrs)/ITEM
V
```

"Although reasonably efficient, the function can only cope with a vector right argument. I would be very interested to see an improved version of <MATCH> which could emulate the standard inner product solution to a higher degree, i.e. matrices on either/both side(s) and numbers as well as text. One final point, the solution I have supplied does not behave consistently when the right argument is made up of characters that are all contained in <badchrs> and the left argument has a row of spaces in it. Can you spot why this is so, and propose a generalised solution?"

We would be glad to receive any comments or improvements on Simon's function. Have any other readers worked in this area?

**Phonetic Searching**

Another technique for locating names in a database or table of names is Soundex searching, which can return entries that sound alike but are spelt differently. The Soundex system works by converting the name to a four digit key. There are four steps to this process. First, each letter of the name except the first is replaced by the corresponding number from the following table:

Letter	Number
A,E,H,I,O,U,W,Y	0
B,F,P,V	1
C,G,J,K,Q,S,X,Z	2
D,T	3
L	4
M,N	5
R	6

Second, all successive occurrences of the same digit are removed from the string. Next, all zeros are deleted. The last step is to make the code four characters long, either by truncation or by padding with zeros. Here's an example of the process using the name "TISSERAND":

Step	Code
1	T02206053
2	T0206053
3	T2653
4	T265

The following functions implement the above algorithm directly:

```

V R←SOUNDEXACODE W;L;N
[1] A Return the SOUNDEX code corresponding to the name in <W>
[2] A <R> is a 4 character code. <W> is a character vector
[3] A Alphabet
[4] L←'AaEeHhIiOoUuWwYyBbFfPpVvCcGgJjKkQqSsXxZzDdTtLlMmNnRr'
[5] A Correspondence vector from alphabet to digits
[6] N← 16 8 16 4 2 4 2 REP '0123456'
[7] A Map all but first letter onto digits
[8] R←(N,'0')[Lil+W]
[9] A Delete consecutive duplicate digits
[10] R←(R≠lφR)/R
[11] A Delete zero digits
[12] R←(R≠'0')/R
[13] A Pad or truncate code to four characters
[14] R←4+(1+W),R,'000'
V
    
```

```

V R+A REP W;□IO
[1] A Return a vector of elements from <W> replicated by <A>
[2] A Arguments may be scalars or vectors only
[3] A EXAMPLE : 2 0 3 1 REP 'ABCD' ↔ 'AACCCD'
[4] □IO+0
[5] A+(ρW+R/W)ρA+(R+xA)/A
[6] R+W+(1+A)ε+VA]
V

```

The function copes with lowercase as well as uppercase alphabetic characters. Users of APLs that support the "replicate" extension of the compress function may replace the call to function <REP> on line 4 by a /. Alternatively the line can be recoded as the catenation of reshapes. Here's the function in action:

```

SOUNDEXΔCODE'TISSERAND'
T265
SOUNDEXΔCODE'TISEROND'
T265
SOUNDEXΔCODE'TIZZEWRONGED'
T265

```

If a database or table of names already exists, it is more useful to be able to directly convert a matrix of names into their Soundex codes, for storage with the original names. The function can be modified to deal with a matrix argument as follows:

```

V R+SOUNDEXΔCODES W;L;N
[1] A Return SOUNDEX codes corresponding to the names in <W>
[2] A <R> is a 4 column matrix of codes. <W> is a char. matrix.
[3] A Alphabet
[4] L+'AaEeHhIiOoUuWwYyBbFfPpVvCcGgJjKkQqSsXxZzDdTtLlMmNnRr'
[5] A Correspondence vector between alphabet and digits
[6] N+ 16 8 16 4 2 4 2 REP '0123456'
[7] A Map all but first letter onto digits
[8] R+(N.'0')[L1 0 1 +W]
[9] A Delete successive occurrences of same digits
[10] R+(.l.R≠lQR)/, '/',R
[11] A Delete zero digits
[12] R+(R≠'0')/R
[13] A Pad or truncate codes to four characters
[14] R+W[;1],MIM R
[15] S+ρR+((1+ρR),4)+R
[16] R+,R
[17] R[(R=' ')/1ρR]+'0'
[18] R+SρR
V

```

```

V R+MIM W;□IO;V;B;I
[1] A Make character vector <W> into character matrix <R>
[2] A Rows of <R> are delimited by 1†<W> in <W>
[3] A 2↔→ppR
[4] □IO+1
[5] V+W.1+W
[6] B+Vε1+W
[7] I+1+I-1+0,-1+I+B/1ρB
[8] R+I°.≥:1[/I
[9] R+(ρR)ρ(.R)\(~B)/V
V

```

The function <MIM> is a standard Make-Into-Matrix function which takes the first element of its argument as a delimiter indicating where the argument should be broken into rows in the resulting matrix.

Here are the results of applying the function to a table of distortions of the word "Wensleydale":

Notice how the last attempt does not produce the same Soundex code as the others and therefore would not constitute a "hit" on the name.

## APL Trivia

compiled by David Ziemann

### Life, the Universe and Everything?

In VECTOR Vol 2, No.2, we looked at a function that applied all the monadic primitives to 0 to produce the answer to the ultimate question, as "Hitch-hiker" fans will recall. A challenge was set to find the largest possible number obtained by scrambling the order of the primitives. Claude Henriod from Switzerland produced this interesting piece of code:

```
    ▽ R+BIGANSWER1
[1] R+εφ▽-●| |+⊞○†+Γ| ×⊞⊙~ρ▽Δ. ? 1 0
    ▽

    BIGANSWER1
1.836014905E10
    □PP+17
    BIGANSWER1
6.53239608360149E17
    □PP+10
```

Can you see why the result gets larger as the value of the system variable □PP is increased? Mark Bassett supplied the following code to produce the biggest answer he could find:

```
    ▽ R+BIGANSWER2
[1] R+|Γ*| |-○φ⊞⊞†~+×ε▽ρ?▽Δ●. 1 0
    ▽

    BIGANSWER2
1.269640335E73
```

Can anyone improve on this, or perhaps prove that it can or cannot be bettered?

It seems that "Ivor Kenson's" article "Steps Toward a Better APL" caused a bit of confusion, and was taken seriously by at least two people! (Now doesn't that name seem somehow familiar?). Claude Henriod was so incensed that he has not only produced a criticism of the original article, but also sent us with his own version of what a simulation of BASIC in APL should look like. Let's have a look at what Claude has to say, with thanks to Helen Piper for providing the translation, but after this, please – no more BASIC emulators!

### Remarks on "Towards a Better APL"

by Claude Henriod

In general one has the impression that the author of these APL functions is a FORTRAN programmer.

The APL listings which accompany these notes have been developed and tested with APL-68000 V5.0 on an Ampere computer.

### 1. Some simple improvements:

- \* ELSE, RETURN, STOP are unchanged.
- \* GOTO, IF are simplified.
- \* BASICDEMO and my equivalent FIBOBASIC are no comparison with the APL function FIBO.
- \* THEN uses 4 lines and 2 branches and is too complicated in APL.
- \* NEXT is so much simpler in APL.
- \* STEP is a BASIC keyword which must be included.
- \* ON is associated with GOTO, for example:  
     GOTO (L100,L200,L300) ON INDEX

### 2. Comments on Mr. Kenson's code

- Allowing assignment to □WA is a weakness in some interpreters. It is always dangerous to use the errors of a system, since a new version could correct them.
- APL-68000 does not allow assignment into □WA, and also does not understand □ALX or □ELX, and so I think it is preferable to use local variables with a naming standard.
- FOR and INPUT contain the same sequence of code which can be extracted as a sub-function (see also READ).
  - Line 3 is simplified
  - Lines 2 and 4-6 are moved to SUBEXTRACT
- INPUT is closer to BASIC if a prompt is incorporated. It would also be necessary to initialise the variable or variables, as with READ.
- PRINT is a very inadequate copy of the BASIC command. It is very simple to simulate it in 3 lines (see the code in FIBOBASIC).

### 3. BASIC extensions

- FOR does not correspond to the BASIC version:
  - lacks the ability to use a decreasing sequence
  - cannot be nested
  - the FOR....NEXT loop is executed at least once (DO-UNTIL instead of DO-WHILE).

I have not been able to resist the pleasure of completing M. Kenson's study:

- READ, DATA and RESTORE are new functions
- READ uses SUBEXTRACT. Constraint: The variable must be initialised with the maximum length of the data, which can be likened to a DIM statement.
- DATA requires an argument of a single data type, but this is not a great handicap thanks to RESTORE.
- RESTORE n. If n=0, the first line of DATA is restored. If n is not a line of DATA, RESTORE has no effect on the sequence of data.

APL-68000 has certain differences which I have used, for example Nomadic functions, but it could be written without the use of such extensions.

□SI is "vectorised", so I have used □BOX. □SS directly gives the index of the argument being sought.

I have not copyrighted, nor do I make any reservations on the presented code, because I am only concerned with the promotion of APL.

```

▽ DATA ΔΣΔ
[1] →(2=DN←DATAΔCTRL)/DL←1 ⋄ DATAΔCTRL←1⋄
[2] ΔΣΔ←1↑2↑DL ⋄ →(ΔΣΔ≠DATAΔCTRL)/0
[3] DATAΔCTRL←DATAΔCTRL,ΔΣΔ

```

▽ R←X ELSE Y

```
[1] R←X,Y
```

▽ X←FIBO N

```

[1] ⌘ FIBONACCI SERIE
[2] X←1 1
[3] →(N)⌘X←X,+/2↑X)/DL

```

▽ FIBOΔBASIC

```

[1] ⌘ REM SIMULATED BASIC PROGRAMM IN APL
[2] N←0
[3] 'GIVE FIBO LIMIT'INPUT N
[4] L100:→IF(N≠0)THEN STOP ELSE L200
[5] L200:I←1
[6] R←0
[7] S←1
[8] L300:→IF(I=N)THEN GOTO L400
[9] →GOSUB L500
[10] PRINT I,R
[11] I←I+1
[12] →GOTO L300
[13] L400:→GOSUB L500
[14] PRINT I,R
[15] X←''
[16] FOR I←1 TO R
[17] PRINT '*,'
[18] →NEXT I
[19] PRINT '* '
[20] →STOP
[21] L500:T←S
[22] S←R+S
[23] R←T
[24] →RETURN

```

▽ FOR FOR;ΔAΔ;ΔEΔ;ΔIΔ;ΔLΔ;ΔSΔ

