OPENDOC OPENED UP • WORLD WIDE WEB • MOTIF'S BARE ESSENTIALS

July 1994

EXE

The Software Developers' Magazine

£3.20

ARE YOU BEING SERVED?

An end to PC databases

The NeXT step to object orientation

dBASE for Windows - exodus from DOS

CAREER DEVELOPMENT SECTION PAGE 83

TECHNOLOGY WITH NO USE-BY

DATE

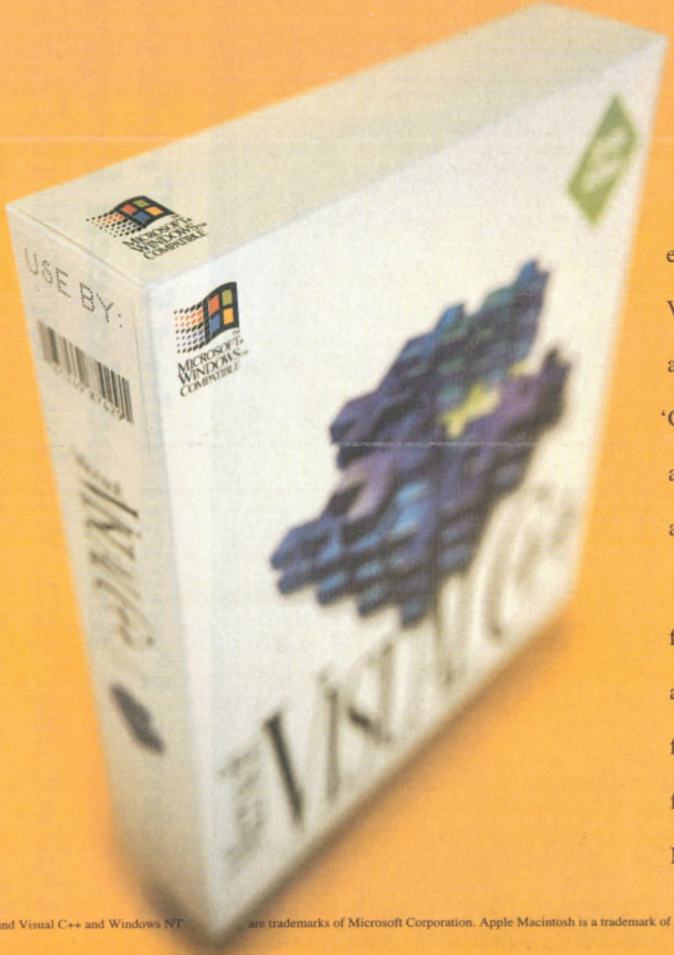
When it comes to developing tools, you've probably heard the term 'Future Proof' before, but rarely seen any evidence of it.

Until now.

Microsoft® Visual C++™ and the Microsoft Foundation Class Libraries (MFC) have been designed from the ground up to easily incorporate innovations as soon as they occur.

For example, the ability to move to newer versions by simply re-compiling code or to easily add support for the latest innovations such as OLE 2.0, has ensured that your investment in code will always be protected with Visual C++.

Future versions of Visual C++ are being designed to guarantee that you can always make the most out of new opportunities such as 32bit and multi-platform developing.



The forthcoming family of Visual C++ tools will enable existing MFC code to be easily migrated to any Microsoft Windows NT™ or Apple Macintosh™ platform, as well as taking advantage of upcoming operating system innovations such as 'Chicago' and 'Cairo'. We are also committed to making further advancements in developer productivity, quality of code generation and C++ language implementation.

With Visual C++'s track record, and our plans for the future, you can be sure that code developed today will never carry a use-by date. If you'd like to know more call **0734 270000** for the white paper on the future of Microsoft Visual C++.

Not to mention yours.

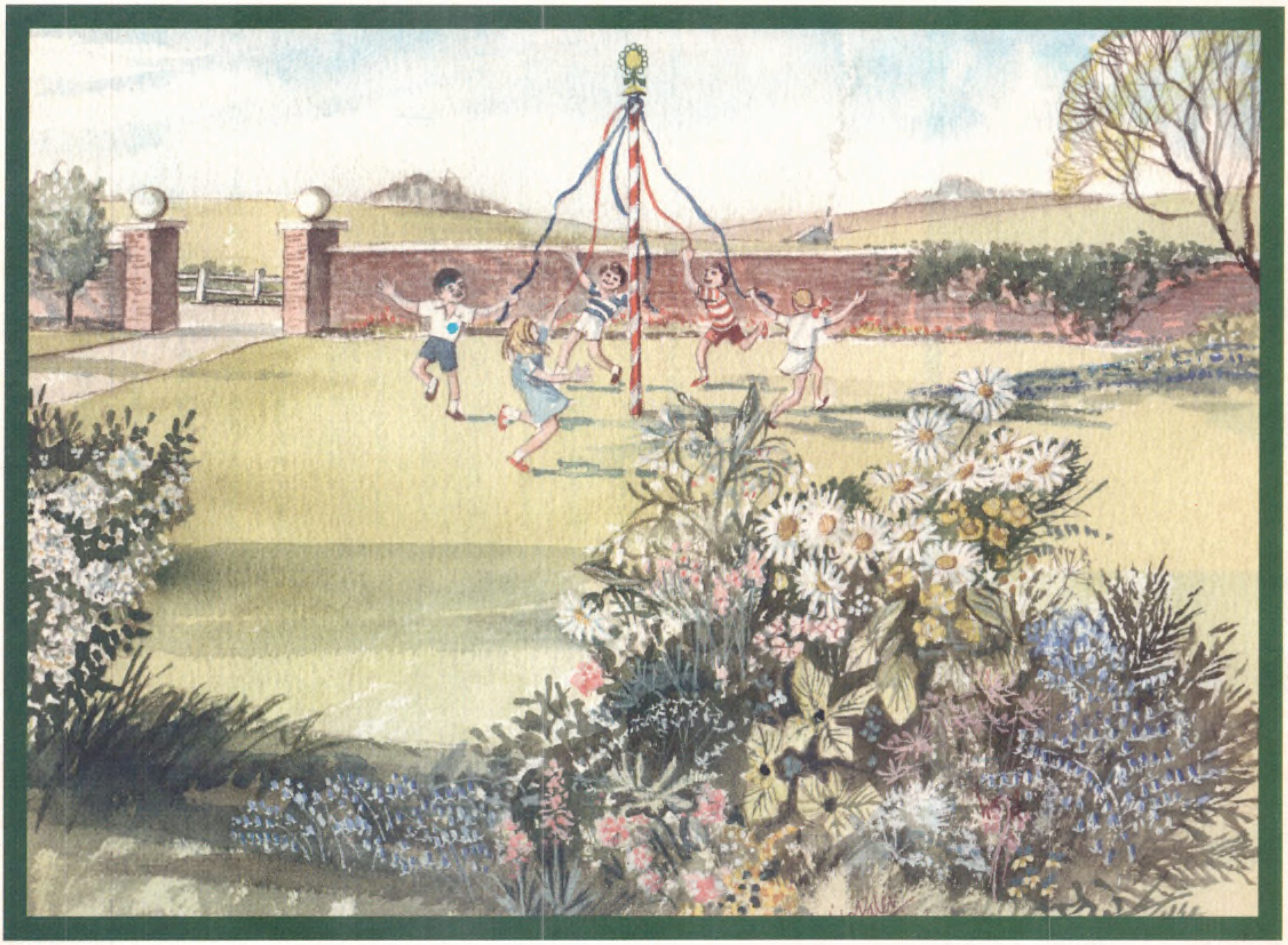
Microsoft®

Making it easier

➤ CIRCLE NO. 555

Microsoft is a registered trademark and Visual C++ and Windows NT are trademarks of Microsoft Corporation. Apple Macintosh is a trademark of Apple Computers, Inc. VCPE

KPWin++ from KNOWLEDGE GARDEN



Windows Programming for Professional Power Users and Experienced Programmers

Deliver intelligent Windows applications FAST in C++ with KPWin++

Ah! the exhilaration of finding a product that transcends the limitations of your existing basic development tools.

KnowledgePro for Windows, KPWin, is a fully object - oriented, event driven programming language supplying a rich, powerful and flexible development environment for Windows applications.

Its intuitive natural feel and unique combination of expert system, list processing, hypertext functions, GUI design tools and multimedia facilities provide an essential integration of modern programming power and productivity.

The advanced features of KPWin help reduce delivery cycles and improve the performance of experienced programmers whilst its short learning curve allows professionals, experts and power users to build systems themselves and exploit their own knowledge.

Plus! applications may be distributed royalty free to users.

KPWin++ is the first complete high level language to generate efficient, reliable, error free, object oriented C++ code for the entire application.

KPWin is now the chosen tool of developers in a wide range of commercial, academic and public service organisations, with delivery times being cut dramatically. An unparalleled ability to combine rule based systems with large amounts of data make it particularly suitable for rapid creation of multimedia applications, computer based learning and intelligent information systems.

TO ORDER OUR SPRING BOUQUET (and SAVE OVER £350) with KPWin++, newly enhanced screen designer, SQLKIT, Maths Toolkit and on-the-fly file compression tool, WRAP, for £895 + VAT, P&P and for details of our Consultancy, Lifetime Technical Support and Training services, contact John Bamford at:

Knowledge Garden Ltd.
Mountbatten House
Fairacres Business Park,
Windsor, Berks SL4 4LE, England
Tel: +44(0)753 833534/833112
Fax: +44(0)753 790755

All Debit, Credit and Charge Cards accepted

Dealer enquiries welcome

➤ CIRCLE NO. 556

EXE
The Software Developers' Magazine

CONTENTS

FEATURE

OBJECTIVE

LOOKING AHEAD The Apple Newton

SOFTWARE TOOLS & TECHNIQUES

OPEN SYSTEMS

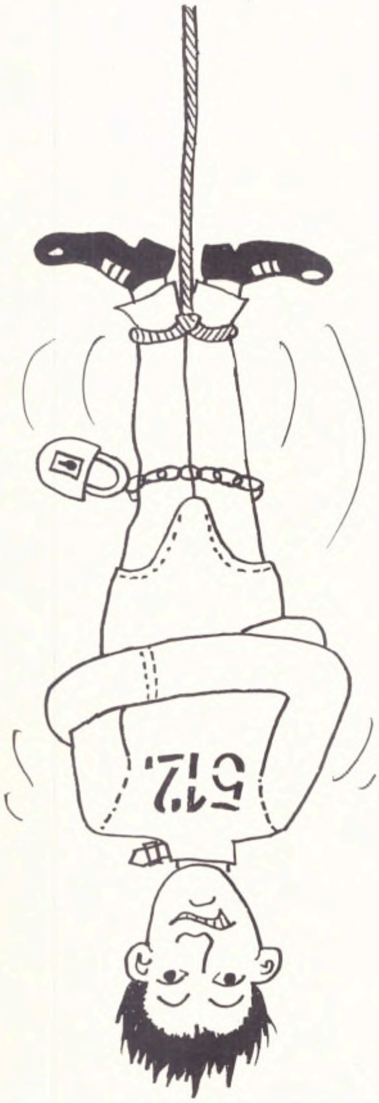
DBMS

DEVELOPERS' CORNER

COMMENT

Traumas of the profession

There was a time when programming was considered the alchemy of computing. Today it is considered a necessary evil.



I remember my first time. I must have been 12. A small group of us had gathered around a little black box with what was then known as a membrane keyboard. The box was wired up to a TV set. We were the privileged few, there to witness the school's first computer. We weren't allowed to touch the box or stab at the keyboard with our grubby little fingers; only to stand back in awe at the wonders that were to follow.

At the bottom of the screen was a rectangular slab which we later learned depicted a bat. A small ball bounced freely around the screen. When it made contact with the bat it would change direction and bounce to the other side. The position of the bat could be controlled with the keyboard. Each time the bat 'hit' the ball a beep would emanate from somewhere inside the black box. Soon we would be creating our own bats and balls that would foxtrot across the screen to our every command.

In those early days the joy of computers was that *we* could make them do things. From printing our names across the screen to playing with graphics and simple animation, it was fun to watch the computer obey the instructions we issued.

Today the 'fun' element of computing has now become 'rewarding'; those early instructions are now programs. Today we say it is a rewarding experience writing one's own program. But it is becoming increasingly difficult for one to do so. Shareware and the public domain are the last bastions of the solitary programmer.

Software is now developed by teams of individuals each working on a small, insignificant proportion of the whole project.

Why must computer programmers be forced into such an intellectual straitjacket? Surely they have minds of their own? Programming is not an exact science. It needs a broad perspective and a jab of lateral thinking in order for it to be done properly. There may be a hundred and one solutions to a given problem but the best one won't be found by leading the programmer by the hand to a solution which the designer thought would be most appropriate.

It all pivots on that sin bin of programming, the black box. The reasons, according to the perpetrators who force their fellow programmers into this Draconian measure, is one of security. Programmers are often told information on a 'need to know' basis. All they need to know is the inputs and the outputs to the black boxes which they have been asked to write. If there is ever such a condition as Programmers' Block, it is here that it would occur. The symptoms are similar to those of claustrophobia. The sufferer wants to climb out of his black box; he needs to know what is happening around him.

Keeping programmers in the dark is a recipe for delivering inefficient applications that don't make the best use of available resources. Imagine what it would be like if, when constructing a jigsaw puzzle, instead of starting with a picture and cutting it into jigsaw-shaped pieces, the pieces were cut first, then each was painted individually. So long as the artist had the original picture as reference, although difficult, it would be possible to paint the jigsaw pieces one at a time. In software development today, programmers are prevented from seeing the whole picture.

Give that person some space to breathe. Break loose those chains that imprison his thoughts. Let his mind envisage the complete picture, the whole project, so that he may return to his black box a wiser person, with a greater understanding of what must be achieved. Only then will he have attained the necessary insight in order to complete successfully the programming task set before him.

Cliff Saran

EXE: The Software Developers' Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, St Giles House, 50 Poland Street, London W1V 4AX.
EXE Advertising/Editorial/Production: 071 287 5000
Subscriptions: 071 287 5000 Ext 3142
Facsimile: 071 437 1350 ISSN: 0268-6872

Subscriptions. EXE is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (12 issues) in the UK, £56 for two years. European Community subscriptions cost £50 per annum, £80 for 2 years - Non-EC £70 1 year, £115 2 years. The magazine is published around the 1st of the month. To subscribe or if you have a subscription query, please call 071 287 5000 x3142, or write to The Subscriptions Manager, **EXE,** (address above). We can invoice your company if an official company order is provided, or take MasterCard, Access and VISA cards. Back issues are available at a cost of £3.50 each.

'A Subscription implies that this journal will be sent to the subscriber until one of the the three expires' - AG Macdonell.

Editorial. Editorial enquiries should be addressed to The Editor, EXE, (address alongside) or emailed to **cliffs@dotexe.demon.co.uk**. We welcome letters, opinions, suggestions and articles from our readers. Information contained in **EXE** is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.

Editor: Cliff Saran
Display Advertising Manager: Marc Warren
Recruitment Advertising Executive: Marc Green
Production Manager: Kate Adams **Sub Editor:** Melanie Welsh
Subscriptionsand Services Executive: Suzanne Chamberlaion
Circulation Manager: Michael Yetman **Publisher:** Sandra Inniss-Palmer
Front Cover Picture: Hulton Deutsch Collection Limited

VISUAL BASIC FOR WINDOWS

Visual Basic 3.0 for Win Std	£96
Visual Basic 3.0 for Win Prof	£230
Comms Lib/Win	£115
Network Library/Win	£80
PDQ Comm for Windows	£99
Agility/VB	£180
Btrieve for Windows	£365
CodeBasic 5.1	£129
Integra VDB for VB Std	£209
Integra VDB for VB Prof	£509
Q+E Multilink/VB 2.0	£270
Q+E ODBC Pack 1.1	£140
Raima Data Manager	£450
VB/ISAM MU	£150
VBtrv Control Pack	£140
Visual Controls for Btrieve	£180
Visual/db Network 3.0	£235
3-D Widgets	£89
ChartBuilder for VB	£75
Control Palette 2.0	£135
Custom Control Factory	£49
Data Widgets	£99
Designer Widgets	£99
Formula One	£220
HighEdit/Lite	£199
MS Visual Control Pack	£69
QuickPak Pro for Win 3.0	£145
Spread/VBX	£175
SpyWorks-VB	£120
Tab/VBX	£49
VB/Magic Controls	£120
VBAssist 3.0	£135
VBTools 4.0	£115

Accusoft Image Lib/VBX	£430
3D Graphic Tools	£130
Compression Plus for VBWin	£200
FXTools/VB	£105
Grid/VBX	£75
GUI Assist	£350
ImageKnife/VBX 1.3	£165
ImageMan/VB	£239
ImageStream/VBX	£220
Microhelp Muscle/Win	£125
PCX Toolkit for Windows	£155
QuickPak Scientific for Win	£105
Spellcheck	£49
TrueGrid Prof 2.1	£120
VB Compress Pro 3.0	£99
Visual Instrument Panel Controls	£150

PROGRAMMING tools

Ada
Basic
Comms
Custom Controls
Debuggers
Editors
Graphics
Linkers/Locaters
Modula-2
Pascal
Smalltalk
Version Control
Windows

Assemblers
C/C++
Cross Dev
Database
Dos Extenders
Fortran
GUI
Lisp
Multi-tasking
Prolog
SQL
Visual Programming
Xbase

We stock many items for which there is no space in these advertisements.

C & C++ FOR DOS

Comms

Async Prof for C/C++	£130
C Asynch Manager	£175
C Comms Toolkit	£215
COMM-DRV	£145
Essential Comms	£215
Greenleaf CommLib 4.0	£225
MagnaComm/DOS	£215
SilverComm "C" Asynch 4.0	£195

Database

Btree Filer for C Network	£170
Btrieve	£365
C-Index+ Pro	£349
c-tree Plus 6.0E	£565
CodeBase 5.1	£259
CodeBase++ 5.1	£259
D-ISAM	£325
d-tree	£490
DataBoss 3.5 (Special Offer)	£79
Greenleaf Database Library	£180
Paradox Engine & DFX 3.0	£125
POET SDK 2.0	£360
r-tree	£305
Raima Database Manager	£360
Raima Object Manager	£360
SoftFocus Btree/ISAM	£75
Topaz for C 4.0	£220
Watcom SQL Developer's	£260

Graphics

Accusoft Image Format Lib	£430
dGE 5.0	£205
Essential Graphics Kernel	£135
Fastgraph 3.0 (Ted Gruber)	£160
GraphiC 7.0	£345
GX Graphics 3.0	£155
HALO Professional 2.0	£245
MetaWINDOW-DOS 4.3	£150
MetaWINDOW-DOS/286 4.3	£210
MetaWINDOW-DOS/386 4.3	£270
PCX Toolkit 6.0	£155
PCXLab-DOS/286/386	£125
Real-Time Graphics & UIT	£225

GUI

4Sight	£300
Essential Graphics GUI	£99
graphics-MENU	£185
Menuet/CPP 2.0	£310
MEWEL/GUI Std 4.0	£390
object-Menu w/Source 2.7	£510
T-Windows	£240
Zinc 3.6 for DOS	£505

Printer

Baby Driver Kernel 3.0	£250
BGI Printer Driver Toolkit	£105
GX Printer 1.5	£240
SLATE	£260

Math & Scientific

C/Math Toolchest & Grafix	£45
Huge Virtual Array & NAT	£215
Linpack.h++ w/Source	£429
Math.h++ w/Source	£257
MIRACL	£39
Science & Eng Tools 8.0	£115

Memory Managers

Heap Expander 3.0	£130
MegaHeap	£115
MoreHeap 3.0	£135
PowerSTOR	£99
VMData for DOS	£665

Screen

C-Scape 4.0	£360
C/Windows Toolchest	£45
Greenleaf Datawindows 3.0	£225
MEWEL/TEXT Std 4.0	£390
Object Professional for C++	£165
Panel Plus II	£260
TCXL UI Dev Sys for DOS/Win	£140
Vermont Views Plus 4.0	£595

General & Systems Libraries

C Tools Plus/6.0	£135
CAD/CAM Dev Kit/2D Personal	£375
Crusher! with Source 2.0	£195
Hold Everything	£130
Interwork	£205
MTASK	£215
Multi-C w/Source	£39
Resident-C	£160
SpontaneousAssembly for C++	£95
TCOMP/Multi-Platform	£105
TTSR Ram Res Dev Sys 2.0	£105
TG-CAD Prof 5.0	£795
Tools.h++ w/Source 6.0	£259
TSRs and More	£115

Tools

C-DOC 5.0	£185
C-Vision for C 3.1	£95
Clear+ for C	£135
CodeCheck (Professional)	£495
MKS LEX & YACC	£240
PC-Lint for C	£89
PC-Lint for C/C++	£139
PCYACC Personal	£250
SourcePrint+ 5.0	£190

C & C++ COMPILIERS

Aztec Embedded C86	£290
Microsoft Visual C++ Std	£65
Microsoft Visual C++ 1.5	£299
Symantec C++ 6.1 Std	£85
TopSpeed C Prof	£169
TopSpeed C/C++ Prof	£215
Turbo C++ 3.0	£59
Turbo C++ Visual Ed. for Win	£84
Watcom C/C++16 9.5	£325
Borland C++ 4.0	£285
Borland C++ for OS/2	£250
High C/C++ 386 3.2	£555
High C/C++ for OS/2	£445
High C/C++ for Win NT	£445
MS Visual C++ 32-bit Edition	£300
Salford C/C++ Developer	£395
Salford C/C++ Prof Plus	£295
Symantec C++ 6.1 Prof	£310
Watcom C32 for DOS 9.5	£135
Watcom C/C++32 9.5	£375

GREY MATTER

Prigg Meadow, Ashburton
Devon TQ13 7DF

Prices do not include VAT or other local taxes but do include delivery in mainland UK . Please check prices at time of order as ads are prepared some weeks before publication. This page lists some products – call us for a complete price list. ORDER BY PHONE WITH YOUR CREDIT CARD

(0364) 654100

FAX: (0364) 654200

C & C++ FOR WINDOWS

COMM-DRV	£145
CrystalComm for Windows	£130
Greenleaf Comm++	£180
Greenleaf CommLib Pro	£355
MagnaComm/Windows	£280
Btrieve for Windows	£365
CB/ISAM MU for Win	£150
CodeBase 5.1	£259
CodeBase++ 5.1	£259
DataBoss for Windows	£410
Integra VDB for C++ Std	£209
Integra VDB for C++ Prof	£509
Paradox Engine & DFX 3.0	£125
POET SDK 2.0	£360
Q+E Database Library	£295
Raima Database Manager	£380
Raima Object Manager	£305
Spread/VBX++	£175
Watcom SQL for Windows	£260
Essential Graphics Chart	£300
GraphiC/Win 7.0	£370
Graphics Server SDK	£205
Windows Charting Tools	£300
Accusoft Image Format Lib/Win	£430
Ad Oculos (Image Analysis) 1.0	£325
Borland Visual Solutions Pack	£59
Canvas.h++	£340
Control Palette 2.0	£135
Data Entry Workshop	£130
DataTable	£249
Diamond Toolkit	£310
IC Image Control 1.1	£325
LEADTOOLS Prof	£520
M.4	£685
Magic Fields	£260
MS Visual Control Pack	£69
PCX Toolkit for Windows	£155
ProtoGen+	£220
SafeWin 3.0	£190
TX Text-Control 3.0	£1160
Win/Sys Library	£105
WindowsMaker Pro 5.0	£750
Zinc 3.6 for Windows & NT	£505

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Access 2.0	**£289**
Microsoft Delta	**£230**
Microsoft Fortran 5.1	**£138**
Microsoft Fortran Power Station	**£230**
MS FoxPro 2.6 Pro DOS	**£360**
MS FoxPro 2.6 Std DOS	**£87**
MS FoxPro 2.6 Pro Win	**£360**
MS FoxPro 2.6 Std Win	**£87**
MS Macro Assembler 6.11	**£110**
MS Multimedia Viewer Pub Kit	**£230**
MS Visual Basic for DOS Std	**£96**
MS Visual Basic for DOS Prof	**£230**
MS Visual Basic for Win Std	**£96**
MS Visual Basic for Win Prof	**£230**
Microsoft Visual C++ Std	**£65**
Microsoft Visual C++ 1.5	**£299**
Microsoft Visual C++ 32-bit	**£300**
Microsoft Windows 3.1	**£78**
Borland C++ 4.0	**£285**
Borland Pascal with Objects 7.0	**£250**
dBase IV 2.0	**£415**
dBase Compiler 2.0	**£245**
Paradox Engine & DFX 3.0	**£125**
Turbo C++ 3.0	**£59**
Turbo C++ Visual Ed. for Win	**£84**
Turbo Pascal 7.0	**£84**
Turbo Pascal for Windows 1.5	**£99**
Paradox for DOS 4.5	**£405**

WITH FULL TECHNICAL SUPPORT

News & Views

The most significant news of the month must be advance details of the forthcoming Microsoft Visual C++ 2.0. Due in August, it will be 32-bit only and there will be versions for Intel, MIPS and Alpha, plus an Intel-hosted cross compiler for the Macintosh (both 680x0 and PowerMac). It will have templates & exceptions (at last!), improved C++ optimisations, faster compilation, incremental linking, MFC 3.0 (the 32-bit version of MFC 2.5), and a completely redesigned and integrated IDE. No pricing or packaging details yet - call or fax us if you want to be on the mailing list.

The Borland DOS Power Pack is an essential purchase for any Borland C++ 4.02 programmer writing large DOS apps. Both 16 & 32-bit royalty-free extenders with Virtual Memory, 32-bit BGI, Turbo Vision 2.0, 16 & 32-bit DLL's (compatible with Windows/NT DLL's). The 4.02 update is available directly from Borland for a nominal media charge - this also supports NT-hosted development. The DOS Power Pack is available at a special price of £75 until the end of August. Call us now to order your copy.

If you want to add a macro language to your application but cannot afford ProMacro, why not check out Cypress Enable. It follows Visual Basic syntax, is implemented as a DLL/OLE Automation Server, and will allow your end users to create macros that can launch and manipulate other applications via OLE Automation, access any DLL and automate complex or repetitive tasks. There is also an OLE editor which allows them to edit, debug, compile and test their macros. All this is royalty-free and only costs £855. Call us for more details.

Are you a power user who can't decide between UNIX V.4 and NT? Get the best of both worlds with the Portage Base System & SDK. This is a port of UNIX V.4 source code to NT, including the Korn & C Shells and over 100 utilities. There is a Windows interface to manage multiple UNIX shells, you can run UNIX commands from an NT prompt and NT commands from UNIX shells, with complete on-line manual pages in Windows Help format. The SDK adds over 20 more utilities (yacc, lex, make, SCCS, etc), over 400 UNIX V.4 system calls as well as curses, and allows both Win32 and UNIX calls in a single program. Needs Visual C++ 32-bit Edition. Call for more details - only £495 for the combination. X & Motif and TCP/IP & NFS are planned for 3Q.

VISUAL BASIC FOR DOS

MS Visual Basic for DOS Std	£96
MS Visual Basic for DOS Prof	£230
Comms Lib/DOS	£115
Network Library/DOS	£80
PDQ Comm	£65
Btrieve 5.1	£365
CodeBasic 5.1	£129
db/Lib Prof 3.0	£195
E-Tree Plus	£165
VB/ISAM MU	£150
Elltech's GUI Toolkit	£125
Fastgraph	£160
Graphics QuickScreen	£105
Graphics Workshop	£105
GraphPak Pro	£99
GX Graphics 3.0	£155
Compression Plus	£130
Compression Workshop	£110
Microhelp Muscle/DOS	£155
PDQ	£110
Printer Plus	£185
ProBas 5.6	£169
QuickPak Prof 4.1	£145
QuickPak Scientific 3.0	£105
QuickScreen 4.0	£105
Science & Eng Tools 8.0	£80
TSRific	£80

BASIC

Compilers & Interpreters	
BBC BASIC-86 Plus	£75
CA-Realizer 2.0 (Win)	£190
GFA-BASIC for DOS	£80
GFA-BASIC for Windows	£80
GFA-BASIC for Win Compiler	£54
PowerBASIC 3.0C	£110
TrueBASIC 3.0	£88

➤ CIRCLE NO. 557

Making the most of IT

The NCC will be running a two-day conference on July 5th and 6th at the Park Lane Hotel in London. The focus of the conference will be 'empowering the organisation - more than rearranging the pieces.' It will address the problems of incorporating new technologies into business strategies. There will be sessions on future technologies, managing change, software process re-engineering and IT management. The speakers will include Ted Keller, head of Onboard Systems for the NASA Space Shuttle and John Hawkins, president of Philips Media Distribution. Registration details can be obtained from Carol Wright at the NCC (061 2286333).

Apple/Symantec licensing

Through a cross-licensing agreement Apple intends to licence the Symantec C++ front-end for use in its own Macintosh Programmers' Workbench compiler for the PowerPC. Symantec will be licensing Apple PowerPC code generator, linker and other tools for the PowerMac version of Symantec C++. The Symantec C++ 7.01 compiler for the 68K supports the Code Fragment Manager needed for OpenDoc development. Symantec will provide enhancements such as the Visual Architect interface builder which allows for rapid development of OpenDoc parts. As an additional aid to developers there will be support for OpenDoc containers in the Think Class Library (TCL).

Open awards

On the 6th June SAP UK became the 1993 Open Software Supplier of the Year. It also won the award for best Business Application Software. The award was made by X/Open and sponsored by ICL. SAP was judged by its work in helping clients move to open systems and by 'leading and facilitating the practical implementation of open systems.' Other winners included Sybase and Westmount who won in the categories of system software and development software.

Dos and Windows on Alpha

Digital and Insignia have signed a joint marketing agreement which should help ease the transition of moving to the DEC Alpha. Insignia's SoftWindows will be sold and marketed by both companies. A version for Digital Unix will be available some time in the summer. A version for OpenVMS is expected in autumn.

Upsize at Borlando

The 5th annual Borland International Conference (BIC) in Orlando last month was kicked off with a keynote from Philippe Kahn. First the statistics. A customer base of 15 million; the 4th largest PC software company and 900 of the Fortune 1000 companies use Borland tools. The recent sell-off of Quattro Pro was referred to as 'entrusting Novell'. The turnover for 93/94 was $290 million excluding the sale of QP and Borland Office Suite. Once the Novell deal is through it will have more than $100 million in cash.

The theme was very much databases. To be more precise it all came down, or should I say end up as upsizing. Borland is upsizing. King of the PC database is transmitting the message that client/server is the way forward. We already knew that... But the question is how will Borland get us there.

That's where the new VP of products, Ken Gardner, maded is debut. Ken is the 'upsizing guy' according to Philippe Kahn, there to increase the presence of the Borland brand within the enterprise. Ken used to be CEO of ReportSmith and has come on board along with the recent acquisition of the company.

On the client side there was dBASE for Windows which *EXE* has been taking a close look at this month, plus a preview of a new development tool for client/server. The server product was, of course Interbase, or Interbase 4.0 to be precise. It's been promised for months. Among the new features, expect to see a small footprint of less than 7.5 MB and a simplified installation requiring only two diskettes and which can be loaded onto NT in 2.5 minutes. If you read our review of Intebase 3.3 last month you'll be glad to hear that Borland has provides an interactive SQL editor and a database administration untily. The dialect of SQL used by the product will be based on SQL '92.