```

[1] ΔAΔ←1↑1+1↑2↑DL
[2] ΔLΔ←SUBΔEXTRACT'FOR'
[3] ΔEΔ←1↑(ΔIΔ+ΔLΔ)'+'↑ΔLΔ
[4] I'ΔIΔ+',(ΔIΔ)↑ΔLΔ

```

```

[5]  IAE, +1+AI
[6]  AI+I+1+2+AI
[7]  AI+I+1+3+AI,1
[8]  FOR=NEXT+(I, AI, 2, AE, I, AE, I, AI, 1) / TAA
[9]  0 0DFX((FOR)+NEXT+NEXT NEXT), (010-0.5)FOR
  ▽
  ▽ P+GOSUB R;AI:AX
[11]  AI+I+1+1+2+AI
[12]  AX+(8I+AX)IAX+R, AI
[13]  0 0DFX((AX)+R+RETURN), (010-0.5)AX
  ▽
  ▽ R+GOTO R
  ▽
  ▽ P+IF P
  ▽
  ▽ AI INPUT INPUT;AL:AE
[14]  AI+I, INPUT  AE+0=0\0 INPUT  INPUT+SUBAEEXTRACT'INPUT'
[15]  +(2=0NC'AI)/DLC+1  PRINT AI
[16]  AE/DLC+1  AE+0  +DLC+2
[17]  AI+(1+I+AI)X; '=1+AI+, AI  AE+AI+0
[18]  AE+(AL, AE)AE
[19]  X:=0\0AE/DLC+1  INPUT, '+, AE  +0
[20]  INPUT, ' ', AE, ' '
  ▽
  ▽ NEXT+NEXT NEXT
[21]  NEXT+(89:1+1+1)/16
  ▽
  ▽ R+L ON I
[22]  +(I>0, L+L)/R+0
[23]  R+L(010-1-I)
  ▽
  ▽ PRINT AX
[24]  +(0=0\0AX)/DLC+1  AX  +0
[25]  +(I; '=1+AX)/DLC+1  AX  +0
[26]  0+1+AX
  ▽
  ▽ READ READ;AL:AE;L
[27]  AL+I, READ  READ+SUBAEEXTRACT'READ'
[28]  +(2=0NC'DATALIST)/DLC+1  +(0=DATALIST)/DLC+1  AI
[29]  DATALIST+(1+DATACTRL)SUBAEEXTRACT'DATA' DATACTRL+1+DATACTRL
[30]  AI:AL+010-DATALIST',  AE+AL+DATA
[31]  DATALIST+(1+AL)DATA  AE+(AL, AE)AE
[32]  +(0=0\0AE)/DLC+1  READ, '+, AE  +0
[33]  READ, ' ', AE, ' '
  ▽

```



## ▽ READ BASIC

```

[1] DATA 1,3,5,7,11,13,17,19,23,29
[2] DATA 10
[3] DATA 'BRITAIN', 'FRANCE', 'GERMANY', 'SWITZERLAND', 'ITALY'
[4] RESTORE 3
[5] FOR NO+1 TO 4
[6] TXT+8$' '
[7] READ TXT
[8] PRINT NO
[9] PRINT TXT
[10] →NEXT NO
[11] RESTORE 2
[12] CNT+0
[13] READ CNT
[14] RESTORE 0
[15] FOR NO+1 TO CNT STEP 2
[16] VALUE+0
[17] READ VALUE
[18] PRINT VALUE
[19] →NEXT NO
[20] →STOP

```

## ▽ RESTORE ΔXΔ

```

[1] +(0≠ΔXΔ)/DLC+1 ◊ ΔXΔ+1/DATAΔCTRL
[2] DATAΔCTRL+(-DIO-DATAΔCTRL:ΔXΔ)◊DATAΔCTRL
[3] 0 8$DEX'DATAΔLIST'

```

▽

## ▽ R+RETURN

```
[1] R+13
```

▽

## ▽ R+X STEP Y

```
[1] R+X,Y
```

▽

## ▽ R+STOP

```
[1] R+0
```

▽

## ▽ ΔZΔ+ΔLΔ SUBΔEXTRACT ΔNΔ;DIO

```

[1] +(2=UNC'ΔLΔ')/DLC+1 ◊ ΔLΔ+1+3+DLC
[2] ΔRΔ+(QR DBOX OSI)(2+DIO+1;1)
[3] ΔRΔ+OCR 1+(ΔRΔ:(')↑ΔRΔ
[4] ΔRΔ+ΔRΔ[1+ΔLΔ;]
[5] ΔRΔ+((1+ΔNΔ)+1+QSS(ΔRΔ;ΔNΔ))↓ΔRΔ

```

▽

## ▽ R+C THEN L

```
[1] R+(C,(~C)×2=ΔL)/L
```

▽

## ▽ R+X TO Y

```
[1] R+X,Y
```

▽

# Application Prototype Environment

*an IBM program product for prototyping  
and developing screen-based applications under VS APL or APL2*

This product is intended to be used both by the professional programmer and by business professionals, such as managers, planners, scientists, and engineers. The product is particularly suited for developing applications in Information Centre and Application Development Centre environments.

Three interactive functions are provided to easily and quickly define the end user interface.

**Panel Design:** For creating menu selection, action, and tutorial panels.

**Chart Design:** Charts (Plot, Surface, Bar, and Pie) are designed via an interactive dialogue.

**File Handling:** The interactive dialogue allows users to specify APL File, CMS, VSAM, QSAM, BDAM, and INSTOR-AGE access methods.

Over 70 cover functions are provided for building the application program. Some examples are:

## *Panel Operations*

- Automatic Screen I/O.
- Automatic display of tutorials.
- Program function key control.

## *File Handling*

- Reading/writing records from/to a file.
- Positioning the record pointer.
- Relative record access.

## *Object Library*

The object library is used for storing, retrieving, and status checking of objects such as panels, functions, declarations, and variables.

### **Free 30-day Test Period**

You have 30 days to test Application Prototype Environment to see how well it performs for you. If you are not completely satisfied, just send all materials back to your local IBM Branch Office within 30 days and you will not be billed.

**For More Information:** Contact your IBM sales representative or the nearest IBM Branch Office.

### **Technical Data**

Application Prototype Environment is an IBM Licensed Program Product, Program Number 5668-896.

The program runs under CMS and TSO together with the following IBM programs or their equivalents: APL2 or VS APL, GDDM (Graphical Data Display Manager). Some examples of terminals supported are: 3277, 3279, 3270 PC/G and CX.



IBM United Kingdom Limited

---

## Introduction to Technical Articles

This issue's contributed articles concentrate on different aspects of mainframe APL.

Mark Lenihan, of Natwest Bank, offers a valuable history of how they are handling the transition from VSAPL to APL2. We hope they will soon be able to tell us about their experiences using the new facilities of APL2 and the benefits it has given them.

For those users still running VSAPL under TSO, David Piper – a consultant with Cocking & Drury – offers some practical suggestions in response to some of the points raised by Adrian Smith a couple of issues ago.

---

# NEW MAINFRAME SOFTWARE FOR IBM'S® APL

Now available in the UK, two new offerings from STSC that enhance IBM's mainframe APL implementations

---

## If you're staying with VS APL ...

---

### **COMPILER**

The first commercial compiler for APL compiles functions individually. Results in significantly faster execution. Interpreted functions can call compiled functions and vice versa.

---

## If you're migrating to APL2 ...

---

### **SHAREFILE/AP**

STSC's popular APL component file system is now available under APL2. Multi-user, nested array storage, libraries, access matrices. Multiple file system support. International language translations.

For full information, contact the APL\*PLUS™ Product Group, Cocking & Drury on 01-493 6172.

Trademarks/Owners: IBM/International Business Machines Corporation · APL\*PLUS/STSC, Inc.



**COCKING & DRURY LTD.**  
THE APL PROFESSIONALS

16 BERKELEY STREET · LONDON · W1X 5AE  
Tel: (01)493 6172 · Tlx: 23152 MONREF G

## Migrating VSAPL to APL2 under MVS/TSO

*by Mark Leniham*

The National Westminster Bank has one of the largest data-processing facilities in Europe and runs MVS/XA with TSO for end user computing requirements. Two centres, one in London and the other in the Midlands, are utilised with the majority of end user computing carried out at the Midlands Centre. Use of these facilities by non-DP staff is made using high level languages such as PL/I, SAS, GIS and APL to produce management information. Many of these APL users are business analysts with very limited knowledge of the TSO system as a whole, but have found VSAPL fits their requirements for a programming language in that it is easy to use, interactive and allows a level of flexibility not available with other languages. The decision to upgrade operating systems from MVS/370 to MVS/XA would impact all TSO users, but especially those using APL.