Delphi, for all your oracles

Borland is planning to blaze into the 4 GL client/server tools market later this year with the launch of a new development tool. Traditionally the tools that have been used for development have fallen under the guise of the 4 GL. While offering the facility of rapid application development through the use of interactive GUI environment these development tools suffer from one major flaw. That being the handicap of an interpreted language. Today's leading tools such as SQL Windows and PowerBuilder incur incredible performance penalties when applied to large client/server systems development.

Borland sees the opportunity to take advantage of this problem. What client/server developers need is a tool which offers rapid development while delivering the raw performance expected of a traditional 3 GL such as C++ or Pascal. The solution which Borland has proposed has been codenamed Delphi '95. The first public showing of this technology was given at the 5th Borland International Conference (BIC) in Orlando in June.

There is expected to be two versions of the Delphi product. Borland hinted that it would be bundling it with a Windows version of Interbase 4.0 and the ReportSmith report generator (recently acquired by Borland). There will also be a cut-down version of the product. At the time of going to press Borland had not announced expected shipping dates for Delphi, pricing or the final product name. However *EXE* will keep you up to date on developements as and when they occur.

Calling the developers

Software Developers Forum is a new event which will be taking place at Sandown Park between 8th and 9th February 1995. There will be six vendor seminar streams and daily debates attached to each stream. There will also be an exhibition. The seminars will include discussions on the building, installation and management of client/server applications; RAD, CAST, Object, re-engineering and legacy systems. Furthermore a conference, organised by Visual (Vendor Independent Software Users Association), will be running alongside the seminars. This will provide case studies from end users of particular tools.

The organiser hopes the format of the event will allow visitors to glean a detailed presentation of a specific product then obtain a real user's perspective before going to the exhibition for a demonstration of the software. As befitting the UK's leading software development magazine, *EXE* is a co-sponsor of the Software Developers' Forum.

FOR DYNAMIC DATABASE REPORTING

Attention Paradox, dBASE, Access, and FoxPro owners!

PC Version
£69.95 RRP +VAT
SQL Version
£199.95 RRP +VAT

Borland ReportSmith™ for Windows

The easiest way to spectacular reports!

ReportSmith is a database reporting and query tool that is the perfect companion to your PC and SQL data management software. Its WYSIWYG "live" data approach to reporting provides users with a drag and drop, visual interface that makes report creation fast and easy!

See your data while you work

With ReportSmith, you work directly and interactively with the actual **"live" data** on screen to get the results you want – *fast!* Discover how easy it is to design the reports you want when what you see is *really* what you get.

Great-looking reports for everyone

No matter what your level of database expertise, you can instantly build attractive **columnar, crosstab, label** and **form** reports. ReportSmith includes default **report templates** as well as **predefined styles** – simply click on the report type and style you want for an instant polished report.

Complete compatibility with ALL popular databases

ReportSmith gives you *all* the reporting capabilities you expect in a database reporting tool: faster report creation, greater power, and more flexibility. ReportSmith works with all popular PC databases, including dBASE®, Paradox®, Btrieve, Access and FoxPro, as well as SQL databases, including Oracle, Sybase/MS SQL Server, Gupta SQLBase, Ingres, and others.

Powerful customisation and extensibility

ReportSmith gives users and developers the tools they need to build advanced reports for applications. Use ReportBasic, our Visual Basic-like **macro language**, to customise and extend your reporting applications.

Borland

Easily integrate ReportSmith with other applications through use of **OLE 2.0** and **DDE** support.

SEE YOUR STOCKIST TODAY!

For further information or details of your nearest stockist complete the coupon or call

0800 212727

Please send me the FREE information pack.
Ref: AS/9426/RS

Name ____________________

Position ____________________

Company ____________________

Address ____________________

____

____

____

____

Postcode ________ Tel. ____________

Please return to:
Borland International (UK) Limited, FREEPOST,
PO Box 9, Stroud, Gloucestershire, GL6 8BR.

➤ CIRCLE NO. 558

Borland International (UK) Limited, 8 Pavilions, Ruscombe Business Park, Twyford, Berkshire RG10 9NN Tel 0734 321150. Dealer prices may vary.

Now compiling SQLWindows

Gupta's latest offering is SQLWindows 5.0 which boasts a 4 GL compiler that generates compiled C code. The architecture has been enhanced with QuickObjects that provides reusable components for building object oriented applications. There are three kinds of QuickObject. Data Sources provide access to SQL and non SQL data. Visualisers control the presentation of data and Commanders allow users to enter commands to manipulate data directly. New QuickObjects can be created and added to the SQLWindows environment. The Corporate Edition includes the SQLWindows Compiler, SQL Console, Gupta Open Repository and team programming facilities. The cost is £3,395. Gupta is on 0628 478333.

Symantec Client/Server

July must surely be the month of the client/server... Borland, Gupta and now Symantec is at it. Yes, Symantec has an all new Windows-based client/server development tool called Enterprise Developer which is based on Symantec's SCALE (Scaleable Architecture for Large Enterprises). Applications are built using the Rapid Forms Builder and visual editors. The built-in 4 GL is SCALEscript an object oriented programming language that provides control statements, procedures and database statements. The environment includes an editor and debugger. Integration with external tools, objects and programs is possible through the Open Object Architecture which supports OLE, DDE and DLLs. In fact, links to PVCS and LBMS System Engineer are included. Symantec is on 0628 592222.

More objects in Pascal

Borland isn't the only company moulding Pascal as an OOP. Cabot Software has developed OOP extensions for UCSD Pascal which allows DOS and Turbo Pascal programs to be used on OS/2 and Unix. The price of UCSD/Pascal OP is £180 from Cabot Software (0272 586644).

Eiffel snippet

For $49.95 you can get a copy of Personal Eiffel for Windows from Interactive Software Engineering (ISE). The product is based on ISE's Melting ICE technology which provides incremental compilation. It includes object-oriented tools for class documentation and browsing. There are also 'several hundred' pre-compiled libraries of classes including EiffelBase for data structure and EiffelLex and EiffelParse for language analysis. ISE can be contacted in the US on 0101 805 6851006.

Workgroups can SQL

At the PowerSoft International User Meeting and Training Conference in May, Watcom revealed a new addition to its SQL database product family. Watcom SQL Network Server for Windows is a multi-user client/server databases aimed at workgroup environment such as Windows for Workgroups, Lantastic and Personal NetWare which compliments the company's SQL server databases for OS/2, NetWare and NT.

Applications written for Watcom SQL are scaleable from standalone, single user through to multi-user networks. ODBC is used as the API to the engine, allowing developers to choose from a range of front-end development tools. Watcom has provided what it calls 'SQL-rich cursor capabilities' which gives ODBC the boost it needs to perform like a native API.

The database engine provides many of the facilities offered by SQL servers. Concurrency and transaction processing is achieved by using row-level locking; symmetric multithreading of server requests; checkpoints; rollback and forward transaction logs and an ANSI transaction model with four isolation levels. Security and reliability is achieved through online backups; referential and entity integrity and data encryption. The API supports bidirectional scrollable, updateable cursors; updateable multi-table views; BLOBs, self-tuning; cost-based query optimisation; database compression and international character set support. Finally, it offers interoperability through Level 2 ODBC and NetBios or NetWare IPX network communications. Watcom is a subsidiary of PowerSoft. Further information can be obtained by phoning 0628 345000.

Leaner on the resources

Microsoft has given a preview of Windows NT 3.5, the 'Daytona' release of NT. One of the key improvements has been in the area of system requirements. The unrealistic 16 MB desktop has gone down by 4 MB to 12 MB. A new feature is the inclusion of the industry standard Open GL API graphics library developed by Silicon Graphics. Microsoft has also addressed the criticism it provoked for not having sufficient connectivity services built into the Windows NT Workstation package. NT 3.5 provides access to files, directories and printers on NetWare servers.

There is now support for multiple VDMs which allow pre-emptive multitasking of 16-bit Windows and DOS applications in protected memory. DDE and OLE links can be maintained between individual VDMs. In addition, OLE is fully supported between the 16- and the 32-bit environments. So in-place editing, OLE automation and drag and drop between applications can occur between 16- and 32-bit applications.

The Advanced Server version has also been improved. 'We have listened to our customers,' commented Julie Cox, business systems group marketing manager, Microsoft Ltd, 'and have made Windows NT 3.5 Server up to twice as fast, using less memory.' The TCP/IP layer has been rewritten to provide improved connectivity. Microsoft also claims to provide better interoperability with Unix and NetWare.

Shipping of Windows NT 3.5 will be in late summer. Customers are recommended to contact their nearest authorised Microsoft Solution Provider. Details can be obtained by phoning 0734 270000.

Fractal SDK

ColorBox pro is an SDK for the latest fractal still image compression from Iterated Systems. It offers XGA screen resolutions of 1280 by 1024 in 24-bit colour and 8-bit greyscales. According to Iterated Systems compression of a 3.9 MB file down to 30 KB is possible 'without appreciable loss of image quality'.

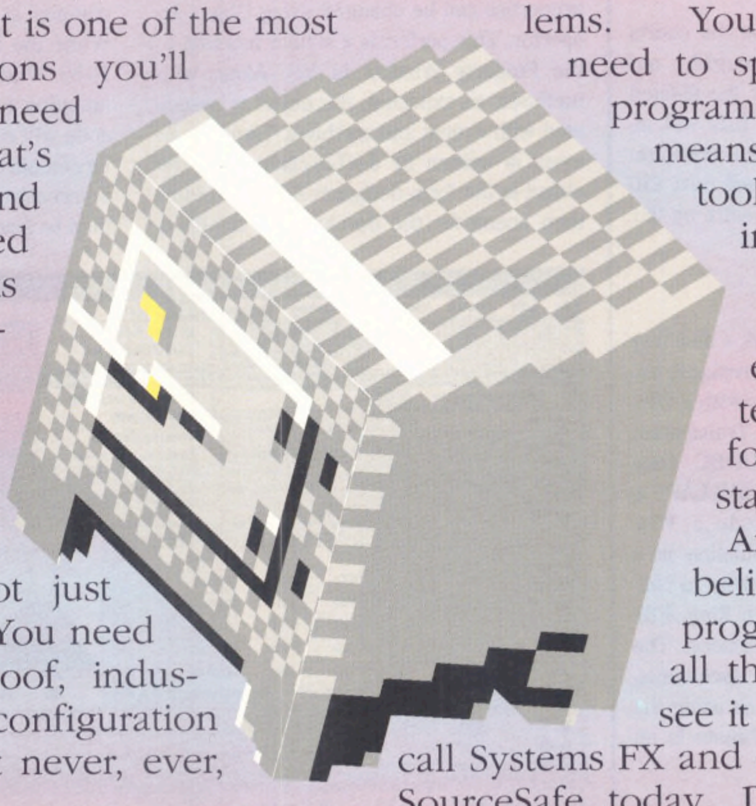
The FTC-III compression acceleration board is include in the SDK which the manufacturer says will offer faster compression by searching for the optimum set of fractal codes in a given image. The complete package costs £2,495. The price includes full source code for Images Incorporated, a Windows-based DTP program which includes sample fractal video clips.

This may be used as a reference for builing compression and decompression applications. Iterated Systems is on 0734 880261.

DISCOVER SOURCESAFE

If source code is your company's most valuable asset, then source code management is one of the most important decisions you'll ever make. You need a system that's powerful and proven. You need a system that links all your development platforms. You need a system that tracks entire programs and projects, not just individual files. You need airtight, bullet-proof, industrial strength configuration management that never, ever, ever loses a file.

What you don't need is hassles, long installation times, difficult learning curves, expensive training courses, and customer support problems. Your programmers need to spend their time programming, which means you need a tool that saves time instead of wasting it. You need modern, easy to use interfaces that conform to industry standards.



And before you believe that one program can offer all that, you need to see it for yourself. So call Systems FX and order a demo of SourceSafe, today. Discover how far version control has come. Discover SourceSafe.

SourceSafe™

project oriented version control

➤ CIRCLE NO. 559

SYSTEMS FX

...Quality Tools for GUI & Client/Server Developers

The Gables Market Square Princes Risborough Bucks HP27 0AN Tel 0844 275175 Fax 0844 343615

SOAPBOX

The greatest burden to the success, or otherwise, of OS/2 is that it is being marketed by IBM. **Dominic Conner** explains why...

At the Windows show I had a meeting with Alan Pullin, a director of the largest software business in the world, indeed bigger than the next four put together. Alas he did not work for Microsoft, but its erstwhile ally, IBM. Not long before I had seen a picture of a South African nuclear reactor, which was arguably the single least politically correct object made by man. The link is of course OS/2, this has the sort of image normally reserved for military regimes that brutalise their populations and have coups on a weekly basis.

Indeed, typical OS/2 victims (or should I say survivors?) are pictured as an array of hapless grey minions in multinationals coerced into the use of a product that combined the user interface of a mainframe with the reliability of British Rail and the openness of a DEC operating system. This view is reinforced by some people's experience. One IBMer described to me the 'high' performance file system in OS/2 1.X as 'mad, bad and dangerous to know'. IBM's marketing is legendary in its wretchedness; OS/2 2.0 was advertised (by IBM) as an elephant doing tricks. In as much as the upcoming ad campaign means anything at all it seems to imply that it is a distorted pig. In the USA IBM regularly runs ads knocking NT. But Microsoft does not retaliate, preferring instead to leave OS/2's downfall to IBM. And yet...

For every copy of NT sold, half a dozen OS/2's break free from the clutches of IBM marketing and attach themselves to one of the hapless users mentioned above. If you ask IBM how many, you get told that 5 million OS/2s have been sold worldwide, but that the UK figure is confidential. Actually the last figure I acquired was 237,000 OS/2 2.X's in the UK. But then two IBMers did have to risk their jobs to get this figure out of North Harbour into my clammy paws.

OS/2 has some merits. You can run multiple copies of Windows, so that you don't have to run out of GDI resources. A stray pointer is much less likely to bring the system down than Windows. It runs a user interface that's suspiciously close to that of Chicago, with threads, drag and drop and object orientation - but you can buy it now. By the time you read this the '4 MB' OS/2 will be shipping. This will mean it will work on the majority of desktops today. I seem to recall that 4 MB was the minimum requirements at the launch of OS/2 2.0 - oh please, get real!

Even now, when the operating system has had time to mature, IBM can still screw up its credibility. Especially when it chooses to use taboo terms in its publicity material. The worst of these is surely the phrase 'supports more non IBM hardware'. Well that's good news. So my spreadsheet can cope with some numbers that aren't odd, as well as certain fractions, provided I download the driver from CIX.

IBM tends to announce each release of its products in a manner that suggests it is its last. Quite commendably it avoids vapourware, but at the price of not telling people what's going on. Microsoft is replacing IBM as the purveyor of orthodoxy, Several major development shops I have visited have no software other than that which is available from Microsoft. Occasionally I have found situations where OS/2 would appear to be the only rational choice for a task: the suggestion of this can just cause blank stares at such sites.

IBM has stolidly trudged on with OS/2, fixing here, optimising here, adding a new chipset, in the hope than the 'next big heave' will change the situation. The hope still lingers that the 'killer app' will appear and drive OS/2 sales to some critical mass. Such Great Blue Hopes have included Lotus Notes, Application Manager, Multimedia and Video conferencing. Each has chipped away, but not broken through. There are now, at last full sets of native OS/2 software from Lotus, Symantec, Sybase, Rogue Wave, Borland, Computer Associates, and Corel. But since so many people are wedded to Excel and Word, there is little prospect of them moving until someone forces them to.

The IBM and Stacker experiences show the problems of having a strategic relationship with Microsoft, but IBM has little in the way of allies. There are several reasons for this, not least the fact that while it is one thing to do business with people you don't approve of, if you believe them to be doomed, why bother? The other main reason is IBM's size, not especially the slowing down of decision making, but the fact that it is mortally afraid of offending anyone. Considering that its hardware division is one of Microsoft's biggest customers, that NT is going onto its PowerPCs *before* OS/2, and that it also has been selling Lotus Notes, Netware, Stacker, Wordperfect and Borland C++. The task of not offending any of that group would challenge the most articulate of culturally aware Californian spokespersons. And yet...

Browse the recruitment pages of EXE and the weeklies and you see corporates recruiting OS/2 skills at least as fast as for NT. Perhaps they just want a devil's advocate? I doubt it. ■

Dominic Connor can be contacted on email as `mod@cix.compulink.co.uk`.

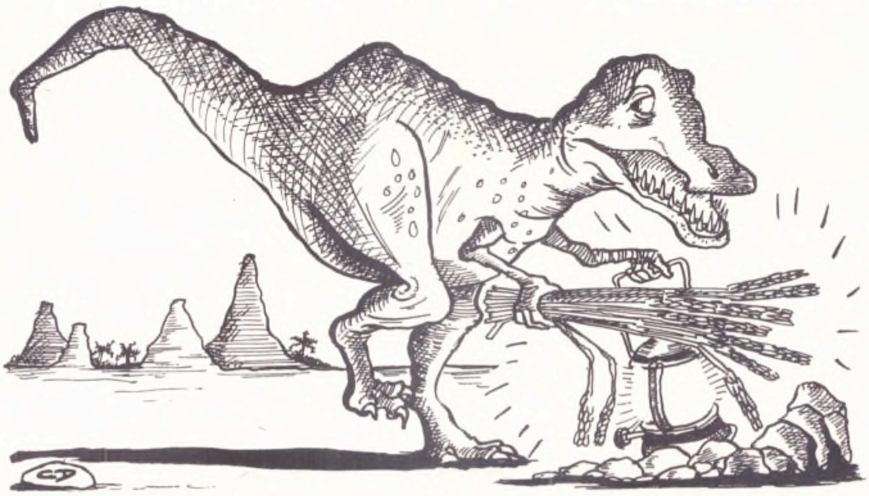
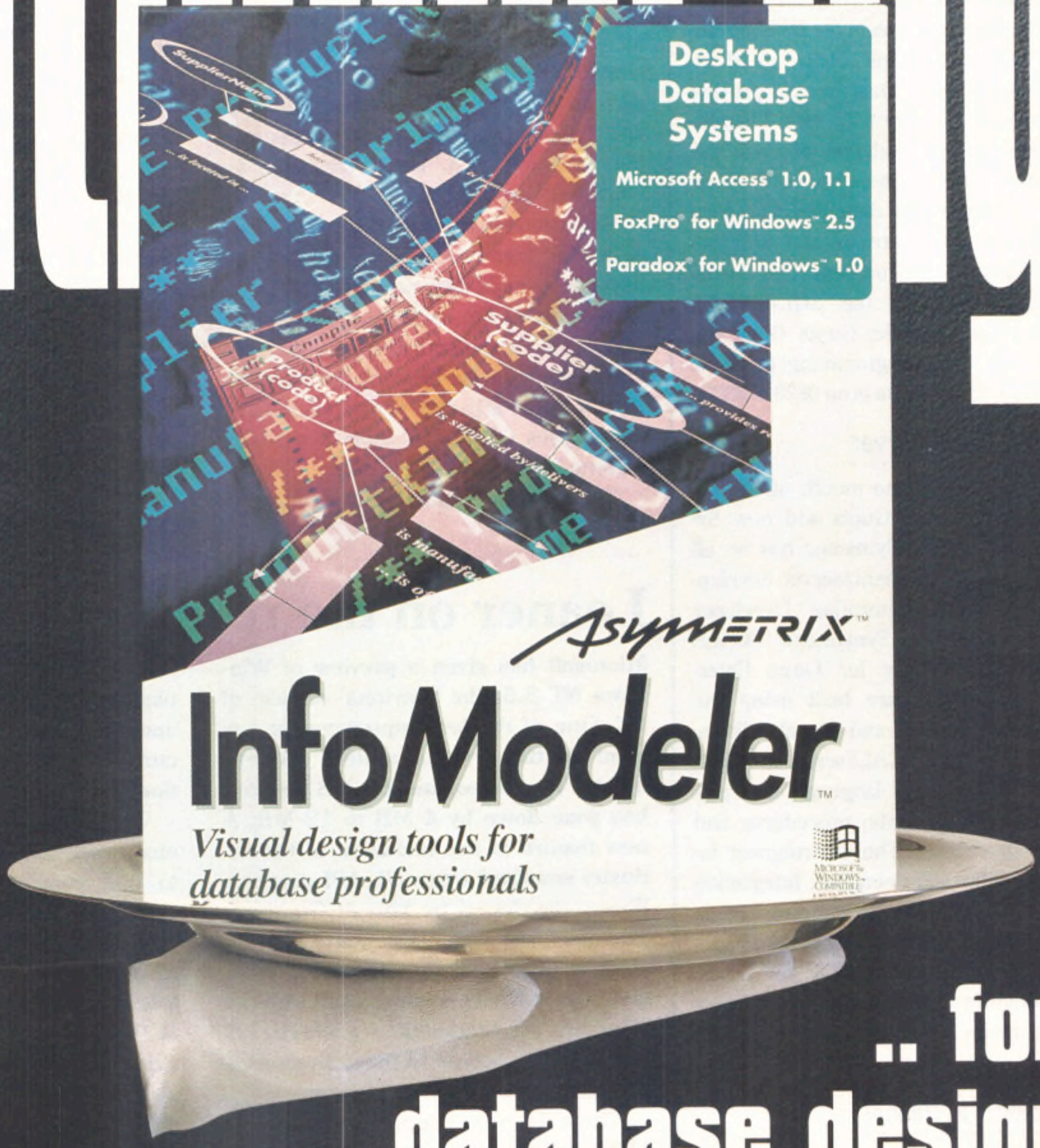


Fig. 1 - The IBM-O-Saurus hiding its light under a Bushel....

Introducing...


Desktop Database Systems
Microsoft Access® 1.0, 1.1
FoxPro® for Windows™ 2.5
Paradox® for Windows™ 1.0
ASYMETRIX™
InfoModeler™
Visual design tools for database professionals
Microsoft Windows Compatible

.. for database design that gets it right.

Wouldn't it be better to get it right first time? If you design, maintain or support information systems, here's exciting news. Now there is a Windows based visual information modeling tool for client/server and desktop databases – Info*Modeler*. It's a revolutionary database design system which helps you get the correct design, first time, every time. Info*Modeler* lets you model and review your design using English facts and examples, so it's readily understandable to everyone associated with your project. And because your design is verified, you'll produce applications which are easier and much less costly to maintain. What's more, Info*Modeler* automatically generates the database at the touch of a button. So there is no need to transfer design information and structure to a different software package, saving time and helping eliminate re-keying errors. As well as desktop databases Info*Modeler* works with Oracle7, Microsoft SQL Server and Sybase and will also support IBM DB/2, Ingres and other relational databases systems.

InfoModeler is distributed in the UK by ICS Solutions Ltd, who provide a complete range of training, support and professional services for InfoModeler users and resellers.
For further information call 0256 469460

ICS

Call ICS for the latest prices and special offers.

MasterCard VISA

To find out how Info*Modeler* can help you send off this coupon.

□ Rush me more information
□ Supply me a free demo disk
□ Supply me an evaluation system (£ 24,90*)

Name : ____________________

Company : ____________________

Address : ____________________

____

Phone : ____________________

Return to ICS Solutions Tempus Business Centre, Kingsclere Road, Basingstoke, Hampshire RG21 2XG Fax 0256 842362

* including VAT and shipping

► CIRCLE NO. 560

The PC database - dead or alive?

Ian Murphy predicts that while the move to client/server architectures is on the increase, PC database will continue to play an important role.

All the prophets proclaiming the end of the PC database have little experience in the history of computing

It is a fact of history that as soon as something is perceived as being better comes along, the original is relegated to the been there, seen it, done it, get rid of it, pile. In our industry this happens several times a year but rarely with any accuracy or realism. Most of the people making such profound statements either have little knowledge of what is going on or they are locked into a particular viewpoint and have forgotten objectivity.

From hierarchical beginnings

As we are talking about databases here, let's look at the previous claims in this area. The first real database structure we had was the *Hierarchical model* which arose out of necessity. We had started to store large amounts of data on our fledgling computer systems: if we were to bring about the brave new world we had to be able to get at the data. The flaws in this model are well documented and researched. They led to a search for a successor which would overcome some of the major problems, in particular access and duplication.

The solution was the *Network model*. Although it was also arrived at out of necessity, there was some planning in the work done by the CODASYL committee to approve a design and degree of functionality. By now we were in the swinging 60's and the world of data processing was yet to encounter its industrial revolution.

As the decade closed, people started to question the way the Network model worked and discussions were taking place on what exactly was a DBMS (Database Management System) rather than how the data was physically stored. During this period the prophets of doom were running around saying that the Hierarchical model had met its end and everyone should be turning to Network models. It didn't happen, however. Instead vast sums of money were spent backing both models as they fought for control of the market.

Getting related

Meanwhile, in 1963 Borje Langefors published a paper called *Some Approaches to the Theory of Information Systems* in the Nordisk Behandlings Informations Tidskrift. Suddenly there was a serious movement developing around data storage and retrieval and in 1967 Levien and Maron's paper *A Computer System for Inference Execution and Data Retrieval* was published in the Communications of the ACM. Yet another paper appeared, this one in 1968 by D L Childs's entitled *Feasability of a Set Theoretical Data Structure - A General Structure Based on a Reconstructed Definition of Relation in the Information Processing-68*, Proc - IFIP 1968.

All of these papers tried to apply set theory, not just to file theory but more importantly to database structure. Having created an environment in which database design could be discussed by academics as a science rather than in the abstract, momentum was 'growing faster than anyone could have thought. In 1970 Dr Edgar Codd published a paper entitled *A Relational Model of Data for Large Shared Data Banks* in the Communications of the ACM. This was to prove the start of a long process of formalising the set structure and algebra necessary to create a relational database model.

The more detailed history of the relational database can be explored in many works but more importantly it was hailed as providing the first, mathematically proven approach to data storage and retrieval and therefore as sounding the death knell for both the hierarchical and Network models.

Birth of the PC database

Those who can remember the late 70s through to the mid 80s will recall the debate, lively at times, vicious at others, when the future of databases was discussed at length. It was into this arena that the PC database was born.

Almost from the start PC databases were split into two camps. The flat file or hierarchical and the relational. The first PC database was dBASE which had been renamed after it was purchased by George Tate in 1979. By 1980 dBase II+ was launched which boasted of being the first Relational database on PCs. Certainly for anyone pur-

Object Studio

Speak the database language

Now we have a problem with the means of storage. If we are going to have a powerful database which requires careful planning, design and knowledge of performance tuning, then it should be in a structured environment where money and resources can be allocated when required. This satisfied the client part of the solution as design and performance were under control.

Accessability was achievable by the use of a standard enquiry language. Eventually it was agreed that by using SQL, it would be possible to enquire on most relational databases to a reasonable degree.

Providing the database conformed to the requirements laid out in v1 of the Relational model then the fact that the data was distributed over several machines was irrelevant: the system catalog would deal with that problem.

Glueing back to the front

Attention then shifted towards the transportation layer and here every vendor had his own idea. All were prepared to embrace ideas from others, if only to ensure compatability and prevent themselves being shut out. But I-NET, ODBC, IDAPI and many others appeared and were all being promoted as viable transportation layers which would take enquiries from the PC and apply these to the databases wherever they were located.

This transport layer had some problems, however. It concerned the way that queries were framed and the resultant data set. An enquiry on a corporate database could return a dataset greater than the capacity of the local machine to use it. What if the user wanted to move forward or backward through the data? If the data was to be held on the server until needed by the PC then there were some serious drawbacks over capacity. These were compounded by the potential problems of distribution of the data, and concurrency and validity.

At this point, most database vendors started to realise that instead of persuading people to get rid of their PC databases, all they had done was to create a greater need for them in order that the results of their enquiries on the corporate data could be stored locally.

Many people felt that the enquiries could be fed into spreadsheets or word processors. But if the data was to be utilised then it had to be pulled down into a DBMS on the local PC. Let us consider an example.

If I were to enquire on a large corporate database for information about suppliers in a particular area and wanted to look at all the goods I had purchased from them, the data obtained is likely to be spread over a range of different tables and, possibly, over a couple of machines depending on the size of the organisation.

Client/Servers quibbles

This information would have to be pulled into pseudo tables and a pseudo database by the corporate system as it assembled the result of my enquiry. If I wanted to then refine my enquiry, or carry out a range of operations on the data locally, then I would have the problem of being able to rephrase my request and select the subsets. First, I would have to wait whilst all the data was deselected, and then reselected using the new, tighter, criteria. I would then have to wait whilst processing time was found at the server end.

> For anyone purchasing in the early days of desktop computing you only had dBASE, Wordstar and either Supercalc or Lotus 123

One major problem with client/server applications is that they stress the server database more and increase the throughput of data. Instead of being able to limit the number of concurrent users by limiting the logons, with client/server, requests are fired at the server from a range of places. These are queued to provide a steady high level of usage. Peaks and troughs often disappear to be replaced by the system working close to capacity for longer.

Should I then decide that there was still too much data to work with or that I had left out a field, then I would have to go through this process all over again. This could only lead to a decrease in performance both for the user and for the system overall. By using a PC database I could make my initial request and then either work from the server or transfer the data locally allowing me to work on the data making subsets and producing reports or manipulating the data without impact on other users or systems. This can only be achieved with a PC database because of the need for a DBMS to handle the volume of data.

All told this makes a very convincing case for the retension of the PC database. Let us now consider another.

A case for the PC

One of the reasons that the PC database gained rapid acceptance was the time and costs of storing data centrally. Centrally stored data reduces access and flexibility and introduces a costing factor to any single piece of data. To allocate storage on a mini or mainframe requires that the data is important to the company in the wider scheme of things. As a user, my view of data is often limited to my particular job function and the my needs in ensuring that I can meet my goals and targets.

Much of the data stored on PCs is not required by other users or other departments so it would be difficult to make a case for central storage where the costs of that storage are much higher than the PC. This would lead to a reduction in data storage with the potential for a reduction in efficiency in working practices because access to these seemingly insignificant pieces of data can mean the difference between success or failure.

A name and address database for one person might, in a sales department, be needed by other people but if it is only there as a list of possible clients can you justify it being held in a central database?

If it is a list of possible clients then it must be possible for any person to delete records from it when they prove to be of no use. How do you deal with the disgruntled employee deleting records to satisfy some imagined slight?

Finally

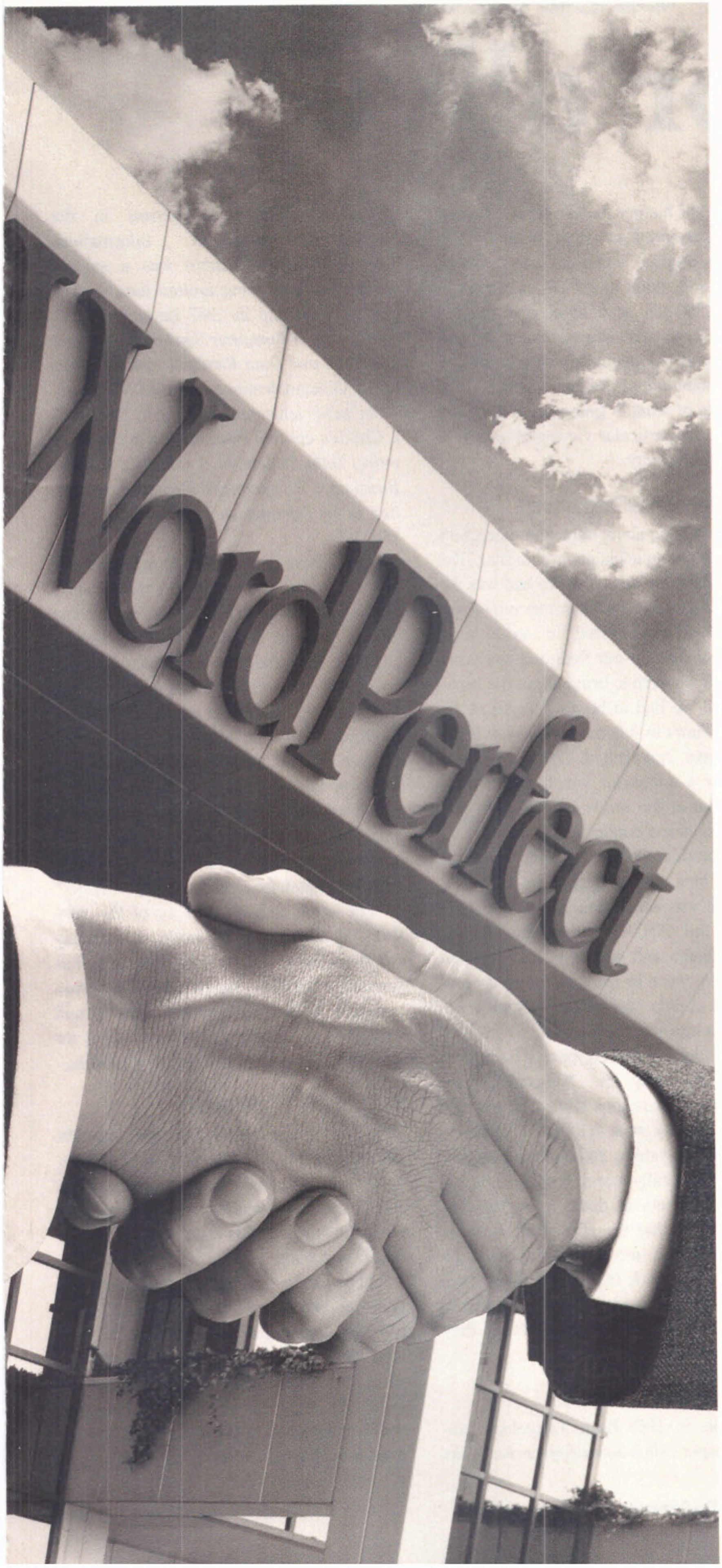
If I want to work through a set of records from a range of suppliers prior to evaluating who we will do business with over the next few months I need to be able to represent that data in several ways. Once I have the initial data set then I require no further access to the corporate data storage so immediately I have the freedom to change, manipulate, delete and play what-ifs with my data set locally.

I am not constrained by the need for links to corporate machines, I am not affected should a computer fail. No one needs to allocate space on the client computer or create a structure to store my data in.

The requirements for PC databases for these reasons alone will transcend the current move to client/server. All the prophets currently proclaiming the end of the PC database have little experience in the history of computing. They need to take time out to step back and look at the problems of only have a mid-range solution with links to spreadsheets and word-processors. ■

Ian Murphy has been in IT since 1983. He can be contacted by email as `imurphy@cix.compulink.co.uk`*.*

IF YOU'RE LOOKING TO DEVELOP YOUR PROFIT POTENTIAL



WE'RE THE PERFECT TEAM

If you're in business to produce and market products and solutions that are perfectly in line with market demands, you need to be in business with WordPerfect® Corporation. We've designed a programme just for you – the Working with WordPerfect Programme.

The Working with WordPerfect Programme gives you the technical and marketing tools you need to create products that integrate with WordPerfect Corporation applications. It's a perfect partnership that offers you complete solutions and allows you to enhance your appeal in the market place.

WORKING WITH
WordPerfect
CORPORATION

Sign with the Programme and you'll enjoy a number of benefits through three developer categories including opportunities for early integration, **OpenDoc developer toolkits**, software developer kits to help you develop "bolt-on" applications and customised solutions: **Developers newsletters**, previews and discounts on listings in WordPerfect Corporation publications and on **WordPerfect conferences and seminars**.

For an application form or for more information on the Working with WordPerfect Programme, please call WordPerfect on

0932 850505

Yes, I would like more information and an application form for the Working With WordPerfect Programme.

PLEASE COMPLETE IN BLOCK CAPITALS

Name____________________

Company____________________

Address____________________

____

__________ Postcode__________

Telephone No.____________________

WWPEX

WordPerfect®

WordPerfect is a registered trademark of WordPerfect Corporation in the USA and other countries.

➤ CIRCLE NO. 562

Object-Oriented down to the bone

Robert Trangmar takes a look at current and future trends in object-oriented operating systems with particular reference to the Mach microkernel and NeXTstep.

I must admit to having had a few reservations when I was asked to write this article. Mostly because despite the large steps forward made using the object-oriented approach to application development, there is still a conspicuous shortage of mature operating system technology based on the same concepts. However, times are changing and several companies including the traditional 'big-guns' such as IBM and Apple (with their Pink/Taligent project) have embraced the object model for a new generation of operating systems with commercial products nearing the end of their development cycle. Even Microsoft, previously not best known for its radical leaps forward in OS technology is having a go, so what sort of things are they actually doing and how will it benefit developers?

Objects, from the kernel up

One of the great things about Unix is, and always has been, its powerful system services (i.e. interactive multi-tasking, transparent networking and device access) all reachable by the developer through a relatively simple, though continually growing, set of system calls. As the years have passed, however, the Unix kernel has grown ever more large and unwieldy, such that the addition of new functionality has had an impact on the size and performance of the system as a whole, making the average monolithic Unix kernel something of a fearsome beast to maintain and debug. Something had to give eventually and along came the concept of microkernels; systems so small that a current aim is to make them fit totally into on-chip cache memory so that even the losses in performance associated with accessing the main address bus are optimised out. However, this reduction in size is not achieved without some casualties in terms of the traditional services a kernel provides. Figure 1 gives a generic view of how microkernel based OS are designed.

As the years have passed the Unix kernel has grown ever more large and unwieldy

The Mach approach

During the early part of the 80s, researchers and students at Carnegie Mellon University (CMU) in Pennsylvania took the then unusual approach of designing an operating environment which dealt only with memory management, process (task) con-

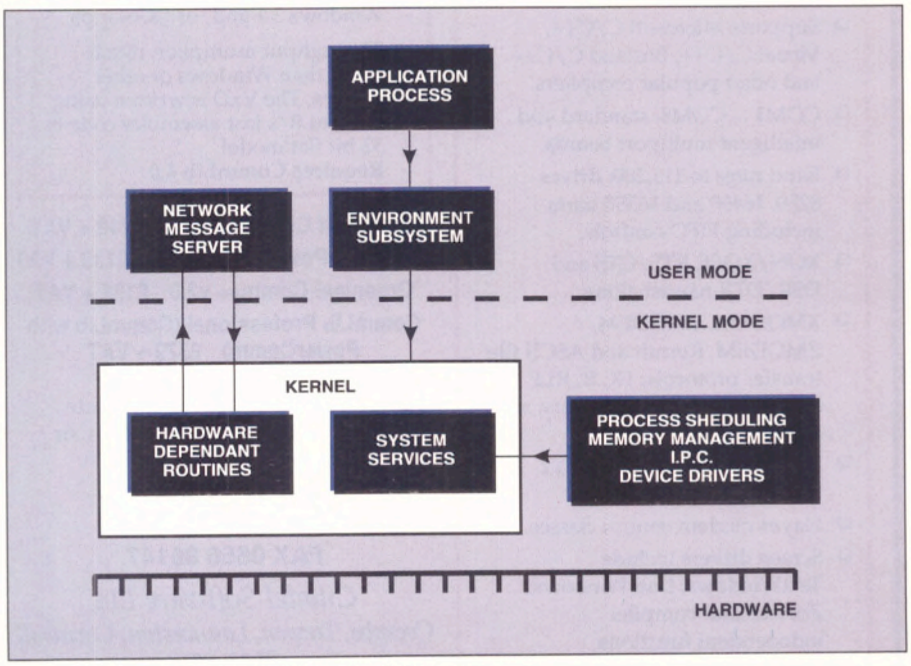


Figure 1 - A microkernel-based architecture

chasing in the early days of desktop computing, you only had dBASE, Wordstar and either SuperCalc or Lotus 123 as your software packages.

The problems with designing relational databases and the conflict as to what should constitute a Relational database was probably responsible for the fact that the flat file databases were able to take control of the desktop database market. Slowly, however, other products came to the desktop as Relational databases and people started to take the PC database seriously.

Size is right

Whilst all this was going on, papers were being published taking about the possibility of actually seeing data as individual objects: the concept of object orientation was being formulated.

During the 1980s the PC database acquired functionality. As hardware got faster and storage capacities improved, much was made of moving systems to PCs from departmental and corporate systems. This paralleled the mainframe/mini computer argument that was currently raging over the ability of computers to cope with the workload required of them. Slowly the minis and super-minis were eating into the traditional mainframe market and taking over the management of corporate database systems. This meant that the data which these systems had previously handled was passed down the line to PCs and onto newly created network servers.

The down side of downsize

Towards the end of the 1980s the cries of downsizing and rightsizing could be heard everywhere. The PC database was becoming the king. It had achieved this remarkable state by default rather than design. Although the systems continued to become more powerful, the traditional database vendors had started to become more and more interested in this market. Informix and Oracle, to name two, both launched PC products and both suffered from performance and storage issues which drastically limited their ability to compete in the PC market.

The growth of the PC database also created serious problems for system analysts who were used to creating a logical model from which the programming teams would derive the working product for the users.

This logical model was created in isolation from the physical product and was often the result of some tortured procedures which required extensive systems design knowledge.

Suddenly there was this group of people who had learnt how to create their own systems, often with little or no formal training and they had very specific ideas on how the system should work and what they wanted from it. When asked what they wanted, they would point to a system on a PC and say, 'Here's our prototype. It does what we want except we need it to run a bit faster and include the following extra functionality.'

This change in circumstances was often followed by demands for better and simpler access to data held within corporate systems and requests for selection of data and downloads to local systems. In addition users wanted the flexibility to move their data around the company, not just their department. Once they had finished with data brought down into their local system they wanted to put the updated data back into the corporate model. Data integrity became a major issue in addition to access and accessibility.

> **Towards the end of the 1980s the cries of downsizing and rightsizing could be heard everwhere**

Client/Server - a necessity

The idea of data stored anywhere within an organisation and accessable from anywhere else was evolving and the terminology of client/server was being bandied about. The question that now remained was 'how could this best be achieved?'

The problems to be overcome were database design, performance, a common transport layer and access rights. The database design was the first sticking point because the user community had become used to building its own databases and applications. When DP departments made it clear, during the 1980s, that they saw PC databases as toys rather than business machines, users developed their own skills, often as a result of many hours playing with the software both at work and at home. This contrasts sharply with the degree courses taken by many DP staff and the structured analyst - analyst/programmer - programmer environment which persists in corporate computer departments.

As databases grew in size, the need for greater access increased with it. The result was that databases were moved off PCs onto network servers where many users could gain that access, but this often resulted in poor performance and even data corruption. The solution was to marry the skills of the PC users and the DP department to evaluate the product and to correct the design of the files. This in turn corrected the second problem, that of performance. Even so, there was often a need to start throwing resources at a database in a way that previously hadn't been possible when it had resided on a PC. In addition there was a need for products which could span the different operating systems and which would allow data to be interchanged easily.

Buying in databases

The solution to this came from two different camps. The first was the PC companies who saw a chance to move into the Unix marketplace with their databases using their existing customers as pressure on DP departments to treat them seriously and picking up vast sums of business along the way. The second group was the more traditional vendors who were convinced that there had to be a way of moving their products down onto the PC platform or, if it came to it, buying an existing product and merging it with their other products.

Amongst the first group were Fox Software, Ashton-Tate and Innovative Software, all of whom had considerable reputations in the PC market and all of whom had much to gain by looking at other markets. Ironically they all looked at the Unix/Xenix market and they all ended up being acquired by other players. Fox Software was acquired by Microsoft who needed a company that was serious about databases to plug a deficit in its own product line; Ashton-Tate was acquired by Borland who saw a chance to enhance its market share by buying the long time market leader and Innovate found itself acquired by Informix Software who had some grand plans for its Smart product.

For Microsoft and Borland, the initial requirement was to have a heavyweight database which could provide the power at the PC and network end whilst being able to compete at the bottom end of the Unix marketplace. Both were working on lightweight database products, Access and Paradox respectively but these lacked some of the sophisticated features and power to be found in their new acquisitions.

Informix saw it differently. It promised the power of its Unix products mixed with the simplicity of a DOS product where power and ease of use would go hand in hand. Unfortunately it misjudged the market and has now leased the rights to Smartware to a Canadian company called Angoss which is showing that the original goal is achievable but through a slightly different route.

themselves on a single machine, it is possible for a task on one machine to send and receive from a task on another. This is a lower level form of the Distributed Objects concept which can allow for, among other things, remote processing, although because of the relative slowness of 10Mibs Ethernet is most often used for network information systems.

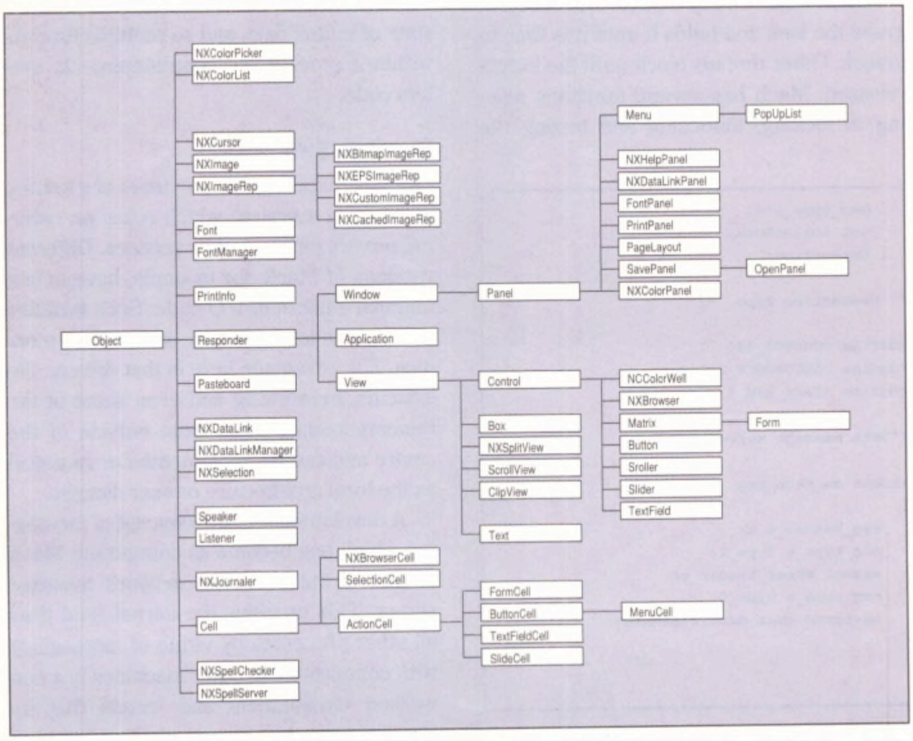
A recent development in the field of lower level servers is that of object oriented process schedulers with differing priority schemes allowing system administrators to load differing modules to suit individual circumstances (see the CHOICES O/S from the University of Illinois at Chicago). For example, if a machine is a dedicated NFS server it may be better to run with a scheme that only has a small amount of interactive capability, spending much of its time servicing file system requests on a high but fixed priority, whereas a user machine would probably have the opposite. Likewise with file system support.

With the legacy of many differing file systems behind us over the years, designers are now looking at ways of implementing many such schemes within new operating systems. OS/2, for example, wishes to have access to HPFS and DOS FAT style systems, yet designers and developers need a consistent interface to them both. By implementing file system I/O drivers as objects, the interface to each can remain the same and relatively clean, being loaded at boot time as the user requires. Developers then ignore the differences (except in the case of long or short filenames) and access the file system(s) using pointers to the system interfaces. Therein lies the polymorphism...

In the past with Unix kernels, device drivers were written to a fairly clean interface and then compiled directly into the kernel which was then rebooted. A feature of the NeXT Mach kernel (and now others as well) has been the facility for loadable kernel servers. These are code modules which provide low level services but which can be loaded and unloaded while the system remains running. Such a module will initialise itself and then register its main communication port in a kernel table.

Mach and other microkernels don't provide much in the way of a user of programmer friendly environment

External programs which wish to send and receive data simply ask the kernel for a handle to the servers port and then send it a simple message which includes a return port for the server to use when supplying the data. The advantages of such modules are in their proximity to the kernel itself. They generally buffer very little data (address space for kernel servers is limited in Mach to 4096 bytes, anyway) but have serious performance gains over higher level system servers and can be unloaded to free memory when not in use. Like the kernel, itself, these modules run in kernel mode and so care should generally be taken with exception handling since they can (and will often during the development stage!) cause system crashes.



Figures 2, 3 and 4 list some basic code for a loadable kernel server which acts as a packet sniffer on computers running NeXTstep's version of Mach. The first two are headers. The MESSAGE.H file includes the service name and Mach message format for communication with a client process (code for this is not included... I'll leave that as a project for the more keen among you!) and so should be included in that source. The SNIFKERN.H and SNIFKERN.C files contain the bare minimum routines needed to intercept a packet as it leaves the driver and make a copy in the local (kernel) address space before wrapping it in a Mach message and broadcasting it to any (single) client that attaches. Note that I have created the Mach message format by hand instead of using the Mach Interface Generator, MIG. This is purely because it is so simple.

A Client running in user space can deal with the packet as it see's fit: probably by putting it in a queue with one thread that listens to its receive port. This thread removes the packet from the queue and sorts it into the different packet types i.e. IP, IPX, ARP, etc using the frame type. The code is compiled with `KERNEL`, `KERNEL_FEATURES` and `Mach` flags defined, then the `SNIFKERN_RELOC` module that results is loaded and unloaded using the NeXT utility, `kl_util`, the kernel server loader.

Applications as clients

That said and done, it's a fair leap to the arena within which most developers do their work. Most programmers work in an environment such as Windows or DOS where a much higher level of abstraction is needed in system calls. Mach and other microkernels don't provide much in the way of a user or programmer friendly environment and another level entirely is needed for developing within a GUI or even for opening files. This is catered for, as previously mentioned, through the use of servers; processes run at system startup which have little or no hardware specific code but answer all requests through the microkernel and provide the programmer with usable tools to do his or her work. An example of this on NeXT computers is the BSD 4.3 server which presents the programmer or user with a Unix compatible front end. Essentially this means that the services of the microkernel can be ignored and a programmer can develop applications which just use Unix system calls. These get translated by the server if it cannot

Rev Up Database Programming

with the Greenleaf Database Library

NEW!

Powerful Functions

- The *Soft*C Database Library is now the new Greenleaf Database Library.
- Unsurpassed speed and flexibility for access to industry-standard database data, index and memo files at an affordable price.
- Databased Advisor says, *"SoftC Database Library has top ratings!"*
- Don't be fooled by pretty ads: the Database Library is all you need to interface with dBASE III, dBASE IV, FoxPro, FoxBASE, Clipper, dBXL, and other xBASE files including new FoxPro CDX index files.
- Supports MS-DOS and Windows, and is portable to OS/2 and UNIX.
- Includes Windows 3.1 DLL, supports Microsoft, Borland and Zortech C/C++.
- Single and multiple user; network access fully supported.
- Database package not required: this product is a complete ISAM library.
- Windows DLL linkable with most DLL-capable compilers (including Visual Basic).

All Greenleaf libraries feature:

- No royalties
- 90-day money-back guarantee
- FREE unshrouded source (ANSI & K&R)
- Top rated documentation AND online documentation with FREE help engine!
- FREE BBS access
- FREE unlimited UK technical support

Database Library version 3.22 £249 + VAT

☎ Telephone today for complete information, demonstration, or details of how to order.

0566 86037

FAX 0566 86147

Citadel Software Ltd
Coombe, Trewen, Launceston, Cornwall PL15 8QF

GREENLEAF
Software©

The Ultimate in Communications Programming

NEW: Comm++ 2.0

- C++ class library for MS-DOS, Windows 3.1 and OS/2.
- Device independence through inheritance.
- Supports Microsoft C/C++, Visual C/C++, Borland C/C++ and other popular compilers.
- COM1 ... COM8, standard and intelligent multiport boards.
- Baud rates to 115,200; drives 8250, 16450 and 16550 uarts including FIFO controls.
- XON/XOFF, RTS/CTS and DSR/DTR handshaking.
- XMODEM, YMODEM, ZMODEM, Kermit and ASCII file transfer protocols; 1K, R, RLE compression, crash recovery, and more.
- VT52, VT100, ANSI and TTY terminal emulation.
- Hayes modem control classes.
- Screen drivers include TextWindows, DataWindows, Zortech and compiler independent functions.

PowerComm

- PowerComm Toolkit for Windows v1.0: CommLib 4.0 DLL and VxD.
- Drives most standard multiport boards; share board among Windows 3.1 and/or DOS apps.
- Throughput multiplier: much faster than Windows or other drivers. The VxD is written using CommLib's fast assembler code in 32-bit flat model. **Requires CommLib 4.0**

Greenleaf CommLib v4.0 £249 + VAT
Greenleaf PowerComm v1.0 £139 + VAT
Greenleaf Comm++ v2.0 £199 + VAT
CommLib Professional (CommLib with PowerComm) £379 + VAT

☎ Telephone today for complete information, demonstration, or details of how to order.

0566 86037

FAX 0566 86147

Citadel Software Ltd
Coombe, Trewen, Launceston, Cornwall PL15 8QF

CommLib 4.0

- Level 2 device independent C functions for these drivers: MS Windows, Greenleaf Standard, Fast, Smart, DigiBoard, Smart Arnet, Smart Star Gate, Sparkle, BIOS, Extended BIOS, Polled, MODEM Assist, FOSSIL.
- Windows 3.1 support includes TextWindows: easy programming.
- 286 Protected Mode for Phar Lap and Rational Systems DOS extenders.
- XMODEM, YMODEM, ZMODEM, Kermit, ASCII protocols: many options.
- Unlimited number of ports for ISA, EISA and MCA.
- Modem controls, keyboard, screen, handshaking and much more.

GREENLEAF
Software©

➤ CIRCLE NO. 563

Nu-Mega™
TECHNOLOGIES INC

AVOID EMBARRASSMENT!

NEW! VERSION 2.0

"A customer found a bug in our software with a tool called BOUNDS-CHECKER . . . Why aren't *we* using BOUNDS-CHECKER?"

Use BOUNDS-CHECKER™ V2.0 For Windows, The Automatic Bug Finder!

Whether you're the boss, the QA Manager or the programmer, it's your responsibility to ensure that your company's Windows programs are "bug-free" before they get into the hands of customers. If you're developing under C or C++, producing a "bug-free" Windows product is no longer a long and stressful experience.

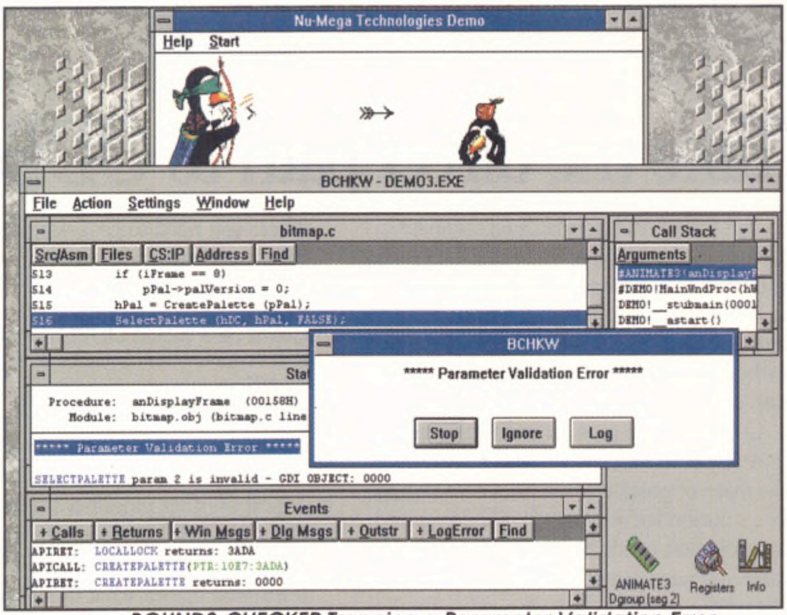
Announcing **BOUNDS-CHECKER V2.0 For Windows**, the software developer's "safety-net". Quickly and easily eliminate the hardest-to-find Windows errors that can take days - even weeks to find like:

- **API Parameter Errors**
- **API Return Value Errors**
- **Data and Heap Corruption**
- **Resource Leakage Problems**
- **Memory Leakage Problems**
- **Processor Faults**

BOUNDS-CHECKER works by transparently setting hundreds of breakpoints within your program to monitor its behavior. When a bug is detected, BOUNDS-CHECKER immediately stops your program and pops up showing the problem. You can then inspect your program's source, variables, stack and heap ... with BOUNDS-CHECKER's powerful display windows.

For those particularly nasty problems, we've introduced new event logging. This lets you look back in time to see what led up to the problem. The events which include messages and API calls can be easily filtered with the click of a button so you can view only the events of interest.

Unlike other debugging tools, there's no learning curve with BOUNDS-CHECKER. Simply select your program's name from BOUNDS-CHECKER's menu; all the rest is automatic. There is nothing to link-in and no macros to compile into your program. All this plus the functionality of a heap checker, debug kernel, API debugger, and a post mortem tool into a single comprehensive automatic bug finder.



```
        if (iFrame == 0)
            pPal->palVersion = 0;
        hPal = CreatePalette (pPal);
        SelectPalette (hDC, hPal, FALSE);
```

```
#ANIMATE3!anDisplayF
#DEMO!MainWndProc(hW
DEMO!__stubmain(0001
DEMO!__astart()
```

***** Parameter Validation Error *****

```
  Procedure:  anDisplayFrame  (00158H)
     Module:  bitmap.obj (bitmap.c line

***** Parameter Validation Error *****

SELECTPALETTE param 2 is invalid - GDI OBJECT: 0000
```

```
APIRET:  LOCALLOCK returns: 3ADA
APICALL: CREATEPALETTE(PTR:10E7:3ADA)
APIRET:  CREATEPALETTE returns: 0000
```

BOUNDS-CHECKER Trapping a Parameter Validation Error

New NT Product!

Don't be Left Behind...Develop in Windows NT With Confidence! Use BOUNDS-CHECKER32/NT, the powerful new debugging tool specially designed to meet the demands of developers using the WIN32 environment. Catch WIN32 bugs at their source and ship your products faster! *Exclusive offer from System Science,*

only £195.00 (plus VAT)!

Call 071 833-1022 to order NOW!

Call now for overnight delivery.

BOUNDS-CHECKER V2.0 For Windows. **Only £195** +VAT

BOUNDS-CHECKER For MS-DOS....**£159** +VAT

Order both versions & SAVE £105... **Only £249** +VAT

BOUNDS-CHECKER, SOFT-ICE, AND NU-MEGA TECHNOLOGIES are trademarks owned by Nu-Mega Technologies, Inc. All other trademarks are owned by their respective owners.

Call: (071) 833 1022
Fax: (071) 837 6411

3-5 Cynthia Street
London N1 9JF

System Science

➤ CIRCLE NO. 565

RISK=NULL
30 DAY
MONEY-BACK GUARANTEE

24 HOUR BBS
603-595-0386

trol and IPC (inter-process communication). All other systems services were left for other layers (in user space) to deal with. In fact, this modular approach was extended further in that most routines within the kernel (known as Mach) were divided into hardware dependant and hardware independent groups so that a high degree of portability could be achieved. Like most OS work it was written in C, with assembler being used only in the most speed critical/hardware specific sections.

Mach has provided a framework for the future. Much of the current and next generation systems have been built by developers drawing on the work at CMU. This is notable from the choice of Mach 3.0 as the base for IBM Workplace OS and Taligent. NeXT Computer Inc started way back in 1986 and modified version 2.0 for the bottom layer of its NeXTstep operating system; the major differences being the kernel size (Mach was not then truly a microkernel) and the addition of 'home-grown' loadable kernel servers.

Sewing A Thread

First off, it's probably stretching things a bit to call Mach an Object-Oriented system (Chorus, a French microkernel has a better claim, having been mostly written in C++, using the concepts much more fully) though the central tenets of OOP systems with their message passing have been adhered to. Mach, itself, relies greatly on external 'servers' to provide the user/programmer with functionality such as network support, and user services such as shells. Communication between these programs are in the form of Mach messages (see the *Fly with MIG* box for an example). Mach splits the traditional notion of a process into two abstractions, the task and the thread; a task being the environment in which program execution occurs (ie the address space) and a thread being the basic unit or point of control of execution.

Each task that the system creates has its own communication 'ports', at least one thread and a set of rights which allow or deny communication to other parts of the system including memory management. Within each task, all threads inherit and share these services which cover exception handling and standard IPC. A thread contains very little state information. Thus on multi-processor systems (one of the design goals of the CMU researchers) the kernel can optimise its usage of the hardware environment. A key concept with threads is the fact that all threads within the same task share address space and thus have global access to the same data. (In general, Forking tasks also share address space until one task wishes to write to a variable, then an exception occurs and the data is copied to the new task's private address space). This scheme is called 'copy-on-write' and is used to speed transitions by delaying and staggering the expensive job of creating new processes). Variables shared between threads should generally be marked volatile or access to them should be protected in code by the use of mutex locking. Basically a mutex is a mutual exclusion flag around the data which suspends threads until another thread has finished updating it. For example:

```
mutex_lock(myLock);
...
/* modify shared data */
...
mutex_unlock(myLock);
```

When one thread enters this code it grabs the lock and holds it until it is time to unlock. Other threads block until the lock is released. Mach has several functions relating to locking, unlocking and testing the state of mutex flags and so multiple threads within a process are commonplace in system code.

```
#include <stdio.h>
#include <kernserv/prototypes.h>
#include <kernserv/kern_server_types.h>
#include <net/netif.h>
#include <netinet/in.h>
#include <sys/errno.h>
#include <sys/types.h>
#include <strings.h>
#include <net/etherdefs.h>

/* The advertised kernel service port  */

#define SERVICE_NAME "SNIFFIT"
#define MSG_HAVE_PACKET  -1000

/*Connection message structure*/

  typedef struct
  {
    msg_header_t h;
    msg_type_t t;
    int inlineData;
  } Connection;

/* Connection type  */

#define CONNECT_REQ  1
#define DISCONNECT_REQ 2
#define STATS_REQ 3

/*Data message structure*/

struct en_Mach_msg
{
   msg_header_t h;
   msg_type_t type_1;
   struct ether_header eh;
   msg_type_t type_2;
   unsigned char data[ETHERMTU];
};
```

Figure 2 - Mach messaging definitions for Packet sniffer and client

```
#include "message.h"


#define SNIF_KERN_NAME "snifit"
#define SNIF_UNIT 0
#define SNIF_TYPE_NAME "Packet Snif-
fer"


#define TYPE_SIZE 2
#define SNIF_MTU (ETHERMTU -
TYPE_SIZE)
#define DADDR_OFFSET 0
#define SADDR_OFFSET (DADDR_OFFSET + \

NUM_EN_ADDR_BYTES)
#define TYPE_OFFSET (SADDR_OFFSET + \

NUM_EN_ADDR_BYTES)
#define DATA_OFFSET (TYPE_OFFSET + \
                     TYPE_SIZE)


// Globals for the SniffIt
// kernel module


static unsigned long in_packets;
// the ethernet device
static netif_t netif_device;
// frame type of ethernet packet
static u_short en_frame_type;
// application receive port
port_t remote_port;
kern_server_t instance;


// message buffer for application


struct pkt_buffer
{
  int in_use;
  struct en_Mach_msg msg_buf;
};
```

Figure 3 - Definitions for SNIFKERN.H header

Services thru objects

Message handling is a key tenet of a kernel, more so in a system which relies on external servers for most of its services. Different versions of Mach, for example, have only a minimal amount of I/O code. Such facilities are mostly implemented in external modules. The advantage here is that drivers, file systems, networking and even some of the memory management exist outside of the centre and can be implemented or removed as the local architecture or user dictates.