The decision to install APL2 followed directly from the decision to install MVS/XA. APL2 would not only exploit the advantages of XA but would also offer access to SQL/DB2 ( a database management facility) and ISPF ( a panel driven development system used widely throughout the Bank) via two new auxiliary processors. It also promised a full Nested Arrays implementation along with workspaces up to 128Mb! The installation of APL2 on the first of the Bank's two TSO systems which support APL users was carried out as soon as all prerequisite products (GDDM, ISPF2, etc) had been installed.

It was decided that APL2 would completely replace VSAPL when the switch from MVS/370 to MVS/XA was made. This entailed the migration of *all* VSAPL workspaces already written and those under development to APL2. Due to the urgency placed on the implementation of MVS/XA, the short-term migration procedure would entail making only the changes essential to allow the workspaces to run under APL2. The migration procedure was further complicated by widespread use of AP124, an auxiliary processor unavailable under APL2 which was used in all VSAPL applications which access full screen panels. Any such workspaces would have to go through a further migration step in which all references to AP124 would be replaced with corresponding calls to AP126 (the Graphical Data Display Manager auxiliary processor). Although AP126 was available under VSAPL, GDDM had not been installed due to the fact that AP124 offered a viable alternative and GDDM was not a prerequisite of VSAPL. However GDDM is now required to allow use of the new fullscreen APL2 function/variable editor.

Under VSAPL, users were allocated a partitioned dataset (PDS), named as 'userid.A.APL' where 'userid' is the users unique logon identifier. In this PDS, users were able to permanently save private VSAPL workspaces by copying them from their loadable form ('userid.APL.wsid') as created by VSAPL to a partition in the A.APL library. The reason for the use of a PDS for permanent workspace storage is based upon system limitations. All TSO users are able to write datasets (including APL workspaces) to several of the many 3380s allocated as TSO workpacks. These workpacks are cleaned up daily by housekeeping compress jobs which delete any datasets on these packs which have not been placed into the housekeeping exceptions listing. Such housekeeping is essential to free required storage available for use by other TSO users. By copying APL workspaces from the TSO

workpacks to the users own PDS on a permanent pack, space is freed on the overloaded workpacks. The use of project libraries create a dataset 'userid.APLnnnn.wsid' where nnnn specifies the library number. A dummy entry is also placed into the master catalog 'A.APLnnnn.userid' to allow VSAPL to cross-reference library number 'nnnn' to the correct 'userid'.

For APL2 implementation, it was decided that *all* APL users would be allocated an A.APL2 PDS in which the user could store APL2 workspaces. This would allow users to have two copies of each APL workspace (one VSAPL and the other APL2) for the duration of the migration, when both an MVS/370 system with VSAPL and an MVS/XA system with APL2 would be available. The APL2 options module was modified for our installation so that the APL2 workspaces would be )SAVED as 'userid.APL2.wsid'. When the VSAPL option module was attached to APL2, )MCOPY was able to reference workspace datasets set up by VSAPL directly. The next step along the migration path was to decide upon the method for APL2 invocation (CLIST, ISPF, etc). Over the past ten years since VSAPL was first installed, each user section has developed its own standards for entering VSAPL, and copying workspaces to and from the aforementioned PDSs. To attempt to force users into using any new centralised methods would only have caused further confusion within these departments. Upon making the decision to allow users to continue to access APL2 in the same way as VSAPL, the details regarding APL2 invocation were dispersed to users in a series of meetings and technical demonstrations.

Once user migration had started, the migration of project/public libraries began. At this stage, a major change to previous VSAPL strategy was made. Under VSAPL users were allowed to create new project libraries virtually at will. This led to libraries being allocated which in some cases were never used, or only used for a limited period. Just prior to APL2 installation, a study showed that of over eighty project libraries allocated, only seven were in use. Control of APL2 project libraries would now fall under the direct responsibility of an APL2 'system administrator' who would allocate the required datasets for project libraries. The use of VSAM datasets to store project libraries was selected. This would remove the requirements for separate datasets for each workspace in a library and the dummy entry in the catalog. The only drawback to this method is that users have to allocate required libraries as follows:

```
ALLOCFI(Wnnnn) DA(-project.APL2.LIBnnnn.WKSPS+) SHR REUSE
```

This method was also applied to the public libraries. Once all the required public libraries had been built, users were able to begin migration in earnest.

APL2 )MCOPY used in conjunction with the TRANSFER workspace became the standard migration tools. Users would initially copy the required workspace from their APL PDS to a VSAPL loadable dataset ('userid.APL.wsid') which could then be )MCOPY'd directly into APL2. When the required amendments to code had been made, issuing a )SAVE wsid would set up a 'userid.APL2.wsid' dataset which could then be copied back into the user's A.APL2 PDS.

By the time migration was completed on the first of the Bank's two TSO systems supporting APL, it was noticed that time required for workspace migration varied greatly upon the style of VSAPL code. Workspaces which had been well designed and written using a modular approach were found to be very easy to migrate. However workspaces which had direct access to APL124 contained in user written functions instead of using the standard FSM routines took considerably longer to migrate. At this stage, a difference between VSAPL and APL2 was noticed which was not mentioned in any of the IBM manuals. Under VSAPL, if a variable offered for sharing with an auxiliary processor did not contain a valid initial value, the default was assumed. Under APL2, if a variable shared with an AP contains an invalid value the variable is not shared successfully and will return a code of I whenever accessed. This problem was heightened by the shared variable offer returning a successful completion code so the error could only be spotted by detecting incorrect results being returned when using the AP.

Usage of APL2 during this initial migration period proved that APL2 was a relatively bug-free product. However the APL2/ISPF interface (one of the major enhancements) failed to work. At this stage it was pointed out that although a new ISPF release (2.1.1) had been installed for MVS/XA, the APL2/ISPF interface required ISPF release 2.1.2 which was not made clear in any documentation received from IBM. After receiving and installing ISPF 2.1.2 in record time, it was then discovered that no documentation regarding the interface could be found in either the ISPF or APL2 IBM manuals. Nor could IBM find any of the required manuals in their libraries, but IBM did find a solution. Along with the new release of ISPF, a load module was included on the ISPF load library (ISPAPAU) which contained the missing auxiliary processor AP317. Also included with ISPF 2.1.2 was an APL2 Transfer Format of a new workspace ISPFWS. This workspace contains a single function which allows access to ISPF from within APL2 by just passing an ISPF service string to the function. In addition to this, a new parameter had been added to ISPF with this new release which is required on the APL2 invocation step. Once this solution had been found, the migration of the second TSO system could begin.

The majority of APL users are based at this second centre. For many APL users who used a financial modelling function, the migration involved a straight )MCOPY of their existing workspace followed by a )COPY of the new version of the modelling workspace. For other more experienced APL users, the same problems that had occurred on the first system recurred. It was at this stage questions began to be raised regarding EDITOR 2, the new fullscreen editor. Although the new editor offers some interesting new facilities (most noticeably the ability to split the screen while editing functions) the removal of the command line seemed to confuse many users. Also the new editor places a large number of entries into the SYMBOLS table and can even be crashed with an index error if deleting and concatenating lines at the same time.

Some deterioration of performance was noted under MVS/XA. However due to environmental constraints it was not possible immediately to go single image at this stage. However after some further investigation one of the sources of the poor performance was found to be APL2 itself. When users began to use large APL2 workspaces which could not be self contained within their logon region size, storage 'above the line' was accessed as designed. However, this produced a large amount of paging and a single APL2 user was able to take over 60% of the systems paging capacity. In an effort to reduce this 'above the line' access, users were requested to limit their workspace sizes to be within their logon region size. The facility for accessing larger workspaces is still available, but if it is used the user will experience longer response times than other APL2 users. The request for the limiting of workspace size has had the desired effect by improving APL2 performance to a level comparable with VSAPL.

At the time of writing this article, migration has been completed on both centres. The migration has highlighted not only the advantages of well written modular code but has also allowed users to review their use of APL. The next step of the conversion to APL2 is for APL users to become aware of the new facilities offered by APL2 and use them to rewrite migrated workspaces which would benefit from use of ISPF, Nested Arrays, SQL/DB2, etc. These new facilities have also created interest in APL2 from users of other languages. In particular the simplicity of the APL2/ISPF interface has attracted attention from non-DP users using PL/1 who have shown that panel driven systems are much easier to write using APL2. The migration from VSAPL to APL2 took over five months to complete, but the process of learning to exploit APL2 to its full potential will take longer with significantly more emphasis placed on design.



## Running VS APL under TSO

By David Piper

In VECTOR 2.2 Adrian Smith raised some points about the difficulty of running VS APL under TSO rather than under VSPC. Whilst I agree that TSO leaves rather a lot to be desired in terms of user friendliness when running from the "READY" prompt, a lot can be done to improve things. TSO is a very flexible environment, in which just about anything can be achieved. The disappointing thing with TSO, is that the user-friendliness is left to the installation to create, and doesn't exist in the product itself. The following describes some directions which might be investigated in order to implement a better TSO environment.

### ISPF for Dialogue Management

One of the great benefits of using TSO is the availability of ISPF, with its flexible, and simple to use Dialogue Management Services. These facilities can be used to create very friendly dialogues running under TSO to conduct the user from the Broadcast Messages (where the READY prompt is normally first seen) into any other part of TSO. The dialogues can use a combination of full screen menus, data entry panels and tutorial panels. ISPF also allows an "expert mode" to allow very rapid navigation through menus for experienced users. ISPF can thus provide an environment suitable for use by both naive and experienced users.

### VS APL Session Manager

A piece of advice that can be freely offered is: *Don't use the APL Session Manager*. When this first became available, I was involved in performing some bench-marks on it. The initial figures derived were so appalling, that the bench-mark exercise was dropped, and the availability of SM withdrawn. The initial figures indicated that SM had the following impact:

- ★ A five-fold increase in CPU usage on screen bound activities
- ★ A doubling of CPU usage on CPU bound activities (no screen I/O involved except to invoke and terminate!)

Along with this impact on CPU, there was a considerable increase in Paging activity etc.

Since stopping the use of the SM, I have often wondered just how useful it is under TSO anyway, since any number of lines on the screen can be re-entered for execution simply by overtyping a single character. The lines entered in this way will be executed in the order they appear on the screen (this incidentally is a feature I miss under CMS, where it appears only INPUT lines can be retrieved for re-execution, and then only one at a time). The only benefit SM appears to offer is to "extend" the screen, so that data, already scrolled off the terminal can be recalled for execution. I think the overheads involved are hard to justify on this small benefit. If the APL SM is to be used, the colour of the output can be changed by using the HILIGHT operand of the VSAPL command. Since, by default, colour is linked to highlighting, specifying HILIGHT(OFF) when invoking APL will cause input from the user and output from the computer to be displayed in green.

### Logging On

Assuming a logon through VTAM, this can be rather daunting! However, if the implementation has a reasonable service level applied to TSO, one bash on the attention key (or PA1 if locally attached) is sufficient to return to the LOGON prompt. The logon process can then start again by typing:

```
LOGON userid
```

The full screen logon provided with TSO/E is another matter however, and seems to be full of Bugs (eg prompting for the correct password to be entered, but protecting the password field at the same time!). Most users seem to find the variety of options on the Full Screen logon overwhelming, so turn it off. The typical user should be able to logon in at most 3 steps:

- ★ VTAM logon to TSO – LOGON APPLID(XXXX)
- ★ TSO will then prompt for USERID
- ★ Once the USERID is validated, a PASSWORD will be prompted for.

If RACF is installed, a fourth stage will occur when the password expires – prompting for a new password. Other information may be prompted for if insufficient data has been added to SYS1.UADS. It should be possible to cater for users' needs simply by defining a logon procedure with a default region size, so no other information need be entered during LOGON. This process can be shortened further if a VTAM menu package is available e.g. NETMON or NET/MASTER. If correctly installed, these will allow the logon to be achieved in a single step; in fact NET/MASTER allows automatic logon to TSO to occur.

### Screen Erasure Problems.

Messages such as:

```
SCREEN ERASURE CAUSED BY ERROR-RECOVERY PROCEDURE
```

will originate from incorrectly customised control units in 99% of cases, and can be solved quite happily with some effort at correcting this type of error. This sort of message normally results from GDDM, or other screen manager, thinking the terminal can do something when in fact it can't.

### Sharing Workspaces

There is absolutely no need for the use of PROJECT libraries under TSO. If it is necessary to share a workspace (eg to fix a bug), this can be achieved quite happily by simply copying it (using the NONUM operand of the copy command). Alternatively, other users' workspaces can be accessed by changing the PROFILE PREFIX, and using )LOAD as though the WS were your own. If a security package such as RACF or ACF2 is installed, access must be granted before the )COPY or )LOAD is attempted.

The RACF GROUP concept can also be used to share a whole set of workspaces between a large number of users without recourse to the project library facility. The RACF Group allows a common PROFILE PREFIX to be shared amongst many user ids. Any workspace with this profile prefix will appear in the )LIB listing of all users with the same prefix. Generic access rules can be used to provide the same facility under ACF2. Whether a user is able to )LOAD or )SAVE a workspace depends on the access level given under RACF. This concept is more flexible than that of the project library, since a given user will be able to )SAVE some workspaces, only )LOAD others, and may have no access at all to the remainder.

### Clearing the Screen

One of the TSO manuals (I think its "OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor", GC28-0648) gives a listing of a very short assembler program which will clear the screen. By placing this in one of the TSO load libraries, it can be invoked as a TSO command by using )TSO or AP100.

### VSAM Record Locking

I am reliably informed that VSAM record locking works perfectly under TSO, for multiple region use. If multiple CPU access is envisioned, the Global Resource Serialisation feature is required.

### Printing

Printing to local 328X type printers under TSO is something of a problem. This type of printing is supported by two installed user programs (IUP) – programs not actively supported by IBM:

- ★ DSPRINT – in use on many sites.
- ★ JES328X – less widely used but more reliable.

Neither of these products will support the use of APL characters without a considerable rewrite. In addition, because these are IUPs, the command syntax may differ from site to site, although a common core syntax is normally implemented. GDDM also provides a print facility – ADMPRINT. This has been considerably enhanced in GDDM R4.0, to allow the use of printer control characters, thus with the use of two GDDM calls (FSLOG – AP126 code 603 and FSLOGC – AP126 code 606), all the standard printing facilities are available to APL, including the ability to print APL characters.

### User Numbers

Under TSO, there is no way supplied with APL of sharing APL data with other users. In fact APL thinks all users are number 1001. However, an IUP is available which allows data to be shared between MVS regions – MVS Inter Region Access Method (IUP 5796-ATX). This IUP provides general auxiliary processor support for two MVS regions to communicate. It also allows communication with an IMS control region, allowing APL to generate DL/1 calls to IMS databases.

### TSO Information

One advantage of VS APL under TSO is that a wide range of additional data about the operating environment is made available to APL through AP100. Extra pseudo-TSO commands have been provided to give the following types of information:

- ★ Current WSID, and save timestamp
- ★ Machine resource utilisation (CPU, DASD and other service units)
- ★ Current USERID, profile prefix
- ★ List of libraries available
- ★ List of workspaces in a given library
- ★ Releases of major pieces of software (MVS/TSO, APL, GDDM, HSM, RACF)
- ★ Terminal characteristics (Height, Depth)
- ★ Suppressing terminal output

These facilities are very useful to the APL programmer when interfacing to other systems, or when displaying information to a user on full screen panels.