A development of the concept of messaging which has become as common in Mach is that of the user-level network message server. This provides the kernel (and thus all other processes by virtue of messaging) with connectivity to other machines in a networked environment and means that instead of tasks just communicating between

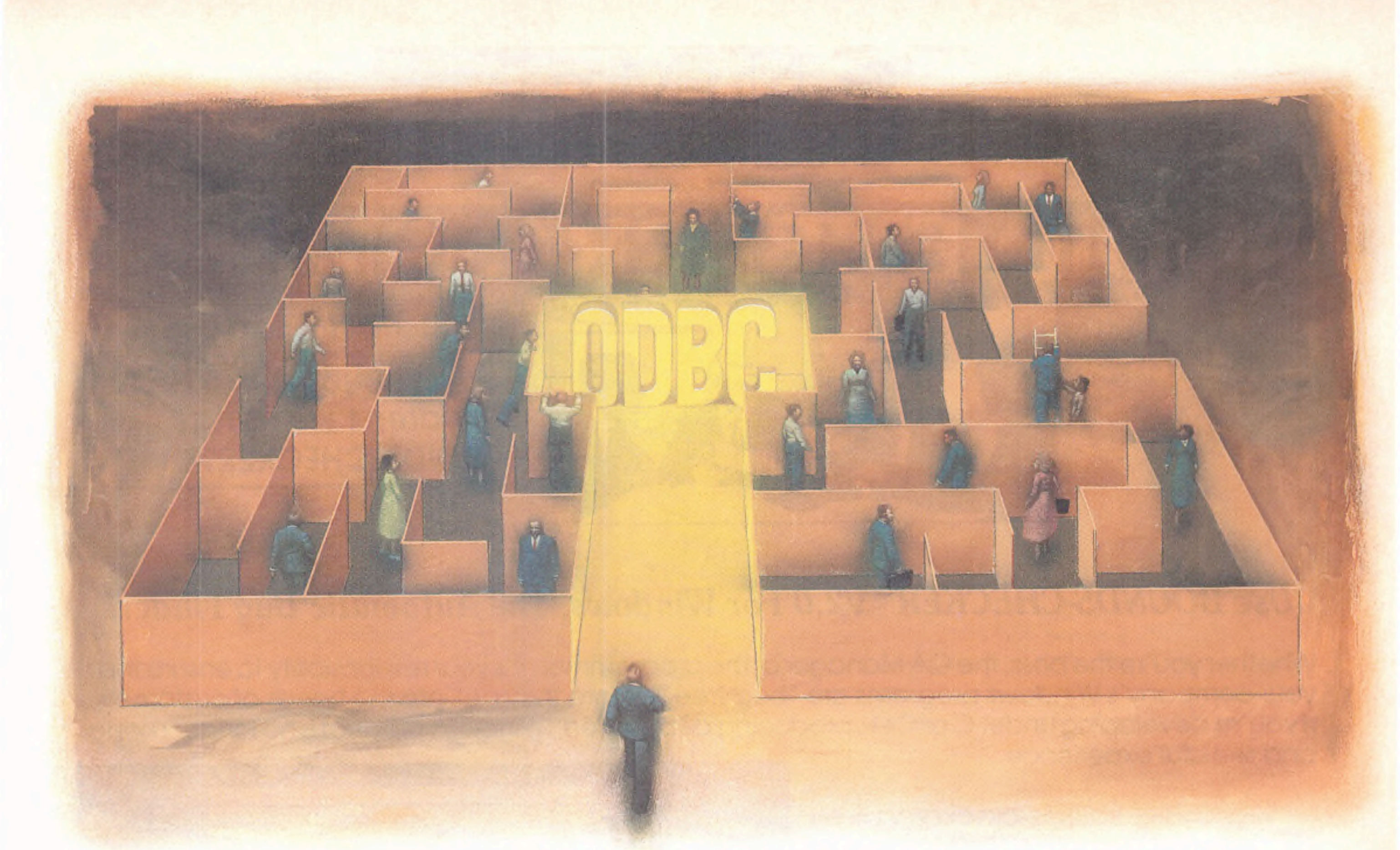
```
/*  Basic initialisation and providing in-
tercept routines for
 *   a packet travelling from the driver.
If successfully read
 *   (using nb_read(...)) then copied into
a Mach message
 *  and sent to any clients that have reg-
istered.
 *
 *  Note: This only handles a single cli-
ent (though multiples
 *   would be only require a table...).
Clients can be
 *  identified by their unique Mach port.
 */

#include "snifkern.h"

static int debug;
struct pkt_buffer *snifPkt;

/*
 *   Simply send the message out to the
client...if queue
 *   is full then don't bother. This is
called by the
 *   kern_serv callout routine and can
block. Since we
 *  have very little space for storage and
packets
 *  can arrive thick and fast i have cho-
sen not to
 *  try to deal with them. Remember a sin-
gle packet will
 *  be ETHER_MTU + 18 bytes big.
 *
 */

void send_msg(void *ignore)
{
  msg_return_t ret;

  if(!snifPkt->in_use)
    return;

  ret = msg_send((msg_header_t *)&snifPkt-
>msg_buf,
        MSG_OPTION_NONE, 0);
  snifPkt->in_use = 0;
}

//Get a message off the driver, format and
send a.s.a.p
void queue_msg(netbuf_t thisBuf)
{
  int psize;
  kern_return_t ret;

/* we want to move quickly but no point
   if no application needs data */
  if(remote_port == PORT_NULL)
    return;

  if(snifPkt->in_use)
    return;

  snifPkt->in_use = 1;
  psize = nb_size(thisBuf);

      nb_read(thisBuf,   0,   ETHERHDRSIZE,
&snifPkt->msg_buf.eh);
   nb_read(thisBuf, ETHERHDRSIZE, (psize -
ETHERHDRSIZE),
         &snifPkt->msg_buf.data);

// now we pack the Mach message types
  // do the header
  snifPkt->msg_buf.h.msg_simple = TRUE;
          snifPkt->msg_buf.h.msg_size     =
sizeof(struct en_Mach_msg);
          snifPkt->msg_buf.h.msg_type     =
MSG_TYPE_NORMAL;
      snifPkt->msg_buf.h.msg_local_port   =
PORT_NULL;
   snifPkt->msg_buf.h.msg_remote_port = re-
mote_port;
           snifPkt->msg_buf.h.msg_id      =
MSG_HAVE_PACKET;

   // do the ethernet header
    snifPkt->msg_buf.type_1.msg_type_name =
MSG_TYPE_BYTE;
   snifPkt->msg_buf.type_1.msg_type_size =
           sizeof(snifPkt->msg_buf.eh);
    snifPkt->msg_buf.type_1.msg_type_number
= 1;
    snifPkt->msg_buf.type_1.msg_type_inline
= TRUE;
     snifPkt->msg_buf.type_1.msg_type_long-
form = FALSE;
   snifPkt->msg_buf.type_1.msg_type_deallo-
cate = FALSE;

   // do the data packet
    snifPkt->msg_buf.type_2.msg_type_name =
MSG_TYPE_BYTE;
   snifPkt->msg_buf.type_2.msg_type_size =
           sizeof(psize - ETHERHDRSIZE);
    snifPkt->msg_buf.type_2.msg_type_number
= 8;
    snifPkt->msg_buf.type_2.msg_type_inline
= TRUE;
     snifPkt->msg_buf.type_2.msg_type_long-
form = FALSE;
   snifPkt->msg_buf.type_2.msg_type_deallo-
cate = FALSE;

    if((ret  =  kern_serv_callout(&instance,
send_msg, NULL)) !=
     KERN_SUCCESS)
   {
     printf("[%s]: Queue packet: kern_serv-
er_callout() failed with [%d].\n",
        SNIF_TYPE_NAME, (int)ret);
   }
}

//Inform device that we want it's packets
static void snifkern_attach(void *private,
netif_t realifp)
{
  netif_t mnetifp;

  debug = 0;
  in_packets = 0;
  remote_port = PORT_NULL;

  snifPkt =
             (struct   pkt_buffer   *)kal-
loc(sizeof(struct pkt_buffer));
  if(!snifPkt)
   {
     printf("[%s]: Error allocating memory
in snifkern_attach()\n",
        SNIF_TYPE_NAME);
     return;
   }
  snifPkt->in_use = 0;

   if(strcmp(if_type(realifp),  IFTYPE_ETH-
ERNET) != 0)
       return;
   mnetifp  =  if_attach(0,  snifkern_input,
0, 0, 0,
                         "snifit", SNIF_UNIT,
                         IFTYPE_ETHERNET,
                          SNIF_MTU, IFF_PROMIS
C,
                         NETIFCLASS_SNIFFER,
                         NULL);
  if(!mnetifp)
   {
     printf("%s: attach: error from if_at-
tach()\n",       SNIF_TYPE_NAME);
     return;
   }
  netif_device = realifp;
}

// initialisation function called by kern-
loader
void snifkern_config(int n)
{

  debug = 0.0;  // Unused as yet..

  printf("SnifIt: Kernel Config...\n");
        if_registervirtual(snifkern_attach,
NULL);
}

// handle and queue ethernet packets
static   int   snifkern_input(ifp,realifp,
buf,extra)
netif_t ifp;
netif_t realifp;
netbuf_t buf;
void *extra;
{
  int retcode;
    u_short  etype;    //  ethernet  frame
type...we could sort on this

  // do we want packets from this device?
  if(realifp != netif_device)
    return EAFNOSUPPORT;

  // read from the network device
        retcode    =    nb_read(buf,     (un-
signed)TYPE_OFFSET,
         (unsigned)TYPE_SIZE,
          (void *)&etype);
  if(retcode)
    return EAFNOSUPPORT; // reject!

// We want all valid packets..sort the out
in the client
   in_packets++;  // will generate a stats
type later
  queue_msg(buf);
  return EAFNOSUPPORT;
}

//  Mach  Message  handler.  Processes  re-
quests
boolean_tsnif_handle_msg(msg_header_t
*msg, void *extra)
{
  register Connection *In0p;

  In0p = (Connection *) msg;

  switch(In0p->inlineData) {
    case CONNECT_REQ:
            remote_port   =   In0p->h.msg_re-
mote_port;
       break;
    case DISCONNECT_REQ:
       remote_port = PORT_NULL;
       break;
    case STATS_REQ:
        printf("Inbound packets = [%u]\n",
in_packets);
       break;
    default:
        printf("[%s]: Unknown request from
application!\n", SNIF_TYPE_NAME);
  }
  return TRUE;
}

// Free data
void snifit_signoff(void)
{
   kfree(snifPkt,  sizeof(struct pkt_buff-
er));
  printf("Unloading Packet Sniffer\n");
}
```

Figure 4 - A simple loadable kernel packet sniffer using the ethernet interface

Conclusion

The main gains from object oriented operating systems will be in their portability between hardware architectures. This, to the developer, has major gains if it also means multi architecture binaries where an application developed and compiled on one system runs straight from the disk on another system or at least can be re-compiled and run. The great tradeoff in this modularity which will certainly lead to more reliability, however, has been speed of execution. Users are demanding ever more performance from their Machines and, if the operating system itself hogs CPU time while messaging down through different layers of itself, they will not be best pleased. That said, of course, new operating systems are accepted on to the market very slowly and hardware just seems to get faster and less expensive, so there is good reason to believe that by the time system software gets into version 1 or 2 (i.e. tried, tested and just a little bit stable), the hardware will be about to make it fly even if the original beta copy ran like a dog with a wooden leg...! ■

Robert Trangmar is an systems integration consultant living in New York.



Take the direct path to ODBC with Q+E Software.

Simplified and standardized database access . . . that's the promise of ODBC. But you have to get there first. It's easy to get lost in the search for reliable ODBC drivers. And trying to write ODBC-enabled applications without the proper tools can lead you straight into a dead end.

Q+E Software puts an end to the frustration and confusion. We'll lead you through the maze of possibilities with a comprehensive solution for all your ODBC driver and development needs. With Q+E you can get what you need out of ODBC without getting lost in the process.

Connect ODBC-compliant applications no matter what you find around the corner.

With Q+E ODBC Pack, you can take advantage of the same database access technology we provide for such leading software publishers as Lotus, Computer Associates, Sybase, INFORMIX, and PowerSoft. Q+E ODBC Pack offers a comprehensive suite of drivers for more than 30 SQL and PC DBMS versions.

And, as new ODBC applications come out, Q+E Software will continue to take you where you need to go. Each of our drivers is fully supported and kept absolutely up-to-date by our technical support staff.

Count on a robust ODBC development tool that removes the obstacles in your way.

When it comes to making sure your applications are ODBC-compliant, you can rely on the same technology that WordPerfect, Delrina Technology, Crystal Services, and dozens of other prominent software companies use. Q+E Database Library gives you an easy way to build ODBC-enabled applications—no matter what development environment you prefer.

You can forget about differences in drivers, because our development tools let you write code once that works with all major databases. And because Q+E Software keeps up with the constantly changing technology, you won't be left on your own when custom applications need modification.

Turn to Q+E Software for your total ODBC solution.

Whether you're looking for ODBC drivers or planning to build ODBC applications, Q+E Software offers the path that's fast, easy, and direct. Our ODBC solution is multi-platform, with versions for Windows, OS/2, Macintosh, Windows NT, and Solaris. And we offer single-source technical support for more databases than anyone in the industry.

When it comes to ODBC, Q+E Software gets you there now. No delays. No confusion. No walls.

Q+E ODBC Pack and Q+E Database Library provide dependable and complete support for the following databases:
ALLBASE, Btrieve, Clipper, DB2, DB2/2, DB2/6000, dBASE, Excel .XLS files, FoxBase, FoxPro, Gupta SQLBase, IMAGE/SQL, INFORMIX, INGRES, Microsoft SQL Server, NetWare SQL, Oracle 6 and 7, Paradox, PROGRESS, SQL/400, SQL/DS, Sybase System 10, Sybase SQL Server, Teradata, Text files, and XDB. Gateways supported include IBM DDCS/2, Micro Decisionware, and Sybase NetGateway.
Not all DBMS are available on all platforms; some may require a gateway. Call for details.

FOR UK SALES
CALL NOW ON
0727 812812

"Q+E: The Key to ODBC"

– Byte Magazine, March 1994

Q+E® Software
Database Access for Client Server Computing™

Q+E Software
Abbey View, Everard Close, St Albans AL1 2PS
Tel: 0727 811177 Fax: 0727 848991

➤ CIRCLE NO. 564

The Windows NT Registry

If you're sick and tired of that plethora of .INI files in your Windows directory, it's nothing that 16 MB of RAM, a new hard disk and an upgrade to NT won't fix, as **Laine Stump** discovers...

Windows 3.1 configuration information is a mess. Some applications store information in Windows' own WIN.INI file, while some create their own INI file in the Windows directory. My `\WINDOWS` directory contains 68 separate INI files, ranging from 48 to 12879 bytes in length. The total size reported by `DIR` is 72506 bytes, but the actual space taken on the disk is somewhere closer to 280 KB due to DOS' minimum block size. Many of these INI files were created by programs I long ago removed from my disk, but I'm not sure which, and I don't want to go about experimenting, either. So I live with it.

The Problem with .INI

Of course, in these days of less than a dollar per MB of disk drive, 200 KB is nothing to quibble about, but there are other problems. For example, if a word processing program stores the default state of overwrite mode in an INI file, and two different users like to have overwrite mode in a different state on startup, those two users must have separate INI files in separate directories. Unfortunately, most applications always look in the Windows home directory for these files, so each user must have (at least a partial) separate Windows installation. Additionally, many applications don't allow for the simultaneous presence of more than one version of the program - the new version writes over the old one's INI information.

And just today I saw another example of problems with INI files. I stood by and watched as someone with minimal Windows knowledge edited an INI file containing hardware setup information by hand (then later did the same myself). While some might consider it an advantage to have all configuration information in a world readable text file for easy perusal and modification, the truth is it makes it easy for an unqualified user to screw things up.

Finally, my pet peeve about INI files: because most are stored in the Windows Home directory, and many applications just plunk their stuff in the middle of WIN.INI, it is impossible to trash and reinstall Windows, or install a different version of Windows while keeping the old version, without losing the configuration of many of your applications.

The NT Registry

The Windows NT designers had heard about many of these problems and decided that solving them was worth adding another 30 or so functions to the already bloated-beyond-belief Windows API. Hence was born the NT Registry, a mutated offspring of the Windows 3.1 Registration Database.

While the Registration Database contained keys and associated values, it was used only for storing file associations (which application or OLE server to use for which type of file). The NT Registry (apparently also in Chicago) is much more com-

Imagine running Windows with no backwards compability - this would certainly deal a stiff blow to RAM manufacturers and silicon wafer plants

Key	
Namey	String used to access the key. Must be unique at this level.
ClassObject	Class name. Not usually used by applications
Security Descriptor	Standard Windows security descriptor. optional
Time Stamp	Time of last modification of this key's values
Subkey(s)	Zero or more Keys
Value(s)	Zero or more pieces of info:
Name	String used to access the value
Type	Type of data stored in value
Data	Arbitrary data up to 1 MB

Figure 1 - Contents of keys and values

service them itself to lower level Mach messages and the results are passed back to the application or task which acts as the client in this transaction.

NT client/server architecture

Perhaps a better example of multiple user level servers is Windows NT which, again, uses a microkernel architecture (though slightly different from Mach). This has several OS servers including those for DOS, 16 Bit Windows, Win32, POSIX and OS/2, and includes a layer above to coordinate messaging the right subsystem. With compatibility in mind this is an obvious choice in the design as most of these sections of the system are usually written in C or C++ which can then be ported from machine to machine easily once the kernel has been completed.

In something of a departure from its forbears, NT follows much of the object model at its lowest levels. Its kernel mode portion is split into two distinct units; Firstly the HAL (Hardware Abstraction Layer), which is a set of routines implemented as a type of DLL which comprises, as the name suggests, all the hardware specific code and secondly, the NT Executive. This layer has much in common with Mach, but manages a few additional tasks. These include: an object manager for creation, deletion and management of system objects which represent system resources; a security reference monitor that effectively traps calls to local services and enforces security policies and an enlarged I/O subsystem with a manager module and network services.

Above the executive, in protected user mode, run the environment subsystem servers which call the native NT services they need to create for the right execution environments of their individual client applications. Unlike the lower levels these generally only pay lip service to the object model at the moment, though this looks likely to change with the advent of Cairo (a ground up replacement for Win32, etc) and further revisions of user mode services such as OLE.

Object layers for Developers

Servers and clients make for a nice modular design, sure, and this is great for the vendors in that it makes for ease of porting across different hardware platforms, but unless the OOP design continues up through the layers to the user environment and developer tools, what use is it to the hard pressed applications programmer? Well, at first glance very little, unless the layers above the microkernel provide a framework with a well designed root/base class from which full blown software projects can be derived. The best example I can give here (mainly because it is the only OS shipping at the moment which claims a full development environment and is object-oriented) is NeXTstep. (Don't pay too much attention to specifics here as most future OOP systems will follow similar design philosophies, as is the nature of the beast).

NeXTstep's application framework is called the AppKit. All user level classes on this system are written using the base class 'Object'. All objects within a typical NeXT program have at the very least, the functionality of this construct. On the face of it this seems to do very little, it actually however, provides each class derived from it with initialisation methods, their own data storage environment, persistence and runtime method selection. A special class (called unsurprisingly the Application class) is derived from this base and each user application has exactly one instance; in the NeXT case a global called NXApp. This is instantiated at runtime when the application is launched.

Much of the Application kit is given over to GUI objects, with a GUI drawing object called a `view` as the main starting point in the hierarchy for on-screen windows, panels, menus and GUI goodies such as buttons and sliders. Other Kit hierarchies for accessing databases and providing distributed object functionality are also included, but generally provide their own path from the Object class.

Taligent

Taligent's system (not available until next year, I'm afraid) will have a simpler design based around the more advanced and smaller Mach 3.0 kernel with a developer framework of classes called, imaginatively enough, Application Frameworks. In fact, Taligent has set itself some fierce design goals in that it hopes to provide a system that dispenses with the non-oop layer of BSD that NeXT provides and will have all system services available through object classes. Being written in C++, Taligent needs to deal with some very complex issues involving that brain of C++ programming - multiple inheritance - to produce a system which hides much of the low level details and yet leaves the programmer with a powerful hierarchy with which to develop applications. Much of this is similar to the MACApp framework, but Taligent hopes to extend the functionality by having API's at two levels.

The first API is the client which is used by other frameworks/class hierarchies or application developers. This is for developers who simply want to be a user of services, e.g. in the case that he or she wishes to write data to a file. A second API is provided with hooks to the developers code, allowing the modification of the framework from the entry level and upward. This allows hardware changes to be reflected in all classes derived in the framework by the addition of routines at a suitable level. The customisation potential looks good in theory.....

Fly with MIG

Since Mach relies so heavily on external servers to provide operating system services, the kind ladies and gentlemen at CMU provide a utility with Mach releases to generate much of the interface code needed in this communication. The utility is known as MIG (The Mach Interface generator) and it lets the server programmer simply define which functions his or her server will respond too in both the main body of the server code and in a definition file. The definition is parsed by MIG which produces two additions source code files, one for the client and one for the server. The server and clients need only include the routines produced to access the remote functions.

An example of a definition file is given here:

```
#include <std_types.defs>
// a user definable type
// is set here...
// a character pointer

type stringPtr =
^(MSG_TYPE_STRING, 8* 80);

routine
makeRequest(server :
port_t;

 recordKey : stringPtr;
 recordName : stringPtr;
 index : int);
```

A client can call a Mach system function to look up the port of the server providing the information it requires (a global name) and simply call the function `makeRequest(....)` to get the info it requires.

A server implements a listening loop on its receive port and vectors calls to specific functions (such as the above) to the appropriate code segments.

prehensive. Structured much like a DOS or Unix file system, the Registry starts with a few 'root' keys which contain other keys. These subkeys in turn may contain other keys and/or values. Each value has a name and possibly some data associated with it. If you replace 'key' with 'directory' and 'value' with 'file' in the previous two sentences, you'll see the resemblance. Figure 1 gives a formal description of the contents of keys and values.

Organisation

Comparing the Registry to INI files, all configuration information in INI files was stored as ASCII text. The registry supports several different data types (including ASCII text and straight binary BLOBs among other things). Also, whereas the location of an application's configuration information in the INI file structure was haphazard at best, the hierarchical structure of the Registry has made it possible to organise info in the Registry in a standardised, easy to comprehend fashion, making it more reasonable for one application to take a peek at the configuration of another application (or the system itself).

To access any key in the Registry you must open the key with **`RegOpenKeyEx()`**. One of the arguments to **`RegOpenKeyEx()`** is a handle to an already open key. Before you start complaining about an obvious Catch 22, you should know that NT keeps several keys perpetually open, with predefined handles for your use. **`HKEY_CLASSES_ROOT`** is a handle to the key below, in which all information related to file type associations is stored (ie all the stuff that is stored in the Windows 3.1 Registration Database). **`HKEY_CURRENT_USER`** leads to user specific configuration information for the currently logged in user, **`HKEY_LOCAL_MACHINE`** leads to all machine specific configuration and **`HKEY_USERS`** is the parent key of all user keys.

Below **`HKEY_CURRENT_USER`** and **`HKEY_LOCAL_MACHINE`** are partly parallel configuration trees, the idea being that the 'standard' configuration for applications, as well as hardware configuration, will be below **`HKEY_LOCAL_MACHINE`**. Any user preferences overriding the standard configuration will be below **`HKEY_CURRENT_USER`**. Figure 2 shows the first levels of these two trees.

> **It is impossible to trash and reinstall Windows without losing the configuration of many of your applications**

As an example of where an application's configuration might be found, let's say there was a clothespin design program called PinIt published by LaunderSoft, and the installed version was 1.2. The information for default editing parameters might be stored in the key: **`HKEY_LOCAL_MACHINE\Software\LaunderSoft\PinIt\Version1.2\EditDefaults`**.

This shows the recommended convention of storing configuration according to publisher, app name and version. Note that **`HKEY_LOCAL_MACHINE`** is a handle to a key, while all the others are actual key names. When opening this key, you could travel down the tree opening each level sequentially and using its handle to open the next, or you could simply specify all the levels at once.

Just as all software configuration is stored below the Software subkey, information about the hardware can be found below the Hardware subkey. Things such as CPU type, number of CPUs, bus configuration, etc can all be found in a standard place.

Exploring

If you want to explore around the Registry and see what's there, you can either use the program REGEDT32.EXE supplied with NT, or you can write your own program to traverse the tree of keys. How to do that? There are two functions that make discovering all subkeys and values or a key quite easy: **`RegEnumKeyEx()`** and **`RegEnumValue()`**. Once you have a list of the subkeys of a key, you simply use their names to open and enumerate them, and so on down the tree. To see the data of a value whose name you already know, use **`RegQueryValueEx()`**.

Figure 3 shows a routine that traverses the registry from a given top level key. Don't worry, I know that having **`printf()`**'s in a Windows program is kind of silly - this was the best way to show the Registry-handling code without clouding things up constructing list boxes, or filling up funny data structures. You can replace the **`printf`** code with whatever processing you prefer. I have completely ignored the issue of security here, too.

Modifying

Use **`RegCreateKeyEx()`** to create a new key and, if your program has proper security clearance, **`RegSetValueEx()`** to modify the values stored in a key. **`RegSetKeySecurity()`** can be used to change a key's security.

Compatibility Mode

No software based on a processor with a history like the 80386 would be complete without some form of backward compatibility. The Registry is no different. Applica-

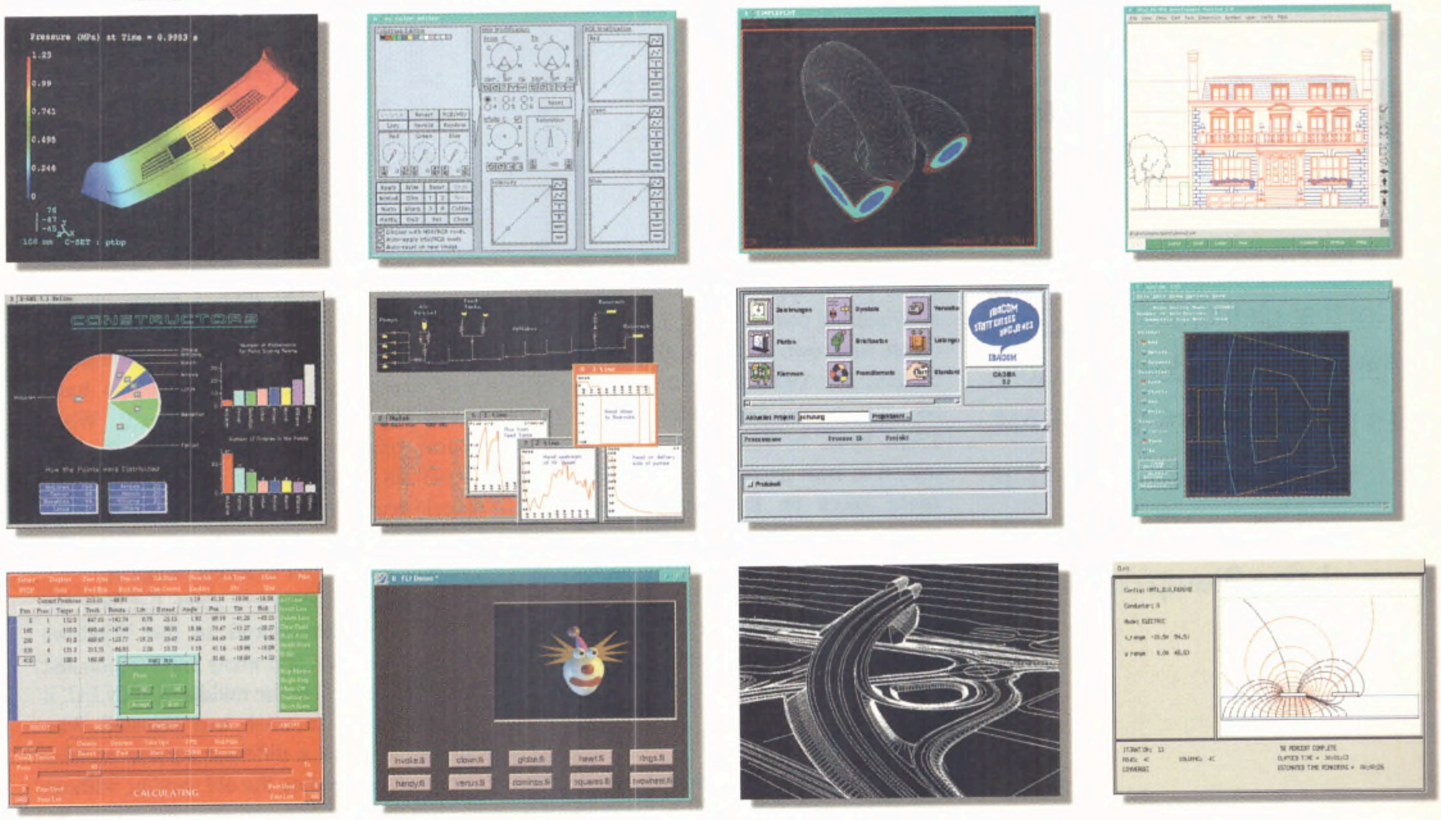
```
HKEY_LOCAL_MACHINE
  Hardware (R/O)
    Description
      CentralProcessor
      MultifunctionAdaptor
        DiskController
        DisplayController
        ...
    DeviceMap
      Keyboard Class
      Keyboard Port
      Parallel Ports
      ...
    ResourceMap
  Security (use security functions
to query these)
```

```
  Sam      (use security functions
to query these)
  Software (R/W)
    Classes
    Program Groups
    <company name> eg:
    Microsoft
      Word
        5.0
        6.0
      Excel
        5.0
    Aldus
    Lotus
  System   (R/W)

HKEY_CURRENT_USER
```

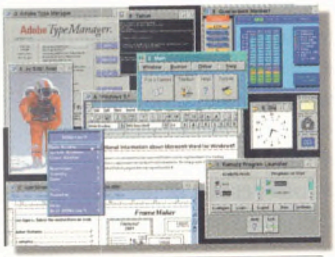
```
  Console
  Control Panel
  Environment
  Keyboard Layout
  Network
  Printers
  Program Groups
  Software
    <company name> eg:
    Microsoft
      Word
        5.0
        6.0
      Excel
        5.0
    Aldus
    Lotus
```

Figure 2 - Top Levels of Keys in HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER

Expand your market 20x.



New DESQview/X v2 lets you port X software to DOS systems in as little as one day.



Until DESQview/X came along, these software applications had a potential market of around 5 million users. Now they have over 100 million potential users, because DESQview/X allows X software to be ported to DOS in as little as a day.

DESQview/X is the definitive pre-emptive multitasking PC environment offering local and remote X Client & X Server capability, together with concurrent execution of both DOS text and MS Windows® programs. And DESQview/X boasts the unique ability to convert these DOS and Windows programs—unmodified —into X Clients for use by any X workstation.

DESQview/X allows X11, Motif and BSD socket-based applications to be ported with full 32-bit functionality (complete with virtual memory) whilst retaining the familiar framework of MS-DOS. And, with new DESQview/X version 2, everything but the network card is included for full DOS to X connectivity.

Phone Quarterdeck today for more information.

Quarterdeck

Quarterdeck UK Ltd., Widford Hall, Widford Hall Lane, Chelmsford, Essex CM2 8TD Tel: (0245) 496 - 699 Fax: (0245) 495 - 284
Quarterdeck International Ltd., B.I.M. House, Crofton Terrace, Dun Laoghaire Co. Dublin, Ireland Tel. (353) (1) 284-1444 Fax: (353) (1) 284-4380
Registered users of International English Editions: for after-sales technical support, voice contact (071) 973-0663 Fax: (071) 973-0664 9a.m. to 12:30 a.m.

➤ CIRCLE NO. 566

```
void TraverseRegistry(HKEY RootKey, int level=0)
  {
  // first learn how many values and subkeys are
in this key
  // variable names borrowed from example in
Win32
  // Programmer's Ref.
  CHAR achKey[MAX_PATH];
  CHAR achClass[MAX_PATH] = "";
  DWORD cchClassName = MAX_PATH;
  DWORD cSubKeys;       // # subkeys
  DWORD cbMaxSubKey;    // max name len
  DWORD cchMaxClass;    // max class len
  DWORD cValues;        // # values
  DWORD cchMaxValue;    // max name len
  DWORD cbMaxValueData; // max data len
  DWORD cbSecurityDescriptor;
  FILETIME ftLastWriteTime;

  RegQueryInfoKey(RootKey,
    achClass, &cchClassName,
    NULL, &cSubKeys, &cbMaxSubKey
    &cchMaxClass,
    &cValues, &cchMaxValue,
    &cbMaxValueData,
    &cbSecurityDescriptor,
    &ftLastWriteTime);

  // process the values and their data
  char *name = malloc(cchMaxValue+1);
  char *data = malloc(cbMaxValueData+1);
  for (int ct = 0; ct < cValues; ct++)
    {
    CHAR vtype;
    DWORD maxname = cchMaxvalue+1;
    DWORD maxdata = cbMaxValueData+1;
    RegEnumValue(RootKey, ct, name,
      &&maxname, NULL, &vtype,
      data, &maxdata);
    printf("%*s %s = ", level, "", name);
    switch(vtype)
      {
      case REG_SZ:
        printf("%s\n", data);
        break;
      case REG_DWORD:
        printf("%l (DWORD)\n",
               *((DWORD*)data));
        break;
      // etc...
      }
    }
  free(name); free(data);

  // Now print the name of each key
  // and traverse it.
  char *keyname = malloc(cbMaxSubKey+1);
  char *classname = malloc(cchMaxClass+1);
  for (ct = 0; ct < cSubKeys; ct++)
    {
    DWORD maxkeyname = cbMaxSubKey+1;
    DWORD maxclassname = cchMaxClass+1;
    RegEnumKeyEx(RootKey, ct,
      keyname, &maxkeyname, NULL
      classname, &maxclassname,
      &lastwritetime);
    printf("%*sKey: %s, Class: %s\n",
     level, "", keyname, classname);
    // recurse
    HKEY SubKey;
    LONG ret = RegOpenKeyEx(RootKey,
      keyname, NULL, KEY_READ, &SubKey);
    if (ret == ERROR_SUCCESS)
      {
      TraverseRegistry(SubKey, level+1);
      RegCloseKey(SubKey);
      }
    else
      printf("Error opening Key\n");
    }
  free(keyname); free(classname);
  } // TraverseRegistry()
```

Figure 3 - Traversing the Registry

tions using the old **GetProfilexxx()** and **GetPrivateProfilexxx()** functions to read information from INI files may actually be getting the data from the Registry instead. This is handled via associations stored in the Registry itself in the key **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping**.

There are even functions provided for writing out entire sections of the Registry to a text file and converting a text file back into Registry keys. This seems like a step backwards to me, but could be useful if you wanted to fax your configuration to a tech support person, for example.

Let us pause for a moment and imagine the elegance of an imaginary PC running Windows with no backward compatibility - this would certainly deal a stiff blow to RAM manufacturers and silicon wafer plants.

Registry Aftermath

Putting configuration information in the Registry solves many of the problems present with the Windows 3.1 INI file scheme. Configuration data is stored in a standardised, controlled fashion, taking up much less space per application. Provisions can easily be made for multiple users with different preferences. They still haven't fixed my favourite complaint though - if you trash your Windows directory, all your application configurations die too. We'll have to see what Cairo holds in store for us. In the interim we should still be careful as to what we do in the Windows directory.

For more details about the standard keys and values stored in the Registry, take a peek at Chapter 52 in the *Win32 Programmer's Reference, Vol. 2.* ■

Laine Stump is a software engineer at Morning Star Technologies in Columbus, Ohio who trashes his Windows installation with alarming regularity. He can be reached via email as laine@morningstar.com. Any example code mentioned in Laine's articles can be downloaded via anonymous ftp from ftp.morningstar.com in /pub/cpp.

“I spent a brain-frying day at Software Development 94. Unlike many conferences this one had a message...‘MULTIPLATFORM.’ The most powerful ‘MULTIPLATFORM’ message, however, came from Zinc Software...”

Peter Coffee
PC Week (US) 28/3/94

At Software Development 94, the world's largest software development tool show, Zinc Software's booth was jam packed with developers searching for object-oriented, multiplatform development tools.

Zinc® Application Framework has library and interactive design tool support for: DOS Text, DOS Graphics, Microsoft Windows (WIN16, WIN32s, NT), IBM OS/2, Apple Macintosh, UNIX character (Curses), OSF/Motif (on major RISC, Intel and Alpha platforms) from just one set of C++ source code.

Peter Coffee continued in his editorial; “But the most remarkable thing in Zinc...is its support for internationalization. It's the chance to offer your software to non-English speakers...outside the United States.” Zinc Application Framework supports single-byte and double-byte characters, 12 international languages, and 18 locales. Zinc even includes a 158 page manual with specific information on internationalizing your product. So regardless of whether your customer is running DOS Graphics and Motif in London, Microsoft Windows in Tokyo, or OS/2 and Macintosh in Frankfurt, Zinc is your object-oriented, multiplatform, fully internationalized solution.

To get Zinc's multiplatform message call for free technical information

Call 44-(0)81-855-9918 or Fax 44-(0)81-316-7778

z i n c

➤ CIRCLE NO. 567

Digital Assistant seeks applications...

Cliff Saran wanted more from his Newton... So this month he's tried his hand at developing applications for it.

As the Personal Digital Assistant (PDA) market blossoms it has become necessary for developers to make the transition from considering these items simply as mere gadgets to realising that they are genuinely useful devices. In the case of the Apple Newton, Apple has recently begun shipping a software developer's kit.

Getting started

The first thing to note is that it is not yet possible to do development on the Newton itself. The NDK is a kind of embedded systems development package. Apple recommends a Quadra with 8 MB of memory running System 7.x as the host platform. It has also said that it's working on a Windows version of the NDK too. The plan is to provide all supporting software for the Newton on both platforms. Connection between the Newton and the Mac is via a null modem serial cable. The link runs at 115,200 baud.

Newton speak

Programming the Newton is achieved through its own programming language called NewtonScript. This is an interpreted language which compiles down to a byte stream that must be squirted down the serial line into the Newton for running and testing the application under development. NewtonScript is object-oriented, with a syntax similar to that of Pascal.

On the object-oriented front it offers inheritance and support for exception handling: everything in NewtonScript is an object. Objects can contain methods and inherit directly from other objects so there is no need for the concept of a class or class instantiation.

To use Apple terminology, a frame is built from a number of *slots*. These are roughly equivalent to the fields in a Pascal record. However, as well as holding data, slots can also store methods. Two types of inheritance are possible. The first is known as prototype inheritance. Here, one object can encapsulate completely the behaviour of another object. A frame inherits those slots from its prototype which it does not already contain. The prototype is placed in a special slot in the frame which is referred to as _proto.

> **Today, the NDK provides the only way to develop Newton applications**

The second form of inheritance is *parent* inheritance which enables hierarchies of objects to be created. The `_parent` slot in a 'child' frame is used to specify a parent.

Painting an application

The starting point for developing an application is a new project in the Newton Toolkit. This is the IDE for building and debugging NewtonScript. The first step is to create the layout, or *view*, of the application using the graphics editor in conjunction with a palette of what Apple calls *templates*. These can be considered as 'position holders' for objects which will be created at runtime. Along with visual elements of the application such as push buttons, input lines and radio buttons, the palette also includes templates for the application itself and Newton services such as fax, modem and AppleTalk connectivity.

The first template which must be placed on the design is that of the application base view. This determines, amongst other things, the actual area of the Newton screen that the resulting application will take up. The remaining controls are subsequently placed within the application frame. Once the layout is completed it is saved as a layout file and can be previewed within the Newton Toolkit. Each layout file can only contain one view but several can be added to an application to provide multiple screens.

Down to the code

Once the application's screen layout has been designed it can be customised using the browser window. This allows the developer to access the templates in the layout file in order to change properties or write methods associated with the view. For instance, the `Visible` property of an input line can be toggled.

A method is used to enable the application to respond to an event. So, if a push button is depressed then the `buttonClickScript` method will be invoked. Using the browser the developer can add NewtonScript to handle this event.

Remote inspecting

When the application is finally ready for testing it can be compiled and then transmitted to the Newton. The Newton Toolkit can be set up to perform this download automatically on completing a successful compile. On the Newton a new icon appears in the Extras folder: the application is ready to be run. Back on the Mac, the Inspector window provides the debugging facility of the NDK. Here, it is possible to issue NewtonScript directly to the Newton. It can also be used to enter debugging commands and browse the object storage system on the Newton. Debugging support includes viewing the call stack, tracing function calls and setting breakpoints.

If only it had...

Today, the NDK provides the only way to develop Newton applications. But it does the job well. The graphics editor and browser combined with the power of NewtonScript should enable developers to create applications relatively easily. There is only one drawback with the product today, that being the lack of an interactive source-level debugger. The only other feature which would have made the NDK complete is a NewtonScript emulator for the Mac. So Apple, what do think? ■

If you are interested in writing applications for the Newton, EXE is organising a seminar on the 4th August at Apple's headquarter in Uxbridge. The seminar is being offered exclusively to EXE readers. Turn to page 46 now for details on registration. Hurry, there's only 65 places left.

Don't even think about using a Relational Database System !

Comparing Object Oriented and Relational Design Methodologies

Feature	POET ODBMS	Relational RDBMS
Storing Objects	As Objects	Break Objects into Tables
Database Model	User Application Model	Separate Database Model Required
C++ Integration	Total	Poor
Database Operations	At Object Level	Must Write Code
Productivity	Increased	Reduced
Complex Object Performance	Excellent	Poor

Use the POET Object Database for C++

C++ and class libraries have made the GUI development much easier. After all, it is only logical that more and more developers think in objects. But object orientation shouldn't end at the user interface programming level.

The Problem: Without POET, a C++ programmer must use flat files or a RDBMS to store objects. He has to write code to overcome the mismatch between the application and the database model. This leads to design restrictions, performance penalties and more code to write and maintain.

The Solution: POET operates at the object level, it speeds up the development process and provides greater performance. Furthermore, the developer can simply take the object-oriented application design in C++ and map it 1 to 1 into the database. Without any compromise at all!

Full featured database: POET provides complete support for Encapsulation, Inheritance, Polymorphism and Containers as well as queries, object locking, and transaction handling .

True cross platform support: Complete interoperability makes the development of network enabled applications a breeze. POET supports Windows 3.1, Windows NT, Win32s, Windows for Workgroups, Novell Netware (NLM), SUN, AIX, HP-UX, SGI, SCO, Macintosh, Power Macintosh, OS/2 and NeXTStep.

Call 081 317 7777

POET Software

Silicon River

Silicon River Ltd •106-108 Powis Street • LONDON SE18 6LU • Tel: 081 317 7777 • Fax: 081 316 7778

➤ CIRCLE NO. 569

Initial thoughts

It's funny how often developers rely on the compiler to wean out problems with C and C++ initialisations. **Francis Glassborow** lays down the rules...

```
#include <stdio.h>
int i=10;
const int ci=10;
int j=j;
int k=i;
int m=ci;
int array[ci];
main()
{
  int i=i;
  int local[ci];
  int j;
  j=j;
  printf("%d",i);
  return;
}
```

Figure 1 - An ill-formed C program

I wonder how well you understand the rules for initialisations in C and C++. I have yet to meet someone who always gets them right first time. I should make it clear that I too have made more than a few mistakes. Generally C compilers manage to detect errors as well as issue warnings for undefined behaviour such as using a value of an object before initialising it. This leads to a dangerous reliance on compilers because, as we shall see, initialisation in C++ is much more complicated.

C behaving badly

First of all, consider the ill-formed C program in Figure 1 and decide what faults there are with it. The only global statements that are correct are the first two. C requires that globals shall either be initialised with a constant initialiser (and that does not include `const` variables). All the other global variable declarations above are breaches of a constraint and therefore require a diagnostic from the compiler - none of them constitute undefined behaviour.

Dodgy autos

Now let's look at the declarations of the auto variables. The declaration of `i` is initialised with itself. Given that the global `i` is hidden as soon as the local identifier is complete, the behaviour of the program at this point in its execution is undefined. That is, the compiler is not required to provide a diagnostic and can do anything it likes with the statement. The declaration of `local[]`, on the other hand, is an error requiring a diagnostic because its size is not a C compile time constant. The declaration of `j` is fine but the assignment on the next line is undefined behaviour because the value of `j` is used before it has been initialised. Unless `i` has been properly initialised the `printf()` statement has undefined behaviour.

What, no warnings?

A return statement from `main()` without a value is also undefined behaviour because there are only two permitted prototypes for `main()` in ISO C, both return an `int`. Oddly, modern compilers normally diagnose the absence of a return statement but fail to notice one without an expression - they are the same category of fault in ISO C. One thing that I find curious is that all modern compilers successfully issue a warning for '`j=j;`' but none of them issue warnings for the equally erroneous, and even more easily detectable, '`int  i=i;`'. Perhaps a compiler implementor would like to comment.

The compiler arranges for global and local static variables to be statically initialised before entry to `main()`. This means that it can only use addresses (lvalues) and literals. C considers a `const` object to be a variable with an added limitation on its use. This means that its value cannot be used by the compiler for static initialisation. C specifies that all global variables that are not explicitly initialised are implicitly initialised with zero (null pointer for pointers). There is no implicit initialisation of auto variables for efficiency reasons (programs should not take time doing default initialisations of auto variables unless the programmer wishes it.)

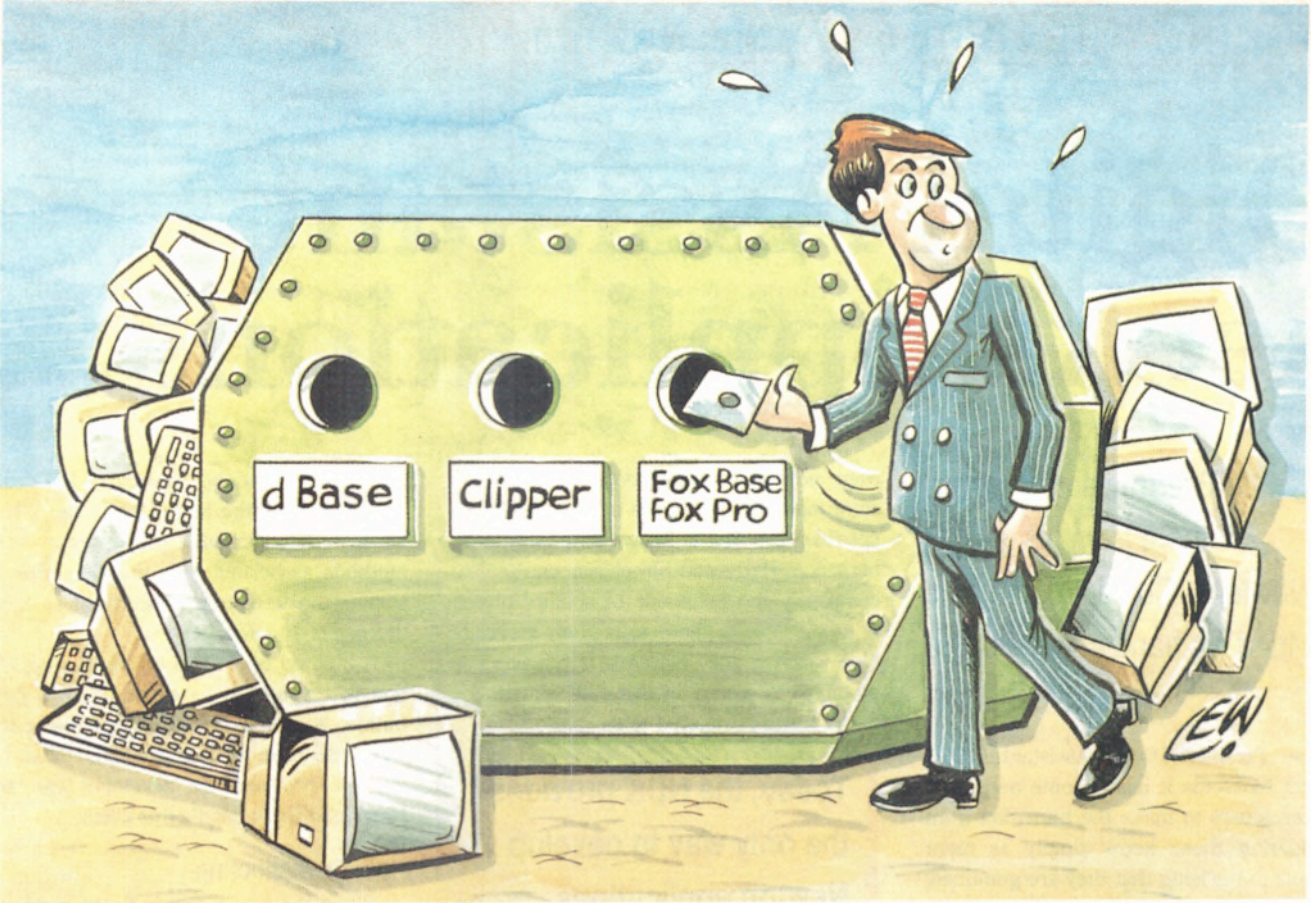
The last point to clear up is why an identifier enters the current namespace as soon as its spelling has been completed and prior to initialisation. This is to allow the following idiom:

```
struct node
{
  int i;
  struct node * next;
  test = {1, &test} ;
}
```

Note that we can use the address of test before it has been initialised because the compiler can determine where it will be, even if it is still deciding what it is.

C++, another language

Now let's move to C++. The rules for initialisation are almost entirely different - they have to be. A C++ compiler will give you one warning and one error for the program listed in Figure 1. The warning will be for the statement '`i=i;`' in `main()` as a possible use before initialisation. The error will be for the `return` statement. C++ has upgraded (rightly so) the failure to return a value when the prototype states that there will be one. Some programmers and most compilers allow `main()` to be written explicitly as returning a `void`. The opinion of most experts I have consulted is that this is



DON'T THROW AWAY YOUR INVESTMENT... ...RECITAL IT

Thousands of organisations throughout the world have realised that they don't have to throw away their investments in Xbase (e.g. dBASE, FoxBase, FoxPro and Clipper), when moving to mini-computer environments.

They discovered RECITAL!

With RECITAL you can move your Xbase applications to Digital, HP 3000 and almost any UNIX computer with little or no re-programming.

The record is currently 12 minutes.

Migrated applications immediately benefit from RECITAL's Automatic Locking and Concurrency Control, Referential Integrity, Integrated Dynamic Data Dictionary and Client/Server Architecture.

Extending the life of applications is just one way that RECITAL gets the power to your users more quickly.

RECITAL's 4GL makes new system development and enhancement just as easy.

So, whether you have existing applications that you would like to re-use or just need the fastest development environment available, take a look at RECITAL.

Recital
CORPORATION

Recital Corporation: Ringside, 79 High Street, Bracknell, Berkshire RG12 1DZ. Telephone: 0344 301444 Facsimile: 0344 300689

For more information on how you can use RECITAL to your advantage fill in and cut out this slip and return it to us today!

□ I would like to receive further information on RECITAL □ Please call to arrange a demonstration □ I don't believe it - prove it to me!

Name ______________________ Position ______________________

Company ______________________ Address ______________________

______________________ Postcode ______________________

Telephone ______________________ Facsimile ______________________

➤ CIRCLE NO. 568

wrong and not an intended meaning of the section on `main()` in the *Annotated C++ Reference Manual*.

Another feature of C++ is that `main()` isn't a true function. Unlike C, it does not have an address and cannot be called. Fundamentally, `main()` in C++ acts as a marker of the point at which all global, class static and static auto variables must have been initialised. The end of `main()` denotes the earliest point at which the above variables may be destroyed. Actually even this is not quite true because the compiler must only compile your program so that the result will behave as if the above were true.

Order of initialisation

We have one other guarantee: destructors will be called in the reverse order of constructors. Because explicit constructor and destructor functions are written by programmers, user written code is run both before and after `main()`. This leads to a number of serious problems. The best known of these is 'the order of initialisation' problem. How can you be certain that an object such as `cin` or `cout` has been constructed and initialised before the first time you use it? Similarly, how to ensure that they are not destroyed before the last time your destructors use them. There are some guarantees about order of initialisation within a single source code file (translation unit) but there are currently no guarantees about the order of initialisation of static objects in different source code files.

C++ has several phases of initialisation of global and static variables. The first is akin to C's initialisation phase but C++ then proceeds to traipse through various (necessary) forms of dynamic initialisation. Sometimes constructors will be called that get initialisation data from an input stream (even the keyboard). One particularly vicious item is the copy constructor (which the compiler will provide if you do not). Consider the code fragment in Figure 2 and decide if it depends on undefined behaviour or not.

```
class Node {
  char * message;
  Node * link;
  // the rest of the private
  // interface
public:
//default safe constructor
// copy constructor ?
  Node();    Node(Node & l);
  // the rest of the public
 //interface
};
Node root(root);
```

Figure 2 - Copy constructors are particularly vicious

Look at that last statement. Can you decide if it gives undefined behaviour or not? If you cannot, how can the compiler? Remember that neither the compiler nor you may have access to the source code for class `Node`. If you are in any doubt I offer you the alternative definitions in Figure 3 for the `Node` copy constructor.

Who's to blame?

These are only provided as minimalist examples of the problem and not as exemplars of real code. The first of these is not a true copy constructor from a semantic viewpoint but how is the compiler to know? There are many programmers who (carelessly) write such 'copy constructors' in the context of complicated data structures. The first of these is also entirely safe as a constructor for self construction and does not rely on any undefined behaviour. The second is likely to result in dire consequences when that random value for link gets used. I can provide similar examples of dangerous abuse with assignment. We all know how to protect our own versions of assignment from accidental self assignment (trap it and return immediately). We could, if we deemed it necessary, trap self initialisation and throw an exception. The problem with both these is that it delays detection until run time. Compiler implementors can easily identify such suspect code because it will be of one of three forms:

> **One particularly vicious item is the copy constructor (which the compiler will provide if you do not)**

```
// explicit self assignment
xxxx=xxxx;
// self-initialisation
sometypename xxxx=xxxx
// explicit call of copy
constructor
sometypename xxxx(xxxx)
```

I believe that such code is always suspect and we should expect any quality compiler to provide a warning message in all such cases. I have yet to find one that does so in all three cases. Most issue warnings in the assignment case if no explicit initialisation has been done, though a user written default constructor may have done all that is needed to avoid undefined behaviour in this case. Some will argue that, as in the Node example, apparent self initialisation may not be an error so we should not insist on a warning. I believe that all such code is dubious and if you must abuse copy constructors you should expect to get warnings. After all it is about the only chance that the compiler has of hinting at such abuse.

```
Node::Node(Node & l)
{
  message=new
    char[strlen("New node")+1];
  strcpy(message, "New node");
  link = &l;
}


Node::Node(Node & l)
{
  message=new
    char[strlen(l.message)+1];
  strcpy(message, l.message);
  link = l.link;
}
```

Figure 3 - 'Fix' for copy constructor

Sound advice...

In C, static objects (globals and static autos) can only be explicitly initialised with constant initialisers. Static objects that are not explicitly initialised will be implicitly initialised - so it is impossible to have an uninitialised static object in C. Using the value (rvalue) of an auto object before the object has been initialised either at declaration point or through an assignment is undefined behaviour.

Now C++. For those that do not know how to handle constructors that take the same type as the only parameter, the answer is simple; use a pointer to type as the parameter. The compiler will not then view the constructor as a copy constructor. Do not mess with the semantics of copying and assignment because down that path lies a madness of which problems with initialisation are but a symptom. ■

Subscriptions: individual £12, student£6, corporate £75, Overload & C++ SIG £15 (+membership). For further information about ACCU write to Francis Glassborow, 64 Southfield Road, Oxford, OX4 1PA, ring 0865 246490 or email francis@robinton.demon.co.uk.

System Science

London's specialist supplier of Software Development Tools for DOS, Windows and Unix

UNIX

SCO Unix Op. Sys   **Open Desktop** - NEW 3.0
SCO Development Sys **SCO** TCP/IP & NFS
Informix 4GL & SQL   **LPI** Fortran
LPI C++ SCO, SUN, HP/UX   **C++ Views** & GUI
Zinc Framework - Motif   **Zapp for Motif**
PC-NFS   **FTP** PC/TCP
... many more products for Intel Unix , Sun, HP, DEC Risc etc

Mathematics

Derive & Derive XM new licenses, lower prices
Mathematica 2.2 Win, Mac, Workstations, Student ver.
MathCad - New 5.0   **Lindo** Linear Prog
What's Best   **SigmaPlot**
SigmaStat, TableCurve   **Origin 3.0**
Unistat, Rats, Systat, PC Matlab for Win ... call

Windows Development

(see Microsoft , Borland, Symantec, Watcom)

Data Entry Workshop by Turbopower	£139.00
TCXL Developers Wks -Multiplat	£call
Smalltalk/V Windows	£295.00
PCX ProgToolkit Windows	£169.00
SmartHeap Windows	£345.00
INSTALLShield ver 2.0 (also for NT)	£345.00
Farpoint Professional Toolkit	£239.00
Spread/VBx++	£169.00
PI-FileFind Control	£375.00
Graphics Server SDK	£call
Doc-To-Help - Help file prep with Word	£call
Q+E Database Library Dev Sys	£455.00
Btrieve for Windows	£call
Lead Tools Imaging Library	£call
High Test - auto- test Windows apps	£345.00

VB Value

Distint TCP/IP Visual Edition	£179.00
Spyworks by Desaware	£115.00
Spread/VBX	£169.00
TrueGrid Pro	£115.00
Q&E Multilink/VB	£275.00
CodeBasic 5.1with New CodeReports	£139.00
VB Assist	£139.00
Designer Widgets - NEW	£99.00
Grid/VBx, Tab/VBx - NEW	£call
VBTools /Win4.0 - new ver	£110.00
Pinpoint	£135.00

CrossDevelopment

Hitech C Cross Z80, 6805, 68HC11..	£495.00
Introl C Cross- 68K, 630x, 6809 etc	£1445.00
Aztec C68K &new Remote Debugger	£call

2500AD Cross Assemblers and Simulators

For Fortran

Lahey F77L-EM/32	£595.00
Interacter Screen & Graphics Lib	£call
For_C - translator	£call
Microsoft Fortran 5.1 for DOS	£145.00
Microsoft Fortran PowerStation	£239.00

32 bit Fortran power for 386/486 Dos & Windows

MicrosoftFortran PowerStation Win NT	£call

Call to upgrade to Microsoft's latest Fortran products

- Please call if the item you are looking for is not listed.
- Call for our comprehensive catalogue
- Prices are exclusive of VAT.
- Prices (except upgrades & specials) include delivery to mailand UK. Sameday London delivery at cost.
- Prices are subject to change - please call to check.
- VISA, ACCESS & Mastercard credit cards welcome with telephone orders.

E&OE

Microsoft

Visual C++ ver 1.5, CDROM only	£299.00
Visual C/C++ - 32 bit Ed CD*NEW*	£299.00
Visual C/C++ PE 1.0 - 3.5"	£call
MultiMedia Viewer Pub Kit	£239.00
Windows for WorkGroups 3.11	£call
Windows NT	£call
Macro Assembler 6.11 *for NT & DOS*	£110.00
Visual Basic Windows 3.0 PE	£235.00
Visual Basic for MS-DOS Prof	£235.00
MS Project for Windows	£375.00

Microsoft FORTRAN - *see **Fortran** section*

Visual C++ 1.5 on CD-ROM from VC++ PE , £99.00 + ship + VAT

Visual C++ 32-bit Edition special offer from VC1.0 PE

Authorised Languages Dealer

Borland

Borland C++- NEW Windows version 4.0	**£295.00**

*Upgrade (incl comp. upg)to Borland C++ 4.0 - **call for details***

Turbo C++ Visual Edition Windows	£85.00
Turbo C++ 3.00 DOS & World of C++ Video	£79.00
Visual Solutions Pack	£59.00
Turbo Assembler - *NEW* Dos, Win etc	£59.00
Paradox Engine & DFX	£125.00
Brief Programmers Editor	£125.00
Borland Pascal & Turbo Pascal	£call

Authorised Languages Dealer

Nu-Mega

BOUNDS-CHECKER/ DOS - 386 etc	£159.00
BOUNDS-CHECKER/W	£195.00
Soft-ICE/W (Windows)	£295.00
BOUNDS-CHECKER/NT -*NEW*	£195.00

Special pricing on Nu-Mega combi packs

PVCS Version Manager

Available for DOS, Windows, OS/2 and many Unix platforms, PVCS supports multi-user development teams on cross-platform projects. Develop with security, knowing that PVCS is archiving your code.