### Conclusions

I hope the above points have gone some way to answering the criticisms levelled against TSO by Adrian Smith in the last issue of VECTOR. It is a pity that IBM have not put a little more effort into making TSO friendly to start with. On the other hand, the TSO environment is very general purpose, and by using add-on products (thus cunningly generating more profits for IBM), can usually be made to perform at the desired level. Once implemented, a TSO environment can be very stable indeed.

## Index to Advertisers

Ampere	108
APL People	29
British Airways	66
Cocking & Drury	40, 122
Dyadic Systems	100
HMW Programming Consultants	18
IBM (Sweden)	120
Inner Product	30
Mercia Software	68
MetaTechnics Systems Ltd.	80
MicroAPL	14
Mine of Information	39
VECTOR – back numbers	2

All queries regarding advertising in VECTOR should be made to the advertising editor, Steve Lyus, at the following address:

Group Management Services,  
Imperial Group plc.,  
East Street,  
Bedminster,  
Bristol,  
BS99 7JR.

Advertisements should be submitted in typeset, camera-ready A5 portrait format with a 20mm blank border. Illustrations should be black-and-white photographs or line drawings. Rates are £250 per page. A6 and A7 sizes (at £150 and £75 respectively) are available, subject to layout constraints.

**BRITISH APL ASSOCIATION****Membership Application Form**

Please read the membership information in the inside front cover of VECTOR before completing this form. Use photocopies of this form for multiple applications. The membership year runs from 1st May – 30th April.

Name: \_\_\_\_\_  
 Department: \_\_\_\_\_  
 Organisation: \_\_\_\_\_  
 Address Line 1: \_\_\_\_\_  
 Address Line 2: \_\_\_\_\_  
 Address Line 3: \_\_\_\_\_  
 Address Line 4: \_\_\_\_\_  
 Post or zip code: \_\_\_\_\_  
 Country: \_\_\_\_\_  
 Telephone Number: \_\_\_\_\_

Membership category applied for (tick one):	86/87	
Non-voting student membership (UK only) . . . . .	£ 5	
UK private membership . . . . .	£ 10	
Overseas private membership . . . . .	£ 18	\$ 27
Airmail supplement (not needed for Europe) . . . . .	£ 8	\$ 12
Corporate membership . . . . .	£ 85	
Corporate membership Overseas . . . . .	£140	\$210
Sustaining membership . . . . .	£360	

For student applicants:

Name of course: \_\_\_\_\_  
 Name and title of supervisor: \_\_\_\_\_  
 Signature of supervisor: \_\_\_\_\_

**PAYMENT**

Payment should be enclosed with membership applications in the form of a UK sterling cheque or postal order made payable to "The British APL Association". Corporate or sustaining member applicants should contact the Treasurer in advance if an invoice is required. Please enclose a stamped addressed envelope if you require a receipt.

Send the completed form to the Treasurer at this address:  
 Mel Chapman, 12 Garden Street, Stafford ST17 4BT, UK.

## The British APL Association

The British APL Association is a Specialist Group of the British Computer Society and a member of EuroAPL, an organisation supported by the Commission of the European Communities. It is administered by a Committee of eight officers who are elected by the vote of Association members at 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.

### 1985/86 Committee

Chairman:	Dick Bowman 01-634 7639	CEGB, 85 Park Street, London SE1.
Secretary:	Anthony Camacho 0727(56 from London)-60130	2 Blenheim Road, St. Albans, Herts AL1 4NR.
Treasurer:	Mel Chapman 0785-53511	12 Garden Street, Stafford, ST17 4BT
Activities:	Stan Wilkinson 01-286 7068	26 Leith Mansions Grantully Road London W9 1LQ.
Publicity:	Romilly Cocking 01-493 6172	Cocking & Drury Ltd. 16 Berkeley Street, London W1X 5AE.
Journal Editor:	David Preedy 01-541 1696	Metapraxis Ltd. Hanover House, Coombe Road Kingston KT2 7AH.
Education:	Dick Gray 0476-860483	Horseshoe House, Sproxton, Melton Mowbray, Leicestershire LE14 4QB
Technical:	David Ziemann 01-493 6172	Cocking & Drury Ltd., 16 Berkeley Street, London W1X 5AE

### Activities Working Group

Peter Donnelly	0420-87024
Steve Margolis	01-670 7959
Tim Perry	04626-77375
Roy Tallis	01-405 7841
Stan Wilkinson	01-286 7068

### Journal Working Group

Jonathan Barman	01-493 6172
Anthony Camacho	0727(56 from London)-60130
Steve Lyus	0272-666961
David Preedy	01-541 1696
Adrian Smith	0904-53071
David Ziemann	01-493 6172

## 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 and a member of EuroAPL, an organisation supported by the Commission of the European Communities. APL stands for "A Programming Language" — an interactive computer programming language noted for its elegance, conciseness and fast development speed. It is supported on many timesharing bureaux and on most mainframe, mini and micro 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.

#### APL People

17 Barton Street  
Bath, Avon.  
Tel. 0225-62601

#### APL Software Technology

14 Rosewood Avenue,  
Alveston, Bristol, BS12 2PP.  
Tel. 0454-415737

#### Cocking & Drury Ltd.

16 Berkeley Street, London W1X 5AE.  
Tel. 01-493 6172

#### Dyadic Systems Ltd.

Park House, The High Street, Alton, Hants.  
Tel. 0420 87024

#### Inner Product Ltd.

Eagle House, 73 Clapham Common Southside,  
London SW4 9DG.  
Tel. 01-673 3954

#### MetaTechnics Systems Ltd.

Unit 216, 62 Tritton Road,  
London SE21 8DE  
Tel: 01-670 7959

#### MicroAPL Limited

Unit 1F, Nine Elms Industrial Estate,  
87 Kirtling Street, London SW8 5BP.  
Tel. 01-622 0395

#### I.P. Sharp Associates

10 Dean Farrar Street,  
London SW1H 0DX.  
Tel. 01-222 7033



The British Computer Society, 13 Mansfield Street, London W1M 0BD.