Call System Science - your PVCS Specialist

MKS

MKS Toolkit -Unix tools DOS & Win	£210.00
MKS RCS - Version Control, new DOS/WIN	£275.00
MKS Lex & Yacc *for C, C++ & TP*	£215.00
Internet Anywhere - NEW	£129.00

multi-user licenses available for all MKS products

Editors

Multi-EDIT Professional	£129.00
CodeWright Editor for Win or NT	£185.00
Ed for Windows Editor- new 2.0	£145.00
Visual Slick Windows prog editor	£call

Tools

Easy Case Professional -DOS or WIN	£call
MetaDesign - Flowcharts, DFDs etc	£195.00
PC Logo for Windows - ***NEW***	£60.00
Dan Bricklin's Demo II - prototyping, demos etc	£195.00
RTLink Plus	£325.00
PC Lint C/C++ - new ver 6.0	£149.00
Source Safe DOS/Win version control	£295.00

WATCOM

C/C++-new 10 .0 - 16 & 32 bi, GUI IDE	£375.00
Fortran 77 ^{32} - 9.5 for Dos32,Win, ADS etc	£375.00
SQL Dev - Win & ODBC or DOS - new ver	£275.00
SQL Servers for DOS, Netware & OS/2	£call
VX*REXX Visual REXX for OS/2 - ver 2.0	£129.00

C/C++

Symantec C++ PE 6.1 16 & 32	£call
Metaware High C/C++ 386 - DOS & Win	£call

C++ Libs

Tools.h++	£call
Zapp -C++ Interface for Win, DOS, OS/2, Motif	£call
C++/Views - Win, OS/2, Motif - New Ver 3.0	£call
Zinc Interface -DOS, Win or OS/2	£call
Comm++ by Greenleaf	£189.00

C Libs

CodeBase or CodeBase++- new 5.1	£245.00
Ctree Plus from Faircom	£call
C Asynch Manager	£139.00
Greenleaf CommLib - new 5.0	£225.00
Asynch Prof C/C++ by Turbopower	£169.00
PCX Programmers Toolkit	£169.00
GX Graphics new ver	£169.00
GIF Programmer's Toolkit new ver 6.0	£169.00
MetaWindows - DOS/286/386	£call
Install Professional	£245.00
BGI Printer Toolkit	£call
Bar Code Library- source	£375.00
Slate Printer Drivers	£call

... many more tools

OS/2 Too

OS/2 for Windows	£49.00
Borland C++ OS/2 1.5 with OS/2 Win	£149.00
IBM C Set++ - 3.5" or CD	£call
DB2/2, Visual Age from IBM	£call

CASE

EasyCASE System Designer 4.1new WorkGroup ed
EasyCASE Professional 4.1new WorkGroup edition

Pharlap DOS Extenders

TNT DOS-Extender 32 bit SDK - new	£339.00
286 DOS-Extender SDK	£339.00

Databases

Access 2.0 by Microsoft (upgrades avail also)	£289.00
Fox Pro - NEW 2.6 - Std, Pro & Upg	£call
dBASE Windows NEW Upg. avail	£call
CA-Clipper	£call

What's New!

Watcom C/C++ 32 ver 10 for 16 & 32 bit dDOS, Windows, Win 32, OS/2 - special introductory price for CD-ROM only - **CALL**

SmartHeap DOS, Windows & NT increase application speed, debug memory allocations

PowerBuilder Desktop £call
build Windows Client Server Apps on your Desktop

DemoQUICK Windows DemoBuilder £395.00
Mimic captures your demo, annotate with textgraphics etc, Royalty free

CA-Realiser - Windows & OS/2 - special £99.00

3-5 Cynthia St
London N1-9JF
Fax: 071 837 6411

071 833 1022

Committed to Excellent Customer Service & Support

➤ CIRCLE NO. 570

The OpenDoc Experience

Paul Smith has just returned from the Apple Worldwide Developers' Conference in San Jose where everyone kept talking about OpenDoc. So what's so great about it?

> **the designers realise they cannot, and should not, try to anticipate all possible human interface and component interactions**

I recently returned from California, where I attended the 1994 Apple Worldwide Developers' Conference (WWDC), in San Jose. I'm limited in what I can report from the conference because of the non-disclosure agreement every attendee must adhere to, so I'll stick to topics that Apple says are open for public discussion. The main topic at the conference, for many attendees, was OpenDoc, the focus of this month's column.

A WWDC update

The conference is always worth attending: it's worth finding the money to get there and pay the conference fees ($1100) and it's well worth making the time to do it. This is partly for the information content: there's nothing to beat attending discussions of new technologies led by the engineers who developed them. It's also for the chance to meet the Apple engineers and products managers in person, to understand better the mind-sets that led them to developing the systems software and hardware of today and tomorrow. I have a friend who works at a Macintosh software house, who decided not to attend the conference on the basis that he already subscribes to all the magazines, gets all the documentation and development software, and as a contractor to Apple his firm receives all sorts of stuff well in advance. Big mistake! He still missed out on the best opportunity to meet and understand fellow developers and Apple staff that is available.

This year I was a speaker, again, this time talking about my company's new advanced script editor and debugger for AppleScript and the Open Scripting Architecture (OSA), ScriptWizard. I was also called up to the podium to answer questions on OSA development, after one of the technical sessions: which underlines the risks of sitting too near the front at a conference. It's not quite as much fun being a speaker as it is being an ordinary attendee, because the time needed for preparations means one has to miss some sessions (although it does have economic advantages). I'd happily do it again, though.

To get an invitation to the WWDC, which is held in May every year, you have to be registered with Apple as a developer. I gave details of the Apple schemes last month, so I won't repeat them all here, but you do have to be an Associate developer (which costs £325 per annum) or a Partner (£975) to qualify for an invitation. Call the Apple Developer Programme on 0800 505094 or 0506 401533 for full details.

Developer programmes Pt II

Last month I promised an update on who's actually running the new Apple developer programmes. Well, it turns out that the nice Scottish lady who answers the 0800 number works for McQueens, which is long established in the Apple market as dealer and distributor. As I thought - not Apple employees. Unfortunately, this means they can't help if you want to arrange an appointment with someone at Apple UK; they simply have the job of answering the phone, sending out information packs and monthly mailings, and doing the accounts.

As for hardware purchase privileges, I'd hesitate to say the scheme is dead, but that's not far from the truth. To start with, just after I wrote last month's column Apple UK announced that the Apple Direct sales organisation through which developers were supposed to buy their hardware (excepting PowerMacs), was being closed down. The latest scheme is that developers are supposed to purchase hardware from Apple's trade distributors, Frontline and Ingram, at trade price. The trouble is, the standard trade price offered by these distributors is a bit higher than the regular street price for most Apple hardware. Not only that, but dealing through a trade distributor means there's nobody approachable (or local) to turn for pre-sales and post sales support.

OpenDoc, the software

Okay, enough of that carping. Let's take a look at OpenDoc, about which we've been hearing a lot of noise recently. I can't show you everything under the hood yet, but let's

3.5"
EXABYTE
TAPE
DAT
QUARTER INCH
CD-ROM
COMPACT disc DATA STORAGE

have a peek inside the OpenDoc hype and have a look at what it's all about (see Fig 1).

OpenDoc is being pitched in the technical press as a new technology: a compound document architecture, which makes it sound complicated. There is a simpler way to look at it: OpenDoc is a cross-platform run-time framework that makes it easier to develop, deliver and use applications software. It is intended that existing applications will be able to make the transition to work within the OpenDoc framework incrementally. OpenDoc is not the traditional sort of object-oriented framework that enforces consistency (and thereby puts up a barrier against development and adoption) but one that concentrates on making sure that desirable things are possible. That OpenDoc is a compound document architecture is a distinct bonus, which will have increasing importance over time, but let's take a bottom-up look at the other benefits of OpenDoc first, to put the subject into perspective. Even if you don't care about helping users build compound documents, you should be interested in what OpenDoc has to offer.

To the user

From the user's point of view, OpenDoc provides the framework for the creation and manipulation of multimedia documents, also known as compound documents. The user model is thus content-centered rather than program centred. The OpenDoc runtime provides a framework within which exist:

- parts - the boundaries within a document at which one type of content ends and another begins
- part handlers - providing content editing and viewing functionality
- storage for the content represented in the parts
- visual and printed layout of the components.

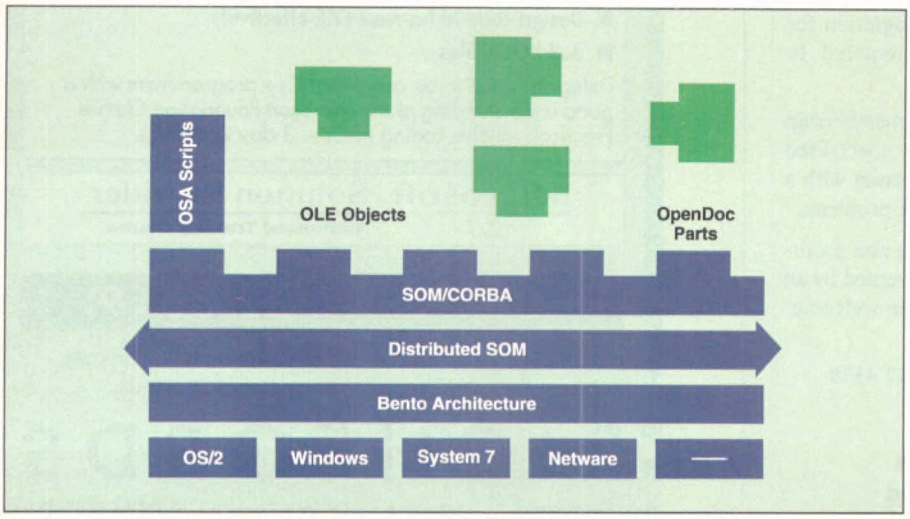


Figure 1 - The OpenDoc Architecture

The OpenDoc framework provides for customisation, through scripting and plug-in extensions, and through allowing users and developers to construct compound user interfaces from the components provided by OpenDoc parts. OpenDoc also provides for collaboration services, by handling platform-specifics of mail, inter-part collaboration, shared documents and document interchange. Don't read all this as implying that an OpenDoc document has to contain multiple parts: a document that contains only a single part is perfectly okay, if that's what you and the user wants.

A question of plumbing

The OpenDoc runtime environment provides plumbing for most of the services that are required by applications software, through a platform-independent API that's defined using IBM's System Object Model (SOM), callable both from object oriented languages like C++ and traditional ones like C. The plumbing works in conjunction with platform-specific services. It can be thought of as an extended toolbox for constructing applications: comparable in intent to the sorts of services provided by (on the Mac) a combination of the Toolbox ROM and the MacApp framework, or (on Windows) by the Windows APIs and the Microsoft Foundation Classes or Borland's Object Windows Library.

The OpenDoc runtime is provided in shared libraries, so the runtime code can be shared between applications that use it. The above makes OpenDoc attractive as a software platform for the development of applications (known as *part handlers* in OpenDoc parlance, I'll explain why in a moment), whether or not they make use of the full range of OpenDoc facilities.

Storage hierarchy

Underlying everything in OpenDoc is the storage system, which presents its clients with an API that the OpenDoc team insists is not an object oriented database, although it shares some characteristics with that class of storage system. The OpenDoc storage model appears hierarchically organised (although it can be used to construct differently organised databases). At the top level is the storage system itself, which contains a number of container objects.

Each container object contains zero or more document objects. Within each document is one or more draft objects. In turn each draft object contains a number of storage unit objects.

A storage unit contains a list of properties; each property can contain one or several values, each of which is strongly typed. Values are streams, like traditional file system files. These can contain simple data values or complex data streams like word processor text or pictures. The storage system also provides plumbing that's needed to support linking of OpenDoc parts, the clipboard for copy and paste operations and drag and drop. Although it's expected that most OpenDoc part handlers will use the OpenDoc storage system for persistent storage, there's nothing to stop a part handler using external services if appropriate.

Parts and their handlers

The OpenDoc user model is object-oriented, in that documents are objects that contain other objects called *parts*. Parts are distinct and self-contained objects that have contents, properties, and behaviour. The contents and properties are maintained as property values by the storage system; the part-specific behaviour is defined by the part handler, which is equivalent to an applications program. The OpenDoc imaging system manages the geometry of the parts of a document, describing and arbitrating the shape and size of the frames in which part handlers represent the content they are responsible for. Part handlers use platform-specific (or cross-platform but independent of OpenDoc) facilities for drawing and editing inside the frames that delineate the parts they are responsible for, but OpenDoc manages the placement of and interactions between the parts.

Part handlers can be part editors, or they can simply be part viewers. The OpenDoc team recommends that part developers create both an editor and a viewer, so that the editor can be sold commercially and the viewer distributed freely, so content can always be viewable whether or not the user has the appropriate editor. This is not a simple issue, of course: it does not address the matter of how the part viewers are going to be distributed, but it does at least make a step in the right direction.

Makeup of a compound doc

The imaging system constructs document geometry from three primitives: a *canvas* that provides the context for drawing, a *shape* that describes a geometric (regular or irregular) area of a canvas and a *transform* that uses platform facilities to translate or perform more complex manipulations of shapes. The OpenDoc imaging system maintains a global window state, a collection of window objects. The windows are platform-specific constructs. Each window holds a *facet* object, a shape that specifies the structure of the layout. A facet is distinct from the *frame* object that may be visible in it; the facet also maintains the containment and depth-ordering of embedded frames. Facets can also be used to embed an off-screen canvas and can be used to arrange the contents of any window or printable layout. Frame objects handle drawing update notifications and act as a repository of geometric information for each part. The geometry is that resulting from the interaction between a part's container and the embedded part being displayed in the frame. Facets and frames are not required to be based on the primitive shape object, they can have any form permitted by the platform's graphics engine.

Parts can be displayed in more than one frame at a time. These frames can have more than one facet. The OpenDoc imaging system will take care of asking all of them to draw, which means your code doesn't need to know anything at all about all this stuff. It only needs to know how to draw and handle user-initiated events inside the frame. During the drawing process, a part handler is sent a message telling it to draw a facet; it asks the facet and frame to provide the geometry (the clipping, transformation, and layout) and then it calls a graphics engine of it's choice to do the drawing. Naturally, having OpenDoc manage everything isn't the only way to do it and parts that need to perform asynchronous drawing (such as movies) are fully accommodated. But, if you're moving a traditional application into the OpenDoc world, you don't need to worry about all these complexities up-front.

Getting the message

OpenDoc dispatches user-initiated events, like mouse clicks, menu selections and keystrokes, to the appropriate part handler. The OpenDoc dispatcher is designed to be extensible, able to accommodate new types of event and be portable across platforms. Part handlers don't need to know anything about the dispatcher. They are simply passed events by it. It is possible for containers to register themselves with the dispatcher so that they get passed those human interface events that are not handled by their embedded parts. The OpenDoc arbitrator manages shared resources, such as menus and the menu bar and the stream of keystroke events. It too is extensible and portable, like the rest of the OpenDoc framework.

> **OPF is based on the remains of the Bedrock cross platform applications development framework project, which failed famously last year**

Built to grow

The OpenDoc architecture permits extensions, because the designers realise they cannot, and should not, try to anticipate all possible human interface and component interactions. The OpenDoc architecture includes a formalised model for extensions, so that all significant components have a built-in extension mechanism, and so that it is not required that classes should implement, or allow clients access to, extensions.

One significant extension, that is likely to be supported on most OpenDoc platforms, is that for *semantic events*. This extension permits the OpenDoc shell, and those part handlers that support semantic events, to be remotely controlled. The tasks thus performed are not defined using programmer-level abstractions, but by user-level abstractions that deal with objects in the user's view of things. Thus, the semantic events subsystem describes a task in terms of *what* the user wants to be done, not *how* the underlying program code interprets the task and the data it relates to.

Semantic events are directed using *object specifiers* and are dispatched by a component of OpenDoc called the name resolver. The foundations of the semantic events subsystem were laid in the Apple Event Object Model, which we discussed in *EXE* in September '93. The semantic events subsystem makes it possible for OpenDoc documents and the parts they contain, to be extended by user-level scripting. On the Macintosh platform, for instance, OpenDoc documents will be fully scriptable and extensible using AppleScript and other Open Scripting Architecture languages.

In the OpenDoc runtime environment, parts are bound to part handlers. This binding is controlled by the type of the part. This is managed by a registration scheme, inside OpenDoc, that relates part handlers to the part types they can deal with. The registration scheme also allows for users to replace part handlers with others that can handle the same types of content.

The shell

The portion of OpenDoc that holds everything else together is the *shell*, which is responsible for creating and initialising a *session* object, which is the root container for all the documents and other objects controlled by OpenDoc while it runs. The shell is also responsible for opening the document specified by the user using whatever human interface is appropriate to the platform, such as the Macintosh Finder, and for accepting human interface events from the platform's operating environment and passing them to the OpenDoc dispatcher. The shell is not formally specified as part of the basic OpenDoc design because it is so platform-dependent.

Multiple platforms

OpenDoc is intended to be cross-platform. Right now implementations for the Macintosh, for AIX, OS/2 and MS-Windows are all under construction. At the time of writing, the Macintosh version is in alpha test and will be beta soon after you read this article. Apple has stated that when the OpenDoc software goes beta, it will be distributed to all Apple developers who are registered as Associates or Partners. So, you won't have long to wait before you can start developing for OpenDoc.

Tools for development

On the Macintosh, to develop OpenDoc parts, it is not 100% guaranteed that everything development environment will do the trick, but I think it's a fairly safe bet that all the leading ones will. The key points are that the environment must be able to include the OpenDoc header files, which being defined using the SOM Interface Definition Language (IDL) is expected to be portable to C, C++, Pascal and so on without any trouble; and to link with the *glue* that is necessary for the part handler's code to link dynamically with the OpenDoc runtime code. Right now, during the alpha test period, one must use Apple's MPW development environment and program in C or C++. But by the time you read this it should also be possible to use the MetroWerks CodeWarrior environment and Symantec's Think environments.

On top of that, you have a choice of object-oriented programming framework for actually implementing your part handlers. Apple and Symantec have created the Open-

Doc Parts Framework (OPF): a set of C++ classes that can be used to construct a part handler quickly, dealing with the aspects of the part handler that don't reside inside OpenDoc itself. The OPF is based on the remains of the Bedrock cross platform applications development framework project, which failed famously last year, but that should in no way diminish the usability of OPF itself. OPF is a kind of grown-up MacApp, designed from the ground up to be cross-platform. It uses modern C++ idioms like templates and incorporates excellent collection and utility classes licensed from Booch.

One disadvantage of OPF is that, at present, you have to use the Symantec C++ compiler to compile your applications (it is the only one on the Mac that supports all the constructs like C++ templates that OPF requires).

Also on the table will be a version of Borland's Object Windows Library, which may appeal more to Windows developers but is also due to become cross platform.

Better, by design

So, to return to my initial point: why do I think OpenDoc represents a better applications development framework, that should be interesting to developers whether or not they want to build compound documents? First, the runtime environment saves the applications developer from a massive amount of grunt work: the event handling, storage, event message dispatching and resource arbitration are all built in. As the Macintosh, Windows and other operating environments have become more complex, the barrier to development of new applications has risen steadily. I used to know almost everything there was to know about the Mac toolbox ROM APIs; now, I seem to spend all my time looking things up!

> **Authors of part handlers will need much less platform-specific knowledge than they do today**

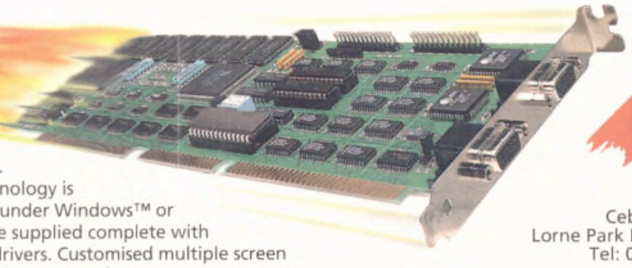
Authors of part handlers will need much less platform-specific knowledge than they do today. For instance, Mac part handler developers need only know about imaging for the purposes of drawing and printing. Menu and dialog management for human interface requirements, simple window creation (if the part is the root part of a document), plus the OpenDoc APIs for storage and message handling are also necessary. Everything else, however, is handled by OpenDoc.

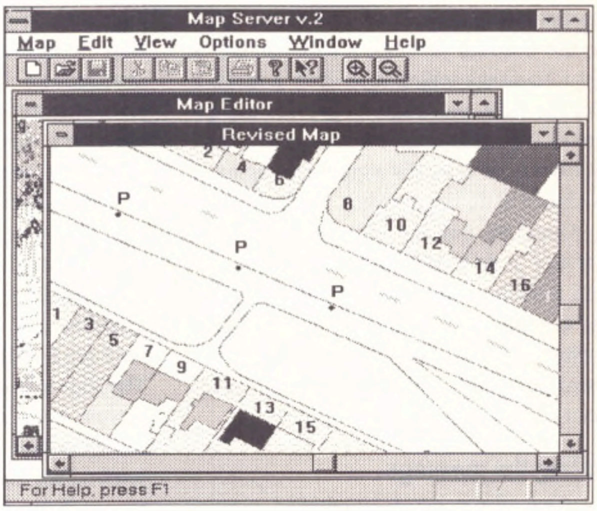
If you are building a new application which ships before OpenDoc does you may find it helpful to think about the problem in OpenDoc terms. This will make your job easier should you wish to make it into an OpenDoc part handler. In six months to a year's time, you can consider OpenDoc as a platform for all your new applications.

I'll return to OpenDoc in a future column, taking a look deeper under the hood, when the OpenDoc betas are released. ■

Paul G Smith enjoys the process of writing a little more than he enjoys low level debugging but like many other writers he loves the feeling of having finished writing an article. He also writes for Apple's house technical quarterly, Develop, and between these commitments still finds time for consulting and developing Macintosh and Windows software. He can be reached by telephone on 0727 844232, or preferably via the Internet at `paul@ctalk.exnet.com`

DUAL SCREEN
TECHNOLOGY
VIRTUAL Screen
TECHNOLOGY.




Map Server v.2
Map Edit View Options Window Help
Map Editor
Revised Map
2
4
6
P
8
10
12
14
16
P
P
1
3
5
7
9
11
13
15
For Help, press F1

Mayhem!

Having been to the movies, **Jules** wonders why his computer can't be as glamourous as those that grace the sidewalks of Sunset Boulevard

Films are a very strange art form. They look as if they're depicting life as it is, but in fact, mostly theydepict life as it should be. The goodies always win, the baddies always get their comeuppance, and nobody ever gets a visit from the Vatman. More interesting, though, are the codes used in films to tell you more about the stories. There was a time when all the goodies wore white hats. Now, audiences are more sophisticated. Heroes often have feet of clay, so white hats are too trivial a symbol. Before Philip Marlowe, for example, the baddies drank straight bourbon. Now, the baddies smoke.

It's not true to say that I can't enjoy a film for what it is, but I do enjoy looking for the little codes, the PCism's and clues as to the morality of the people who make the film. The codes are everywhere. They all add to the story by telling you what life should be like.

One of the codes that keeps appearing in films is the computer. In the forties and fifties, computers were huge edifices, electronic brains, and they were glamorous. In the sixties and seventies, they began to turn into real tools, such as the computer in Star Trek, but mostly they had a sense of menace associated with them, such as the machine in the Forbin Project or HAL. Even the Star Trek computer was given a nasty mechanical voice. This was perfectly reasonable; after all, these computers were almost always plot devices in their own right; they reflected the opinions of the age.

But in the eighties, something started to happen. Computers were appearing on everybody's desks. We no longer felt threatened by them. In the new Star Trek, for example, Data represents an automaton searching for an identity. I think that represents the spirit of our age very nicely. But Star Trek is an exception. Films these days also portray computers, but not in a realistic sense. Characters have computers on their desks only when the computer will become a plot device, otherwise they are entirely absent. And furthermore, these computers are called on to do things that no real computer could.

Take, as an example, *Jumpin' Jack Flash*, a comedy thriller from the early eighties. Whoopie Goldberg plays a bored computer operator, whose terminal is jammed by a secret agent. After a few tries, she guesses the secret password. Immediately her character's screen erupts into a riot of colour and animation which even modern graphics computers would be hard-pressed to copy. I accept that a simple welcome message and a plea to delete some files from the server would not have had the impact that the story needed, but the implausibility of the screen shot must have been obvious to everyone.

Consider, again, *The Fly*. This is an interesting example, because the original story (which was quite horrific) was written in the twenties, when the public as a whole were very optimistic about the future of science. The 1939 version of the movie was as close as Hollywood ever got to fidelity to the original, but the eighties version was quite different. Certainly, it was brought up to date, with timely throw-away lines like 'I've built a gene splicer - actually a very good gene splicer' - lines which would have made little sense even fifteen years ago. But look at his computer. It could zoom in on a bit of DNA, extrapolate that into a fly: All this in near real-time. But, even more significant, it was able to synthesise and use emergent concepts; commands like 'Name union of Brundle and fly Brundlefly' could not really be understood by any machine available today, and the volunteered 'Fusion of Brundlefly and telepod complete', while loaded with significance, was entirely implausible given the accidental nature of the.operation (which would have caused any real machine to core-dump).

Unlike practically every other commodity depicted on films, such as clothing, cars, or soft drinks (which are placed by manufacturers as part of the sponsorship of the film), computers always have a plot role. In *Sliver*, Sharon Stone is chatted up by the anti-hero who jams her Mac at work and puts a graphic of a rose on the screen. In *Missing*, the hero is saved because his girlfriend guesses the password to a part of a Word file. The list goes on, and of course we haven't even considered films where the machine is the central plot element, such as *Lawnmower Man* or *Short Circuit*.

I don't begrudge Hollywood using faked behaviour to make better films; as I said, films are only supposed to *look* like real life. But computers are everywhere. They are on desks in every office, every secretary uses them, and most managers too. People play games on them, write letters on them, bal-

VISUAL BASIC® TOOLS

THE DEVELOPERS SOFTWARE

Save time and money with the best range of powerful and easy-to-use Visual Basic® development tools from Contemporary Software. Featuring leading products from MicroHelp, Sheridan, Farpoint and Crystal Services.

MICROHELP

- VBTOOLS 4
  The best VBX control pack anywhere: Now over 50 controls **£115**
- HIGHEDIT PRO 2 & HIGH EDIT LITE
  WYSIWYG word processing & text editing for Visual Basic applications **From £199**
- MUSCLE 2
  600+ ASM routines for power programmers **£165**
- COMMS LIBRARY
  Reliable, easy-to-use communications routines **£155**
- NETWORK LIBRARY
  Easy-to-use node and admin routines **£80**
- LIST, LABELS & BARCODES
  WYSIWYG reports, labels, barcodes & more **£175**
- VBXREF 2
  Project management for Visual Basic applications **£80**

SHERIDAN

VBASSIST 3
Adds powerful design tools and new data access extensions for Rapid Application Development **£145**

DATA WIDGETS
Bound data controls for designing database application front ends **£99**

DESIGNER WIDGETS
Create apps that resemble current popular commercial packages **£99**

3-D WIDGETS
Give a 3-D State-of-the-Art look to your Windows applications **£89**

CRYSTAL

CRYSTAL REPORTS 3
The standard edition lets you create reports using data from the major PC databases easily and simply. **£149**

CRYSTAL REPORTS PRO 3
Pro 3 gives even more reporting functionality, a full copy of TrueGrid 2 + access to SQL databases. **£299**

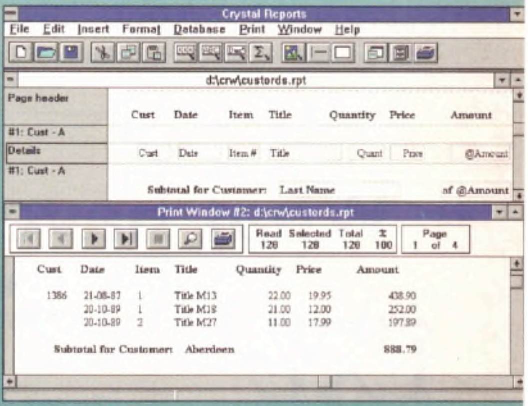
FARPOINT

SPREAD/VBX 2.1 & SPREAD/VBX ++2.0
The best spreadsheet control for Visual Basic & C++ **£175**

TAB/VBX
Gives your applications an organised, professional feel **£49**

GRID/VBX
The ultimate grid control on the market for VB & C++ developers **£75**

AWARE/VBX
The specialist tool for building database front ends **£110**


Crystal Reports
File Edit Insert Format Database Print Window Help
d:\crw\custords.rpt
Print Window #2: d:\crw\custords.rpt

```
Page header      Cust   Date   Item   Title        Quantity  Price    Amount
#1: Cust - A
Details          Cust   Date   Item # Title        Quant     Price    @Amount
#1: Cust - A
                 Subtotal for Customer:   Last Name                 of @Amount
```

```
Read  Selected  Total   %     Page
128   128       128     100   1 of 4

Cust   Date       Item   Title       Quantity   Price    Amount

1386   21-08-87   1      Title M13   22.00      19.95    438.90
       20-10-89   1      Title M18   21.00      12.00    252.00
       20-10-89   2      Title M27   11.00      17.99    197.89

Subtotal for Customer:   Aberdeen                        888.79
```

PRODUCT FEATURE

FARPOINT SPREAD/VBX 2.1 & SPREAD/VBX++ 2.0

£175

The two versions of Spread/VBX have over 250 properties that make them the most powerful and comprehensive data aware controls for Visual Basic, Visual C++, Borland C++ and Symantec C++. They are designed to implement spreadsheets, grids, listboxes, toolbars and more.

Look out for:

- **Microsoft Access Engine support** for VB users
- **Spreadsheet Interface Designer** reduces the time it takes to generate and prototype code
- **Database connectivity to Q+E Database Library** is a read/write control that gives virtual data management for oversized files and automatic updating of other data controls
- **Single or multiple cell operation** for back and foreground colours, formulae, borders, etc
- **Data Events** include Advance, ButtonClick, DblClick, EditError, LeaveCell, ColWidthChange, DragDropBlock, UserFormulaEntered, and others
- **Database Events:** EnterRow, LeaveRow, QueryData & TopLeftChange
- **GUI operations** include text overflow, read only mode, sizing of rows and columns, scroll freezing, hide/show single or all rows/columns, grid line adjustment, auto-sizing to fit parent spreadsheet, etc, etc.

Visual Basic is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective holders.

➤ CIRCLE NO. 580

CONTEMPORARY Software

FOR MORE INFORMATION AND TO ORDER

CALL NOW ON
0727 811999

Abbey View
Everard Close
St Albans AL1 2PS
Tel: 0727 811177 Fax 0727 848991

Prices quoted exclude VAT and delivery and may be subject to change without further notice

A purveyor of fine programs

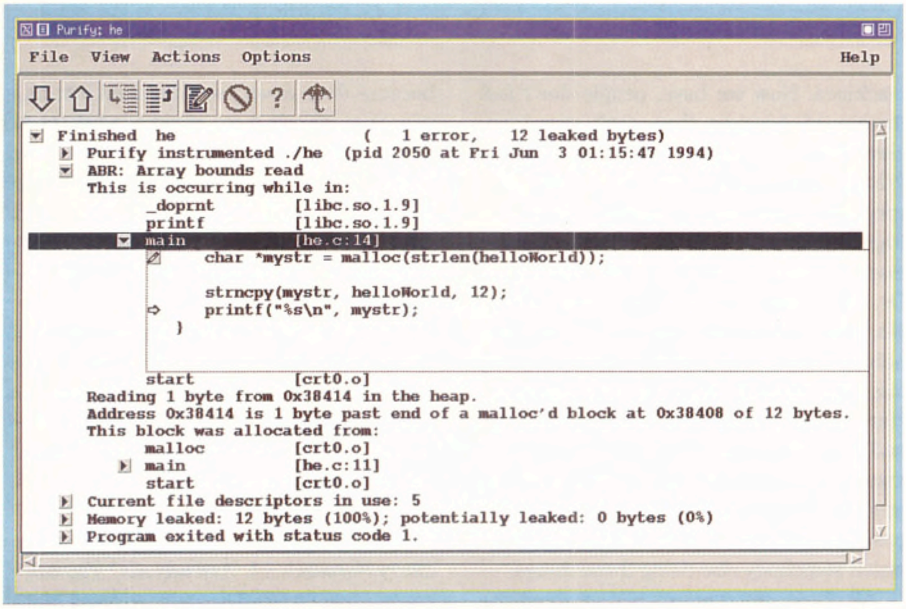
And there shall be no memory leaks. And null pointers shall be assigned. And data shall not be overwritten by errant programs. And **Peter Collinson** will be purified...

Recently I was at a meeting of programmers at a company I consult for. The company is concerned with the quality of its software. Much of the meeting was taken up discussing its experiences with code inspection. Towards the end of the day, someone started to talk about Purify. It sounded like just the thing for an *EXE* article.

I contacted the UK distributor who was very helpful. Not only was I supplied with a current release (Purify 2.1.1), I was also put in touch with the US authors who gave me a beta release of the next version (Purify 3.0-Beta-1, see Figure 1). A quick tour of the new features is given below.

Worth getting excited about?

Well, I write C programs and know that 'C is an unsafe language.' It's unsafe because it allows you to have untrammelled access to the address space in the virtual machine. C allows you to use pointers to memory. Pointers are often miscomputed to point at somewhere odd or loaded from uninitialised data. Pointers often suffer from 'one-off' errors, where they point outside the area of memory that they are intended to operate in. The program appears to work until one day the pointer access picks up some data that causes a crash.



```
Finished  he                  (   1 error,    12 leaked bytes)
  Purify instrumented ./he  (pid 2050 at Fri Jun  3 01:15:47 1994)
  ABR: Array bounds read
  This is occurring while in:
        _doprnt         [libc.so.1.9]
        printf          [libc.so.1.9]
        main            [he.c:14]
              char *mystr = malloc(strlen(helloWorld));

              strncpy(mystr, helloWorld, 12);
              printf("%s\n", mystr);
          }

        start           [crt0.o]
  Reading 1 byte from 0x38414 in the heap.
  Address 0x38414 is 1 byte past end of a malloc'd block at 0x38408 of 12 bytes.
  This block was allocated from:
        malloc          [crt0.o]
        main            [he.c:11]
        start           [crt0.o]
  Current file descriptors in use: 5
  Memory leaked: 12 bytes (100%); potentially leaked: 0 bytes (0%)
  Program exited with status code 1.
```

Figure 1 - Purify 3 runs under Motif

The other problem area is memory management. When I started programming in C, you allocated structures and buffers in the program using an array that was compiled into the data area of the program. These days, you tend to use `malloc` or one of its siblings to grab some memory, dynamically creating data structures or buffers as needed. Hopefully, you remember to free these blocks of memory later. If you don't, then you have a memory leak which may be okay, but often isn't. It is completely unacceptable that a long running program keeps getting bigger and bigger and bigger...

These kinds of errors are very hard to find. You can spend days tracking them down. You usually do this after a crash. By then it's too late. The problem actually happened sometime before the crash: the evidence of what's wrong is long gone. There is no good way to predict these errors. There is no good way to monitor memory accesses and verify that they are legal. There *was* no good way, that is, until Purify.

When you Purify your program, you generate a version where all memory accesses are checked to see if they make sense. Are you reading from some memory that hasn't been written to before? Are you trampling about outside the space allocated to arrays? Are you reading or writing to some memory that has already been freed? Are you calling `free` with odd values? Worse, are you freeing freed memory? I had a kernel bug that did this once. When the program terminates, Purify will tell you whether your program has leaked memory and which line in the program created the leaked block. In short, Purify diagnoses memory errors in your program. Better, it spots them before you ship it to customers. You debug the memory behaviour of your program when you write it and not when it has crashed, destroying the evidence of misbehaviour.

Making pure...

Purify needs to catch every program instruction that modifies memory. You could

Hardlock E-Y-E® – tying the hands of software pirates

The effective way to protect your software

FAST Electronic has made life a lot harder for software pirates. Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's number one in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool.

The technology programmers have at their fingertips

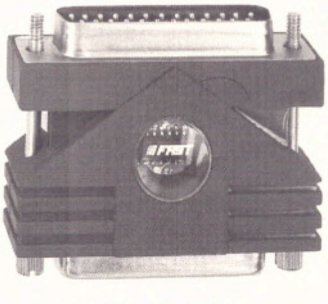
Hardlock E-Y-E is based on a custom chip and combines all the features that a programmer would expect from such a device: secure, algorithmic query routines and an optional non-volatile memory for custom configurations. With the Crypto-Programmer card from FAST you can program the algorithmic parameters and the memory within seconds. This unique card guarantees that no one else can burn Hardlock E-Y-Es with your codes. Linking Hardlock E-Y-E to your software is easy: you can either protect your .COM and .EXE files with the automatic encryption software HL-Crypt or integrate FAST's high level language routines into your source code.

Readily acceptable to your customers

Hardlock E-Y-E allows unlimited backup copies of the master floppy. The customer gets the device together with the software and plugs it into the parallel port between the printer and the PC. Daisy-chainability, outstanding reliability and the compact high tech design guarantee that your customers will accept Hardlock E-Y-E.

The benefits your management will appreciate

Hardlock E-Y-E can be programmed by the software house with the Crypto-Programmer card. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented.



Hardlock E-Y-E
programmable, algorithmic response and memory option – all in one

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio # 6, W 11 4 EY,
Telephone 071 221 8024, Fax 071 792 3449.

Magnifeye is a subsiduary of Fast Electronic GmbH.

➤ CIRCLE NO. 581

```
% cat he.c
#include <stdio.h>
#include <malloc.h>

static char *helloWorld = "Hello, World";

main()
{
   char *mystr = malloc(strlen(helloWorld));

   strncpy(mystr, helloWorld, 12);
   printf("%s\n", mystr);
}

% ./he
Purify'd ./he  (pid 22121 at Fri May 27 10:02:29
1994)
Purify 2.1.1 SunOS 4.1, Copyright 1992, 1993
Pure Software Inc.
For contact information type: "purify -help"
Purify licensed to Purify Evaluation User

****  Purify'd ./he  (pid 22121)  ****
Purify (abr): array bounds read:
  * This is occurring while in:
        _doprnt        [doprnt.o, pc=0xf7537d04]
        printf         [printf.o, pc=0xf75461fc]
        main           [line 14, he.c, pc=0x1b974]
        start          [crt0.o, pc=0x2064]
  * Reading 1 byte from 0x46154 in the heap.
  * This is 1 byte past end of a malloc'd block
at 0x46148 of 12 bytes.
  * This block was allocated from:
        malloc         [p6.o, pc=0x485c]
        main           [line 11, he.c, pc=0x1b92c]
        start          [crt0.o, pc=0x2064]
Hello, World

****  Purify'd ./he  (pid 22121)  ****
```

```
Purify: Searching for all memory leaks...

There are 12 leaked bytes (100% of the 12 allo-
cated bytes in the heap)

Purify (mlk): 12 bytes at 0x46148 lost, allo-
cated from:
        malloc         [p6.o, pc=0x485c]
        main           [line 11, he.c, pc=0x1b92c]
        start          [crt0.o, pc=0x2064]

Purify Heap Analysis (combining suppressed and
unsuppressed chunks)

                         Chunks      Bytes
              Leaked          1         12
  Potentially Leaked          0          0
              In-Use          0          0
  ----------------------------------------
     Total Allocated          1         12

****  Purify'd ./he  (pid 22121)  ****
  * Program exited with status code 1.
  * 1 access error.
  * Basic memory usage:
      135584 code
      140024 data/bss
       16392 heap
         696 stack
  * Shared library memory usage:
      696320 libc_pure_211.so.1.9       (shared
code)
       16384 libc_pure_211.so.1.9       (private
data)
        8192 libinternal_stubs.so.1.0   (shared
code)
        8192 libinternal_stubs.so.1.0   (private
data)
```

Figure 2 - Sample output from a special buggy version of 'Hello World' program

contemplate writing a debugger that does this job. A Unix debugger generally uses the standard hooks that the kernel provides to trace another process. However, watching every instruction being executed is slow because of the context switch overhead involved in passing information between the debugger and the traced process. Try tracing a program under control of say **dbx**; you'll soon see quite how slow things are.

The authors of Purify tried a different tack. They embedded calls to their monitoring routines into the compiled program. Purify does this before the user's program is linked using `ld`. So it takes the `.o` file generated by a normal compilation run and adds calls to its own routines before and after any instruction that makes a memory access. There is no difference between the object files created by the user and the contents of the **ar** files that contain the system libraries. A program that can add instrumentation to the user's object files can do that to the system libraries. You don't do this in place, of course; you copy the libraries somewhere else and modify them.

If you can deal with the program that the user is compiling ***and*** the system libraries, then you can monitor all of the user's program in operation. This is good because sometimes the user will pass rubbish values into library calls. Now you can trap this. Sometimes, however, the author of the system library routine has been at fault and the user might wish to supply his own fault-free routines.

The final piece that allows you to track memory usage is a way of trapping calls to the **malloc** and **free** routines. Purify doesn't provide its own code for these. It merely captures calls and maintains a memory map with the result. It then goes on to call the **malloc/free** routines that exist in the program. This means that the Heisenberg effect of Purify is minimised. It runs your program as accurately as possible, giving it more chance to find those errors. The Purify manual gives you instructions on what you do if you don't use the standard memory allocation code. However, the majority of programs will use **malloc** and **free**, simply because they use **printf** and in turn that will use the **stdio** library. The **stdio** library uses the routine pair to allocate and release data structures and buffer space.

Heap patrol

So we have some calls around memory accesses and we have trapped calls to **mal-**

JAKE ABRAMS

ance chequebooks on them. Everybody, in their heart of hearts, knows what a computer can do and what it can't do. So why is it that audiences can accept this distortion of reality, where in any other context they would reject it out of hand? If you don't believe me, look at the lengths they went to in order to make plausible the central plot elements of things like *Knight Rider, Quantum Leap*, and other neo-fantasy series.

I have a theory. I think that the general public was deliberately prepared for a computer revolution that never seemed to happen. Oh, it happened all right, but nobody actually noticed. Computers were supposed to turn people out of work by the millions, but in the event it was the politicians who got the blame. We were supposed to hand over responsibilities for all sorts of things to machines. Now we have, people don't feel threatened because they can't even concieve of all the machines being unplugged. What's more, people now actually crave the very impersonalness of machines which so frightened them before; that's why ATM machines are so popular, and that's what Dr. Theophilus in *Buck Rogers* was saying. No, there was no revolution at all, because nobody was hanged, nobody started shooting, and nobody destroyed the factories. Here we are, at the end of the revolution, and these wonderful, terrifying, miraculous machines are merely boring. It's boring to buy one, it's boring to use one, it's boring to program one, and it's incredibly boring to watch somebody else doing these things.

All along, we were all talking about the revolution and we were all reading about it, because deep down, we all wanted it to happen. Sure, people were afraid of what the future might bring, but they were more afraid of the future passing them by. Now, the future, the brave new revolution, has been and gone, and nothing significant has changed. Deep down, we all still want it to happen.

Hollywood tells us how life should be. In Tinseltown, computers are still unpredictable, wild, a frontier. Hollywood is guarding the last guttering hope of what people want this technology to be. That, after all, is what it is there for. ■

Jules is a programmer who would rather like Barry Norman's job. And why not? You can contact him on 0707 644185, or email him as jules@cix.compulink.co.uk.

A quick tour of Purify 3

The new release has a very nice X windows interface that acts as your point of contact to the system. It also comes with a tool called Purecoverage, which tracks the path that is taken through the code of the program. Sadly, I'm not going to have space to talk about that.

The basic reporting and leak detection mechanism is not altered too much in the new release. There are some new categories of error that are checked, mostly concerned with stack activity. It will also check for file descriptor leaks in the program.

If you run X, you will see a very well designed display viewer that summarises the results and allows you to interact with the Purify system. You can see this in Figure 1, (my Sun only has a monochrome screen, so here I am displaying the image on the colour screen of my BSD/386 system). The main display area behaves like an outline editor. It shows titles for the data, but doesn't display the full details until you hit the sideways pointing button. The button changes to point down. I have done this on the ABR line and again on the line for `main`. This displays the source line where the problem was spotted. If you click on the small pencil, your editor is started in a separate window. The editor is taken from the `EDITOR` environment variable and the current line will be set to the line of the error (if your editor supports this).

Various different methods of working with the viewer are supported. You can use the left mouse button to expand and contract by pointing at the buttons, while the right mouse button generates a context sensitive action menu. You can use entries from the drop down `Action` menu on the top line or you can click on one of the set of icons in the 'toolbar'. You may also set suppressions (the 'stop' sign), get help (the question mark) and start Purecoverage to analyse the run of the program (the umbrella). You can choose to display or suppress the toolbar, if screen real-estate is important to you. The viewer makes extensive use of the X feature of 'greying out' actions that are not applicable in the current context. I am very impressed with the viewer. This whole notion that an outline editor is used to control the amount of information that is displayed to you is excellent.

The viewer can do more than just look at the program while it is running. If you select the program toolbar from the View menu, then you get a toolbox containing Make, Run, Debug and Kill buttons. You can use these (and the editor button) as a development management centre.

`loc` and `free`. What happens now? Purify maintains a two bit state variable for every byte in the program's address space. And yes, this can be a large table.

Memory in the 'heap', the arena that `malloc` uses, starts off as unallocated or the 'red' state as the Purify manual calls it. It's illegal to read or write in this area. Once the memory has been allocated by `malloc`, it moves to the second state: allocated but uninitialised. This is the 'yellow' state. You can write to this memory, but reading is flagged as an error. The memory can be returned to the 'red' state by passing it into `free`. The memory moves to the `green` state when it is written to. Here it can be read or written. It only returns to the `red` state by calling `free`.

Once you have the ability to tag memory in this way, you can use the scheme to set up guard blocks around, say, an area of memory obtained by `malloc`. These extra guard zones can be used to check writing outside arrays. You can also flag the stack as unwritable and catch a routine that is trampling on a previous stack frame.

To catch a leak

The final piece of the jigsaw is to provide some code that detects memory leaks. This is actually fairly hard to do accurately. The key idea is to look for chunks of memory allocated by `malloc` that do not have pointers to them. First, Purify follows all the pointers that it can find in the data and stack areas of the program and marks all the blocks that are in use. Then it searches the list of allocated blocks looking for memory that has not been touched by the previous sweep. If unconnected areas are found then these are leaked memory blocks.

> **It is completely unacceptable that a long running program keeps getting bigger and bigger and bigger...**

Sounds easy? Yes. But think a little about what the sweep phase means. What happens if the sweep phase finds an integral value *that looks like* a legal pointer? It knows nothing about the types of the value, it must believe that it's a valid pointer to memory. What happens if the sweep phase finds a pointer to the middle of an area that is on the leaked list with no pointers addressing the start of the block? Purify marks such blocks as 'potentially' leaked.

The final arbiter of whether the memory is leaked or not rests with the programmer. Purify helps here by tagging each block that `malloc` returns with the stack trace of the caller. So you can find out where in your code a block was allocated and so deduce what the block contained once and from there (perhaps) why it is possibly unallocated.

Purifying your own code

So now you know what it is and how it works. The next question is how you use it in practice. Using Purify couldn't be easier. Rather than saying:

```
% cc -g -o he he.o
```

you say:

```
% Purify cc -g -o he he.o
```

The following output results.

```
Purify 3.0-Beta-1 940412
SunOS 4.1, Copyright 1992-1994
Pure Software Inc.
Instrumenting: hello_world.o
Linking
```

Notice that it replaces the load phase of the compilation. This makes it easy to alter makefiles to use the package. Purify will generate a file called `he.pure.o` which is the altered version of the object file.

It also needs to doctor the libraries that the program uses. Purify'ed system libraries are stored in a cache directory, so the work is only done once. This makes compilation not at all painful after the first run. The Purify phase is not really noticeable, once the libraries are established in the cache. You now have a runnable object that is instrumented.

Levels of perfection

What happens when you run it depends on the version of the program. With version 2, you will see reports from the program coming out on the console (see Figure 2). The `he.c` program is actually an example program from Pure Software. You can see the code at the top of Figure 2. It's a special buggy version of our old friend, the 'hello world' program.

When you run the program, it first outputs some identifying messages telling you that the program is Purify'ed and by what version of the code. We then hit a bug. This is an 'ABR': the program is reading outside the bounds of an array. The program has failed to allocate space for the null byte at

How to save your share of over £3,100,000,000 a year.

Data Encryption Systems Limited

Silver Street House,
Silver Street, Taunton,
Somerset. TA1 3DL

Telephone 0823 352357
Fax 0823 352358

That's how much money was lost in europe through improper or illegal use of software last year. Your money.

If you are in the business of software development, then the emphasis is on you to protect your own investment by securing your software products.

Since 1985, Data Encryption Systems have been the UK's leading software protection specialists. They have created a range of state of the art security solutions for software developers and IT managers alike.

Whilst retaining an impenetrable level of security the DESkey is easy to implement, reliable, transparent in operation and very flexible.

Our unrivalled range covers all standard operating platforms including DOS, Windows, UNIX and environments from stand alone systems to LAN's and mainframes.

DESkeys are available in a variety of low-cost high-speed models from the standard DK12 to the immensely sophisticated and versatile DK2 and DK96.

And as with all Data Encryption Systems products, DESkeys are easy to use.

With DESlock and DESnet you can implement DESkeys in a matter of seconds – by just typing a simple DOS command or selecting a Windows icon.

What's more, Data Encryption Systems can give you total technical support including bespoke product development and, where appropriate, access to our highly experienced team of designers.

Now, thanks to Data Encryption Systems, there is a genuine alternative to lost software sales.

They're worth looking into now – to save your share of the £3,100,000,000!

If you would like to find out more about DESlock and DESkey, or would like a demonstration kit, please call us on 0823 352357 or fax us on 0823 352358.

Member of FAST
Promoting the legal use of software

➤ CIRCLE NO. 582

the end of the string and has called `printf` that expects to see the null byte. As you can see, the program still succeeds in printing its outputs, luckily the byte after the allocated space is null. This is a bug in a tiny program that we would have missed.

The program now finishes and Purify will start looking for memory leaks. It finds one - the string we have `malloced`. It thinks this is a leak because we have not called `exit` in the program. Had we done that, Purify would have reported the block as 'In-Use'. Finally, we are told about the memory usage of the entire program and libraries. This is the size *with* Purify. The program has 16 KB of program and data when compiled without Purify.

There are various things you can change about the output that you see. You can change the default behaviour when you run Purify to link the program by supplying one or more of zillions of options. For example, you can arrange to have output in a file or have it sent in the mail to the user.

More importantly, you are able to suppress output from the program by adding a line to a file called `.Purify` on the current directory. The program will also read suppression directives from a system-wide control file buried in its installation tree. This is important because you want to reduce the amount of output that some programs make.

Installation

Purify is a good package not only because it contains some novel software technology but also because the technology is easy to use and comes well packaged with a readable manual. It installs easily too. It's just a matter of creating a directory to load the `tar` image from the floppy, loading the `tar` image and running an install program that does all the business of asking for licensing details etc. It also unpacks some `gziped` files, then runs a test program. If that gives the message that all is well, it considers that Purify is working. The whole thing is quick, easy and painless. Getting the system off your disk is easy too; you just remove this directory from the disk. Good engineering at this stage tends to indicate right thinking later, I always feel.

Docs

The manual describes Purify in terms of a couple of example programs that are available in the distribution tree. The manual presents a tutorial approach that explains what is happening and what to do about it. It's a well designed book in a spiral binding so it will lay flat on your desk without the help of the nearest heavy object to keep it open. Almost my only quibble with the whole package is that the printing for the examples in this book is too faint, so I need to don my glasses. Old age is setting in and I am becoming short-sighted.

Licensing

You buy Purify based on the number of people that you want to use it. When someone has used the package more than twelve times, then they are deemed to be a 'regular' user. The seats that you buy are 'regular' users. Purify will send them mail saying 'Welcome to the club'. The system administrator can adjust this list of users. The license manager is advisory and probably easy to avoid, but that's not the point.

> **Purify diagnoses memory errors in your program. Better, it spots them before you ship it to customers.**

Debuggers and Purify

Both Purify versions work with `dbx` or `gdb` to allow you to control the program while looking for leaks and other memory problems. The viewer starts `dbx` up in a new window if you hit the Debug button. It also works with my favourite debugger, `ups`, although `ups` hides the actual addresses used in the program rather too well, making it a little less useful than `dbx` for some tests.

Purify supports debuggers by providing you with a place where you can set a breakpoint (`purify_stop_here`) so that the program will stop when Purify finds an error. You can inspect the stack or other variables in the running program to see what is wrong.

Purify also provides a number of routines that can be compiled into your program to provide dynamic checking at specific points. For example, you might like to run the memory leak code at some point in your program to prove that the leak happens in one path through the code and not another. You can do this by adding calls to the relevant Purify routines into your code.

These routines can also be called on the fly from `dbx`, should that seem sensible. Perhaps the most interesting of these routines are the set of routines that allow access to 'Watchpoints'. These allow you to track actions on a particular piece of memory. You can trace which bit of your code is responsible for changing a specific piece of memory. As the Purify manual points out, this is much better than using a debugger to watch a piece of memory because the debugger has to single step the program an instruction at a time and this can take forever.

The good news and the bad

Well, as you may have detected, I like Purify. I think that the whole package is well engineered and documented. Installation is trivial; you do need lots of space for those caches though. It's easy to use and the resulting program runs fast enough for you to feel that it's something worth doing. Caching of libraries speeds the linking phase immensely. Coupled with Purecoverage, the package is a really great development tool.

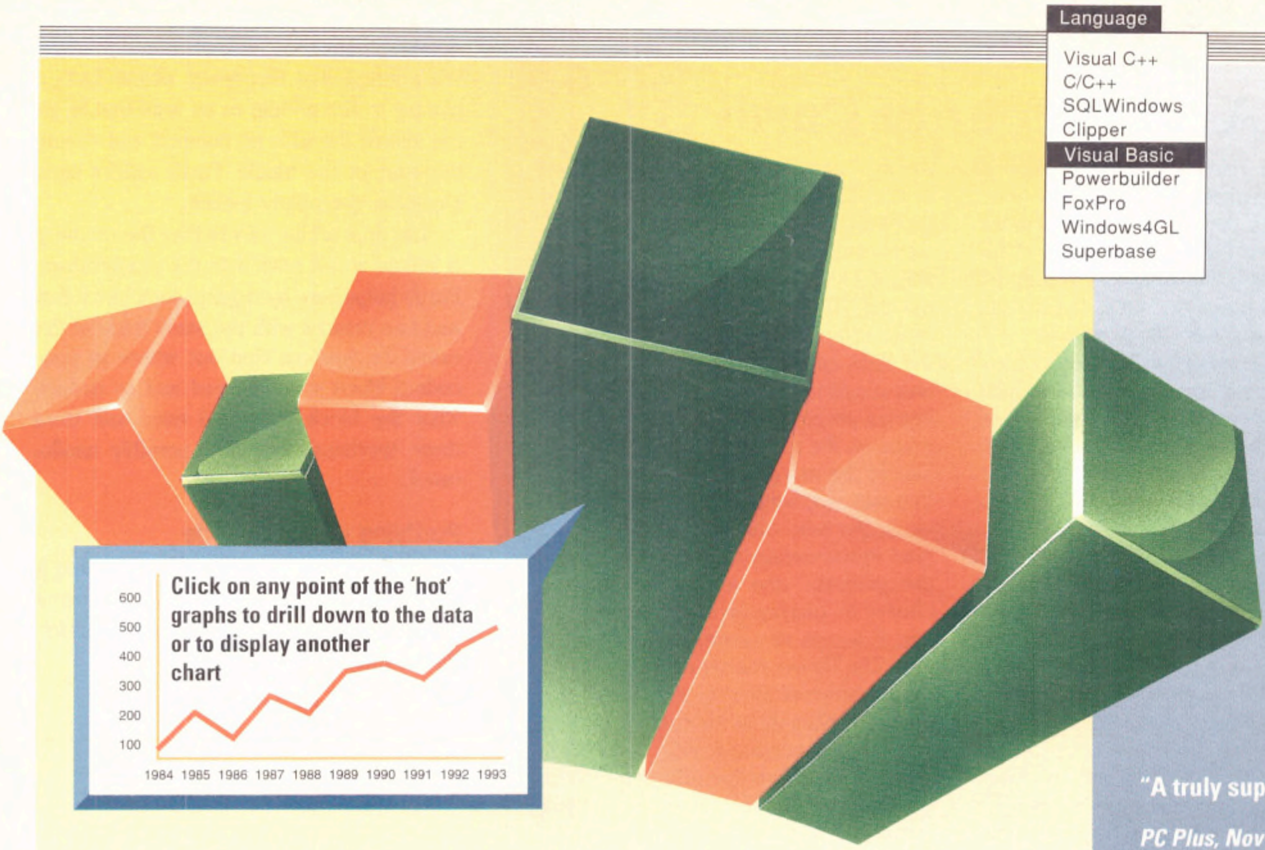
I have spent a happy hour or so tracking down bugs in various programs that I have lying about the system, some of my own and some written by others. Tracking down some errors can be non-trivial but the debugging hooks are a great help. The spectacular thing is that it really does find bugs that you just don't see in supposedly working programs. I am appalled at how bad the X libraries are. They rely very heavily on the fact that Unix will guarantee zero filled blocks when memory is expanded.

A friend at a local University tells me that he wouldn't be without Purify as a tool. He is a C++ programmer and that language is even more prone to memory leaks than C. He does also tell me that the licence manager is not that good in a distributed environment. He has difficulties with the number of 'frequent' users, which of course, changes constantly in a University. I imagine this is less of a problem in a commercial environment where the number of users change less frequently.

There is some more bad news for those of you without Suns or HP machines. It's limited to Sun Sparcs running Solaris or SunOS, HP9000 Series 700 workstations running HP/UX 8.07 and above, and HP9000 Series 800 workstations running HP/UX 9.0 and above. It does seem to work with most of the C and C++ compilers that I have heard of for the Sun systems. Unfortunately, porting to different architectures is a non-trivial exercise as Purify needs intimate knowledge of the instruction set to work. Let's hope that the authors add support for more platforms sooner rather than later. ■

Peter Collinson is a freelance consultant specialising in Unix. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Purify is available, at £1,330 per seat from from Contact Productivity through Software. Phone: 0928 579700, Fax: 0928 579701, email: info@pts.co.uk.



"A truly superb tool"

PC Plus, Nov '93

"You won't find an easier way to add charting to your Windows applications"

PC Magazine, Sept '93

Programmable graphs

DOS and Windows

Whatever language you have chosen for DOS or Windows development, you can use one of our libraries to integrate graphs and charts into your application.

Best graphics tool

dGE has been voted Best Graphics Tool for six years in a row, and in 1992 Graphics server was voted Best Add-on DLL. Microsoft licensed our technology for the graphs in Visual Basic Pro (we've added 'hot' graphs to our VBX since then).

Re-usable components

The future of applications development lies in 'components' which programmers can use time and again. Graphs and charts are part of so many of today's applications that it makes sense to use the same graphing component, especially one which 'gets full marks for coming with so many interfaces' *(Windows Tech Journal, April '93).*

Windows NT and OLE 2

If your long-term plans include Windows NT, you can be confident that Graphics Server will still deliver the charts you want. In the latest release we have added a 32-bit version for Windows NT. And we have developed an OLE 2.0 automation server - the beginning of the end for function calls!

Leaders in our field

We founded Bits Per Second, here in Brighton, eleven years ago, when getting graphs into a dBASE app was the leading edge of PC development.

We've stayed out in front and now count Microsoft and Gupta among those who license our software.

Thousands of developers in business, engineering and science now use our programmable graphs (and distribute the run-times royalty-free). If you'd like to join them, do give us a call.

Graeme Kitt John Marsh

BITS PER SECOND

14 Regent Hill
Brighton
England
BN1 3ED

Telephone
0273 727119

Fax
0273 731925

► CIRCLE NO. 587

A recipe for good X

Niall Mansfield begins a series on Motif programming with a line-by-line guide to 'Hello World'.

Motif programming is easy. And for anyone who has to write GUI applications for Unix it's essential, because it's the only realistic option available. Even as recently as six months back the issue wasn't so clear cut; Sun Microsystems was pushing OpenLook as *the* standard. But OpenLook never caught on and Sun has at last jumped on the bandwagons and adopted Motif. On top of this, Microsoft's Windows NT has scared all the big Unix vendors: to fight off this threat they combined in yet another consortium to produce the Common Open Systems Environment (COSE, pronounced 'cosy'). A very important part of COSE is the Common Desktop Environment (CDE), which uses the Motif look and feel. So in summary, Motif is the way to go.

In this and the next few articles I'll take you through Motif programming step by step. You don't need to have done any X programming before, but I assume you have some idea of what X is about. At the end you'll be writing useful programs (hopefully). But just as important, you'll have a grounding in the basic techniques of Motif programming, so you'll be able to find your way around the lots of good books about Motif and explore more advanced topics on your own.

Before we begin I'd like to say something on terminology. First, the difference between X Windows System and Motif. X is the basic window system, but you can think of it as the low-level 'window operating system'. Motif is like the standard libraries layered on top of that, to make programming easier, and to hide lots of the nasty bits. Its full name is OSF/Motif (Open Software Foundation).

An X application is called a 'client' and the display is called the 'X server'. These don't have to be running on the same machine: X can work over a TCP/IP LAN or wide area network.

The ingredients

We assume you are using a Unix system. You'll need a C compiler, some implementation of the X Window System, the standard X libraries and a Motif development system with its header files and libraries. Some workstation systems come with all these bundled, but you'll probably have to buy a Motif development system if you don't already have one which shouldn't set you back more than £100 per CPU. You will also need documentation. The minimum is the *X Toolkit manual*, and the *Motif Programmer's Reference*. You'll use these *a lot*, so it's worth having your own copy, at least at the beginning.

Familiar ground

We are in Unix land, so the first program we have to write is the one which prints 'Hello, World'. The makefile in Figure 1 is used to compile the program. The code is listed in Figure 2. Don't worry too much about some of the code details yet. We'll get programs working first, then go back and explain what the various components are, how they fit together, and the options that apply. You can experiment with them on your own, which is the best way to learn. Then you can run the program by typing the command 'hellow' at your shell prompt.

The program draws one **big** pushbutton (see Figure 3). Click on it with the mouse and you'll see that it 'goes in' and 'comes

A very important part of COSE is the Common Desktop Environment (CDE), which uses the Motif look and feel

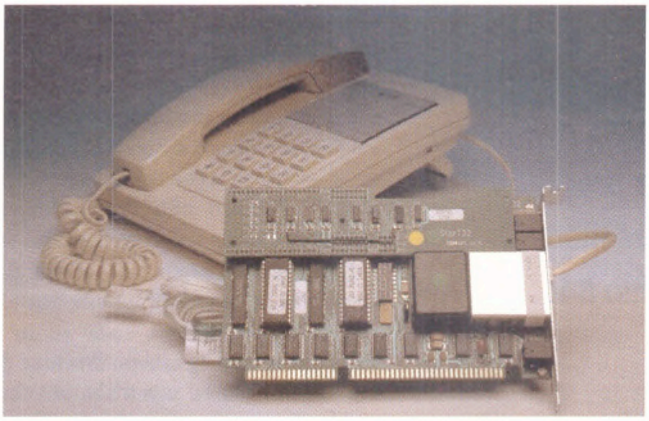
```
hellow: hellow.o
	${CC} -o hellow hellow.o -lXm -lXt -lXmu -lXext -lX11

hellow.o: hellow.c
	${CC} -D_NO_PROTO -I/usr/local/motif/usr/include
	-I/usr/X11R5/include -c  hellow.c
```

Figure 1 - Makefile for 'Hello World' program

VOICE PROCESSING

INTERACTIVE VOICE RESPONSE

COMPUTER TELEPHONE INTEGRATION



ST32

THE DEVELOPER'S CHOICE FOR SINGLE LINE TELEPHONY APPLICATIONS ON THE PC *

A half length card for ISA Bus IBM compatible PCs

- WINDOWS DRIVER FOR WINDOWS V3.1
- TSR TOOL KIT FOR MS-DOS AND OS2 V2.x
- WINDOWS BASED SOUND EDITOR
- SCRIPT LANGUAGE FOR EASY PROTOTYPING
- MICROPHONE AND LOUDSPEAKER
- DTMF RECOGNITION AND DIAL OUT
- PULSE DIAL OUT
- 16 BIT DIGITAL SOUND QUALITY
- SOCKET FOR LOCAL PHONE
- RECORDING FROM LINE, MICROPHONE OR LOCAL PHONE
- PLAYBACK TO LINE, LOUDSPEAKER OR LOCAL PHONE
- TIME BREAK AND EARTH RECALL
- OFF HOOK SENSORS FOR BOTH CARD AND LOCAL PHONE
- JACK FOR ALARM EVENTS
- SOFTWARE READABLE SERIAL NUMBER

NEW!

Speech Recognition
Speaker Independent
Connected Speech

APPROVED
for connection to telecommunications systems specified in the instructions for use subject to the conditions set out in them
S/2562/3/P/502756

Staria has been developing single line telephony calls since 1988. Our customers have produced applications in voice messaging, message broadcasting, telemarketing, plant monitoring, computer monitoring, telephone servicing, network testing, caller conferencing etc. Now, with added Windows and Speech Recognition capability, the ST32 is available for personal desktop use.

COMING SOON:

- FAX AND DATA MODEM OPTION
- SUPPORT FOR .WAV FILES
- DIAL OUT FROM OTHER WINDOWS APPLICATIONS

PRICES:

- DEVELOPER'S KIT: **£600** (INCLUDES ONE CARD) - SPEECH RECOGNITION **+£200**
- ADDITIONAL SINGLE CARDS: **£300**
- QUANTITY DISCOUNTS

* ADDITIONAL CARDS CAN BE ADDED FOR UP TO 32 LINES

IVR for the PC
Staria

Staria Limited (Ask for Brian Atherton)
TEL: 0989 768 687 FAX: 0989 768 980
13 Market Place, Ross-on-Wye, Herefordshire HR9 5NU

➤ CIRCLE NO. 584

```
*labelString: Hello, World
*fontList: *Times*-r-*-240-*
```

Figure 6 - The Hworld resource file

pen next, but it must be able to handle whatever the user does - whatever events occur. Figure 7 gives a schematic representation of an event-driven program looks like. This structure is mirrored in the code of our program. First we initialise the toolkit and create and initialise our button (ie the calls to XtAppInitialize(), XmCreatePushButton(), XtManaheChild() and XtRealizeWidget()). Then we go into the event loop (and stay there) through the call to XtAppMainLoop().

In fact all Motif programs will have this structure; what we'll see below is how to set up the program so that your own code can be triggered whenever certain events occur in the user interface. This will actually be the heart of your application - the code which performs the function your application was written for in the first place, say printing a file, or running a calculation or displaying a chart. So much for the overall structure.

Now for some line-by-line detail. First we include the Motif header file that relates to the pushbutton which we are going to use. Each Motif object (or widget, as we say in the trade) has its own header file. Its name is described in the Motif documentation. These widget-specific headers also automatically include other, more general, X and Motif headers which are necessary.

The local variables declared at the start of main() define some Toolkit objects we are going to use. pbutton is fairly obvious - it's the button that we have seen in use. The mytop widget is what is called the 'application shell widget'. It is the application's main window; everything else in the application is contained within it. It is created by the XtAppInitialize() call as we initialise the toolkit.

The final object we define is myappc which is an 'application context'. This holds, amongst other things, information about the display the application is using. It is really only used in anger with single-process multithreaded applications, so don't give it much thought.

XtAppInitialize() connects this client application to the X server and initialises the underlying X and Toolkit subsystems. It processes the command line flags which we've already used above. This is the reason why it must be passed the argc and argv arguments of the main() program. It can also take lists of other strings specifying default resources for the program and definitions of any command-line flags specific to this program and how they should be handled. However, that's all very detailed so we omitted using any of this by setting all the lists to NULL. Finally, as we already mentioned, XtAppInitialize() creates the application's shell widget, mytop.

> **Different types of widgets have different callbacks, depending on what type of object the widget is**

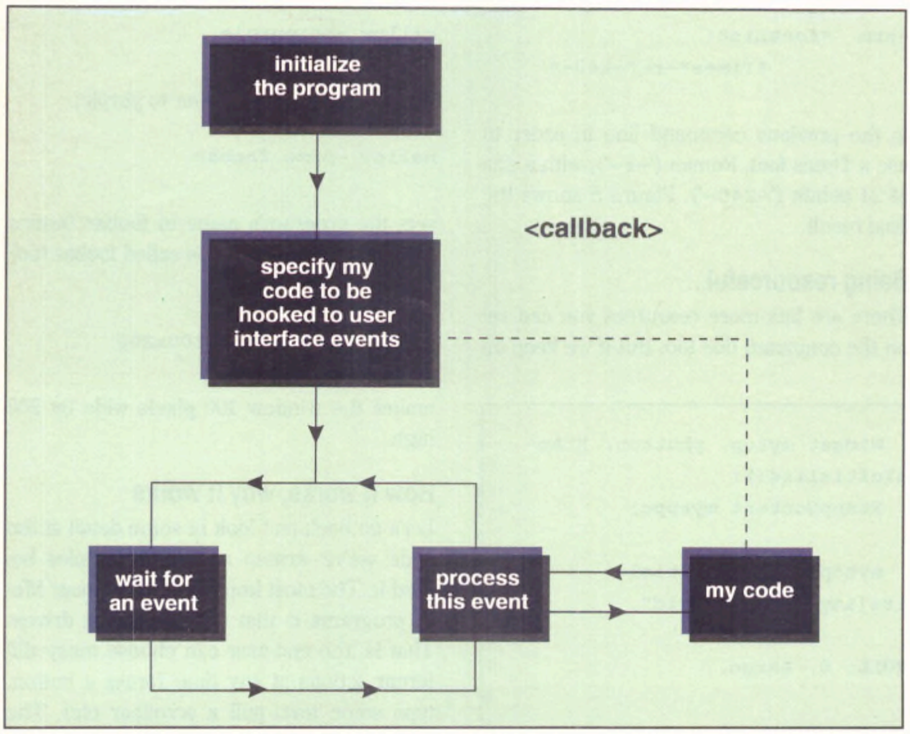


Figure 8 - Doing something useful through callbacks

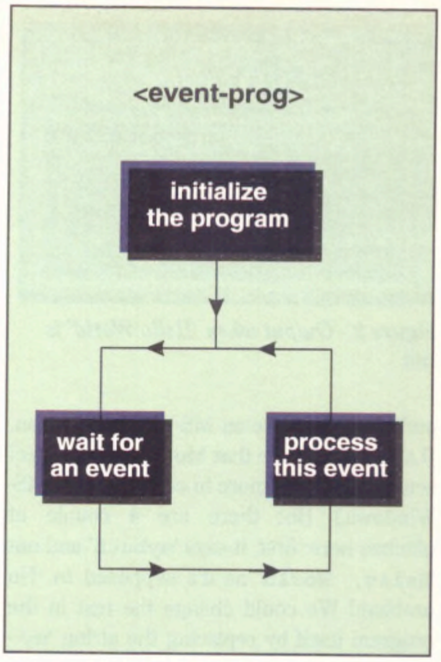


Figure 7 - Structure of an event driven program

In a Motif program, widgets are organised in a tree structure. At the root of the tree is the application shell widget. This can have exactly one child widget. The Motif function XmCreatePushButton() creates a pushbutton widget (surprise, surprise!) as a child of the widget specified in the first argument (mytop here). The address of the newly created pushbutton is returned and assigned to the Widget pointer pbutton. The *name* of the pushbutton, for use in resource files and elsewhere, is specified as a string in the second argument, 'mybutt' in this instance. The final arguments let you set some resources at the widget's creation time but we don't need to do so in this simple program. XtManageChildWindow() says that the parent of the widget pbutton should now take note that it has a child and be prepared to handle it (positioning it, sizing it etc).

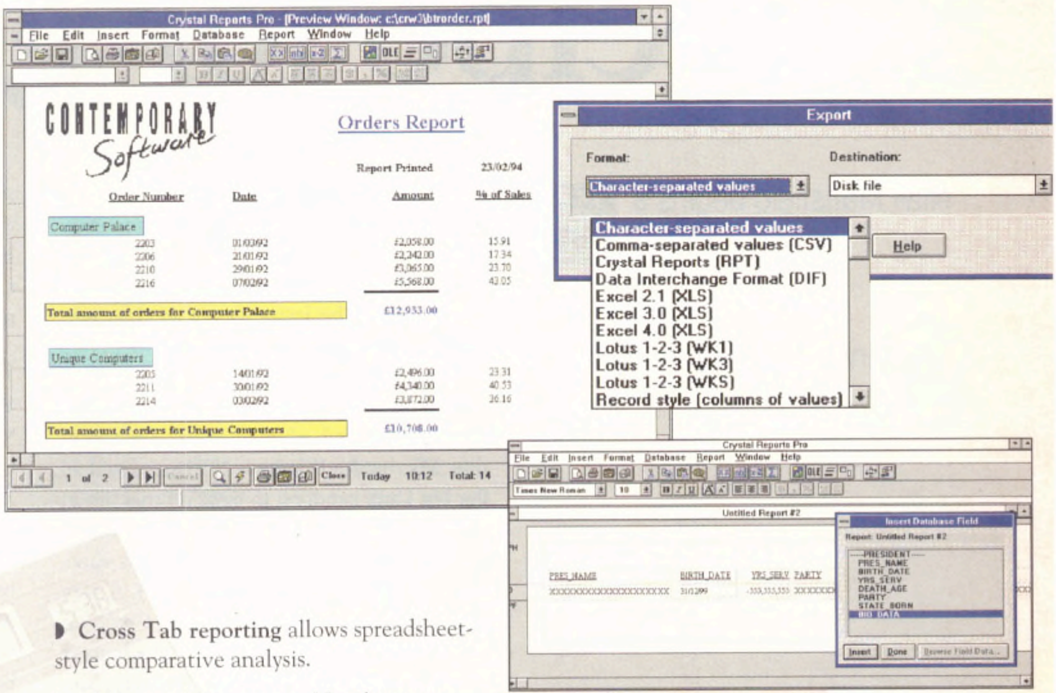
The function XtRealizeWidget() tells the toolkit that all the widgets we want, at least for the time being, have been created, and that all the X windows associated with these windows should actually be created on screen. The reason for this threefold split - widget creation, widget management and window creation - is just efficiency. It means that the calculations about size and position can be done once, when everything is ready, rather than cre-

CRYSTAL REPORTS 3

Clearly the first choice for database reporting

Award-winning Crystal Reports has made life easier for thousands of developers, and users. Now Crystal Reports 3 takes a dramatic step forward with a host of extra features which make it the most advanced database report writer ever.

New Crystal Reports 3 makes it **faster** and **easier** than ever to design, print and distribute reports. You can easily transform complex data, diagrams and tables into reports that are, literally, crystal clear and presentation-ready. It's *so* good that Microsoft includes Crystal Reports 2 as a key component in Visual Basic Pro 3.



Orders Report

Report Printed 23/02/94

Order Number	Date	Amount	% of Sales
Computer Palace			
2203	01/03/92	£2,058.00	15.91
2206	21/01/92	£2,242.00	17.34
2210	29/01/92	£3,065.00	23.70
2216	07/02/92	£5,568.00	43.05
Total amount of orders for Computer Palace		£12,933.00	
Unique Computers			
2205	14/01/92	£2,496.00	23.31
2211	30/01/92	£4,340.00	40.53
2214	03/02/92	£3,872.00	36.16
Total amount of orders for Unique Computers		£10,708.00	

Format: Character-separated values

- Character-separated values
- Comma-separated values (CSV)
- Crystal Reports (RPT)
- Data Interchange Format (DIF)
- Excel 2.1 (XLS)
- Excel 3.0 (XLS)
- Excel 4.0 (XLS)
- Lotus 1-2-3 (WK1)
- Lotus 1-2-3 (WK3)
- Lotus 1-2-3 (WKS)
- Record style (columns of values)

PRES_NAME	BIRTH_DATE	YRS_SERV	PARTY
XXXXXXXXXXXXXXXXXXXX	31/12/99	-333,333,333	XXXXXXX

---PRESIDENT---
- PRES_NAME
- BIRTH_DATE
- YRS_SERV
- DEATH_AGE
- PARTY
- STATE_BORN
- BIO_DATA

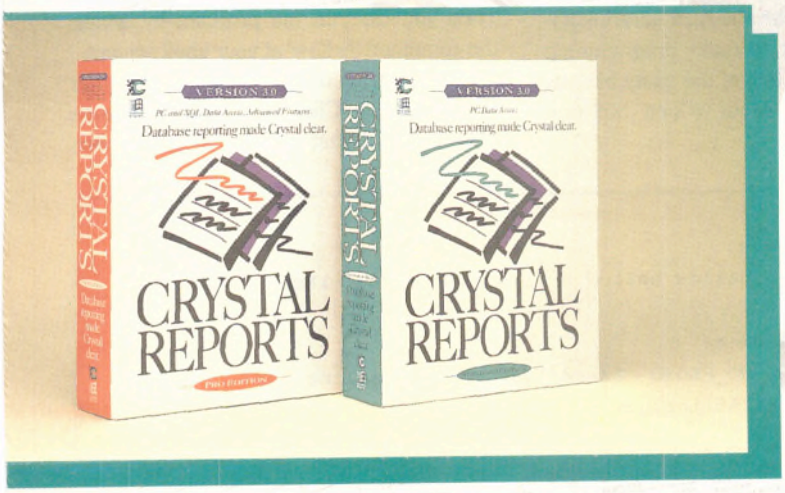
REPORTS

New features in both Crystal Reports Standard 3 and Pro 3 versions include:

- **Fine Tuner** for tweaking the "look" of a report in the Print Preview Window.
- **Data Views** give intuitive field names and can hide formulae and links.
- **Drag & Drop, Formatting Ribbon + Selection Lasso** to speed report design.
- **Mail support** (VIM, MAPI) distributes reports via e-mail.
- **Cross Tab reporting** allows spreadsheet-style comparative analysis.
- **More PC data access** - Use the same intuitive interface to query/report on all supported databases: ASCII, Access, Excel, DB2/2, dBASE, Paradox and Btrieve.
- **OLE 2.0 compatibility** - include graphics, text or tables from other software.
- **Two pass reporting.**
- **Export formats** facilitate information transfer. These include RTF, WordPerfect, Word, Excel, Lotus 123, etc.
- **Integration with Crystal Reports Server** removes report processing and so frees up your machine. And you can schedule reports to be printed after hours or automatically.

Additional features in the Pro 3 Edition include:

- **Free copy of TrueGrid 2.**
- **SQL data access** for IBM DB2/2, Gupta SQLBase, Microsoft SQL Server, NetWare SQL, Oracle and Sybase SQL Server as well as any ODBC-compliant databases.
- **Dictionary builder** for building data dictionaries in simple Data Views.
- **30 extra Print Engine Calls** for optimal report production at runtime.
- **Customisable Print Preview window.**
- **Formula Editor** lets you add your custom functions.



VERSION 3.0
Database reporting made Crystal clear.
CRYSTAL REPORTS
PRO EDITION
VERSION 3.0
Database reporting made Crystal clear.
CRYSTAL REPORTS

Crystal Reports 3 continues to set the standard in database reporting.

CONTEMPORARY Software

Abbey View
Everard Close
St Albans AL1 2PS
Tel: 0727 811999 Fax: 0727 848991

ate one widget, calculate size and position, create the next widget, recalculate size and position, create the next one and so on.

Finally, the call to `XtAppMainLoop()` says that the user interface (such as it is) of our program is ready and the Toolkit should now wait for events and process them in turn.

```
#include <Xm/PushB.h>
#include <stdio.h>

XtCallbackProc my_exit_cb(wgt, myinfo, tkinfo)
  Widget wgt;
  XtPointer myinfo;
  XtPointer tkinfo;
{
  fprintf(stderr, "callback: %s\n", (char*) myinfo);
  exit(0);
}

main(argc, argv)
  int argc;
  char **argv;
{
  Widget mytop, pbutton, XtAppInitialize();
  XtAppContext myappc;

  mytop = XtAppInitialize(&myappc, "Hworld", NULL, 0,
                          &argc, argv, NULL, NULL, 0);
  pbutton = XmCreatePushButton(mytop, "mybutt",
                               NULL, 0);
  XtAddCallback(pbutton, XmNactivateCallback,
    my_exit_cb, (XtPointer) "to exit");
  XtManageChild(pbutton);
  XtRealizeWidget(mytop);
  XtAppMainLoop(myappc);
```

Figure 9 - Handling a click event on the push button

Cooking with Callbacks

So far, the program is all user interface and nothing else. Let's make it execute some of our own code when the button is pressed. We achieve this through a technique called 'callbacks'. These are hooks within each widget for attaching functions to perform the actions we want. Figure 8 illustrates how they work.

Different types of widgets have different callbacks, depending on what type of object the widget is. For example, the Motif widget to contain text has callbacks which are executed when text is being typed in the widget (allowing certain classes of characters to be rejected) and other callbacks which are executed when the user presses <Return> or <Tab> to say that this field is complete (so the final field value can be verified). On the other hand, Motif pushbutton widgets have callbacks relating to the pressing and releasing of the button: `activateCallback()` for when you click on the pushbutton; `armCallback()` for when you press on the pushbutton but haven't released the mouse button yet and `disarmCallback()` when the pushbutton reverts to its normal state having been armed. The only one of these that you normally use is the `activateCallback()`, but the others are handy for experimenting with when you are getting used to Motif.

I will now demonstrate the way `activateCallback()` can be used to print a message and exit the application when the button is pressed. First, we have to write the function to be executed by the callback. When the Toolkit calls this function, it will be passed certain standard information (about which widget's callback this is and the reason - the callback type - it was called) so we always write callback functions with a standard set of arguments (see parameters to `my_exit_cb()` in Figure 9).

Next we must 'register' the callback function, ie tell the Toolkit that this function is to be executed whenever `pbutton` is activated. We do this with the function `XtAddCallback()<N`. Its arguments are the widget that we are dealing with (`pbutton`), which of that widget's callbacks we are attaching our function to (`XmNactivateCallback`), the name of our function (`my_exit_cb`) and finally the 'client data' - any old information that we want to have passed to the function when it's invoked. This last argument means that you can use the same function for lots of callbacks for lots of different widgets; when the function is called, you use the client data to distinguish between the various cases.

Where there's a Widget there's a Way

So you see, Motif programming is easy. What we've done here is the basic structure used by the biggest of programs. Most of the problems you'll meet are not because the task is hard, but because there's an awful lot of low-level detail. And if you structure your program well, and make it modular, you can hide most of this anyway.

> **You can use the same function for lots of callbacks for lots of different widgets**

In the next article we will build on this tiny program. We'll start using many more types of widgets, in particular container widgets which handle layout of groups of other objects. To illustrate this we'll build a simple menu system which you can use for lots of routine tasks. For example, in-house we use a tool like this as a handy way of performing various system management functions; non-specialist users can click on a button instead of having to remember the appropriate Unix commands and all the weird arguments they take.

This program will also introduce you to some of the other areas where Motif is like MS-Windows, especially how you can drive the user interface from the keyboard in addition to the mouse. And from there we'll move on to pop-up and pull-down menus.

■

Niall Mansfield Managing Director, User Interface Technologies Ltd. e-mail: niall@uit.co.uk

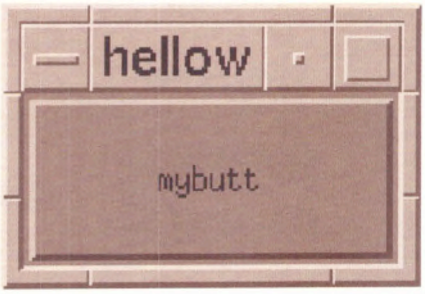


Figure 3 - Output when 'Hello World' is run

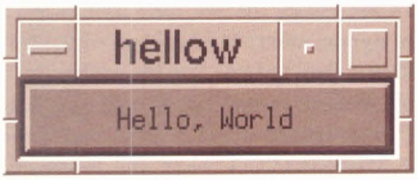


*Figure 4 Changing the *labelString resource*

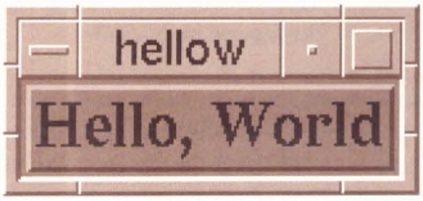


Figure 5 - Set the font and point size set

out' again, just like an MS-Windows button. (Later on we'll see that Motif's look and feel actually has a lot more in common with MS-Windows.) But there are a couple of glitches here: first, it says '`mybutt`' and not '`Hello, World`' as it's supposed to. No problem! We could change the text in the program itself by replacing the string '`mybutt`' with '`Hello, World`' in the call to `XmCreatePushButton()`. This would work, but isn't such a good idea, since the string represents not only the default value for the text to be displayed in the button, but also *the name of the button object.* We'll be using this name later to customise parts of the program. Also, it would be messy having spaces and commas in the name.

So we'll do it differently instead: we'll use Motif (or X) *resources* to change the value of the text. See Peter Collinson's excellent article in *EXE*, April '94 issue if you want to remind yourself about this. There's no need to change the program or recompile. All that's required is to set the `-xrm` as follows:

```
hellow -xrm '*labelString:
Hello, World'
```

This generates the screen output shown in Figure 4. What we've done here is to say that any object in the program which has a resource (think of it as a parameter if you like) called '`labelString`' that assigns the value '`Hello, World`' to the resource, which is the one that specifies the text to be displayed. But how did you know *which* resource or name to use? Good question. The short answer is (alas) read the manual, but see the box *Finding out about resources* for something more helpful. And by the way, the flag '`-xrm`' stands for 'X Resource Manager': the value you give it is taken as a resource specification.

The second problem with our tiny program's output is that the font it's using is so small you can hardly read it. We can fix that too by using a resource setting. The resource '`fontList`' says which font an object is to use. So we can add:

```
-xrm '*fontList:
          *Times*-r-*-240-*'
```

to the previous command line in order to use a Times font, Roman ('`-r-`'), with a size of 24 points ('`-240-`'). Figure 5 shows the final result.

> **The program doesn't know what is going to happen next, but it must be able to handle whatever the user does**

Being resourceful...

There are lots more resources you can set on the command line too. But if we keep on going like this, we'll have an enormously long line to type. So first, let's put these basic resource settings into a file so they can be picked up automatically by the program whenever we run it. Figure 6 lists the resource file for our Hello World program. You can see how these lines relate very closely to the strings you had on the command line. Note that the name of the file is the value of the string given in the call to `XtApplicationInitialize()` in Figure 2. In fact this is the 'application class name' of your program: its value is used precisely for this purpose of extracting resource settings which are to apply to this program and this program only.

The program can now be run with the command '`hellow`'. All the correct settings are being picked up automatically from the file. Command line parameters can still be provided. For instance:

```
hellow -fg pink
```

sets the foreground colour to pink;

```
hellow -bg purple
```

sets the background colour to purple;

```
hellow -name foobar
```

sets the program's name to foobar (notice that the program's icon is called foobar too) and

```
hellow -geometry 100x200
```

makes the window 100 pixels wide by 200 high.

How it works, why it works

Let's go back and look in some detail at the code we've written and the principles behind it. The most important point about Motif programs is that they are event driven. That is, the end-user can choose many different actions at any time (press a button, type some text, pull a scrollbar etc). The program doesn't know what is going to hap-

```
#include <Xm/PushB.h>

main(argc, argv)
     int argc;
     char **argv;
{
```

```
  Widget mytop, pbutton, XtAp-
pInitialize();
  XtAppContext myappc;

  mytop = XtAppInitial-
ize(&myappc, "Hworld",

NULL, 0, &argc,
```

Figure 1 - Ubiquitous 'Hello World' in Motif

A different model for databases

One of the challenges of information processing in the 1990's is how to process larger and larger databases, containing increasingly complex data, without sacrificing response time. **Jim Dorrian** of Arbor Software discusses a possible solution.

Until recently, organisations have tried to target relational database management systems (RDBMSs) for the complete spectrum of database applications. It is becoming increasingly apparent that there are major categories of database applications which are not suitably serviced by relational database systems. Oracle, for example, has built a totally new Media Server for handling multimedia applications. Sybase uses an object-oriented DBMS (OODBMS) in its Gain Momentum product which is designed to handle complex data such as images and audio.

Another category of applications unsuitable for the relational database model is online analytical processing (OLAP). OLAP applications have query and response time characteristics which set them apart from traditional online transaction processing (OLTP) applications. Specialised OLAP servers are designed to give analysts the response time and functional capabilities of sophisticated PC programs with the multi-user and large database support they require.

Differences to OLTP

Online Analytical Processing (OLAP) is primarily involved with reading and aggregating large groups of diverse data. Unlike OLTP applications, OLAP involves many data items (frequently many thousands or even millions) which are involved in complex relationships. The objective of OLAP is to analyse these relationships and look for patterns, trends, and exception conditions. With OLTP the application comprises a large number of relatively simple transactions. The transactions usually retrieve and update a small number of records (usually less than one hundred) that are contained in several distinct tables. The relationships between the tables are generally simple.

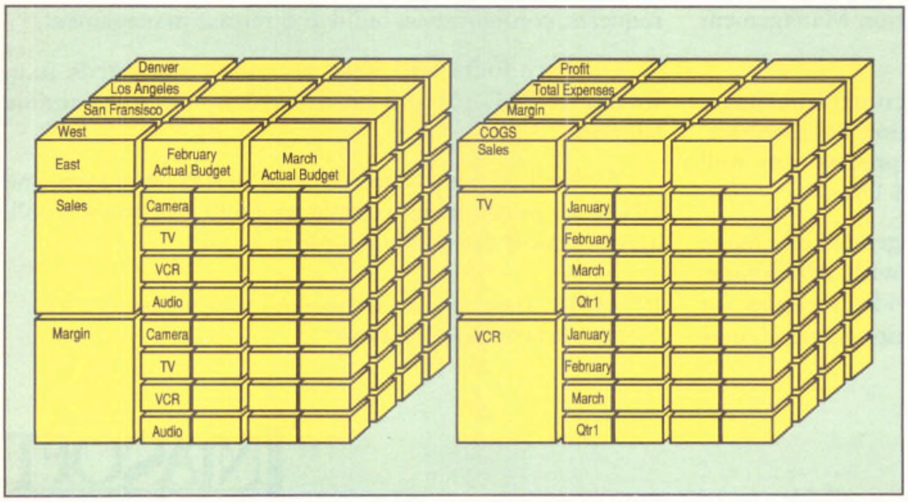


Figure 1 - Slicing and dicing for different view of the data

As an example of an OLAP database, consider sales data which has been aggregated by region, product type, and sales channel. A typical OLAP query might access a multi-gigabyte/multi-year sales database in order to find all product sales in each region for each product type. After reviewing the results, an analyst might further refine the query to find sales volume for each sales channel within region/product classifications. As a last step the analyst might want to perform year-to-year or quarter-to-quarter comparisons for each sales channel. This whole process must be carried out online with rapid response time so that the analysis process is undisturbed.

OLAP queries can be characterised as online transactions which access very large amounts of data eg several years of sales data. They perform analysis of the relationships between many types of business elements such as sales, products, regions, channels. Queries often involve aggregated data like sales volumes, budgeted dollars and dollars spent. Sometimes it may be necessary to compare this aggregate data over hierarchical time periods (monthly, quarterly, yearly).

Data could be presented in different perspectives. For instance sales figures could be shown as sales by region versus sales by channels by product within each region. Furthermore, complex calculations between data elements may be required to perform analysis such as determining expected profit as a function of sales revenue for each type of sales channel in a particular region.

Above all, the OLAP server must be able to respond quickly to user requests so that

Put an end to software piracy!

Meet the growing family of security keys from Software Security.

Each one a specialist at enforcing your license agreement in virtually any user environment you can think of. Whether it's DOS, UNIX, Macintosh or OS/2... whether it's a single user installation or a LAN.

Simply connect the appropriate key to a single user computer, or a non-dedicated file server in a network, and you control all access to your protected application.

Users, however, won't even know it's there. The keys are transparent and won't impact software functionality or the ability to make back up copies. Normal node and LAN operations are unaffected.

Simple. Unassuming. Ever vigilant. Easy to incorporate into your application package. And quite possibly the most profitable hardware investment a software developer can make.

To find out more, call:

(0784) 430060

THE ULTIMATE PROTECTION FOR SOFTWARE PUBLISHERS

All product names are trademarks or registered trademarks of their respective holders.

Software Security International Ltd

21a The Precinct, High Street, Egham, Surrey, United Kingdom, TW20 9HN

National telephone: Egham (0784) 430060 fax: (0784) 430050 International telephone: +44 784 430060 fax: +44 784 430050

➤ CIRCLE NO. 586

users can pursue an analytical thought process without being stymied by the system.

What Are OLAP Servers?

OLAP database servers use multi-dimensional structures to store data and relationships between data. Multi-dimensional structures can be best visualised as cubes of data - and cubes within cubes of data. Each side of the cube is considered a dimension.

Each dimension represents a different category such as product type, region, sales channel and time. Each cell within the multi-dimensional structure contains aggregated data relating elements along each of the dimensions. For example, a single cell may contain the total sales for a given product in a region for a specific sales channel in a single month. Multidimensional databases are compact and easy to understand vehicles for visualising and manipulating data elements that have many interrelationships.

There are a number of common analytical operations which can be performed by the OLAP server. Consolidation involves the aggregation of data. These can be simple *roll-ups* or complex expressions involving interrelated data. For example, sales offices can be rolled-up to districts and districts rolled-up to regions.

OLAP data servers can also go in the reverse direction and automatically display detail data which comprises consolidated data. This is called *drill-downs.* Consolidation and drill-down are an inherent property of OLAP servers.

Another feature is the ability to look at the database from different viewpoints. This is refereed to as *'slicing and dicing'*. One slice of the sales database might show all sales of product type within regions. Another slice might show all sales by sales channel within each product type. It is like rotating a child's alphabet block and seeing the different letters on each side depending on how you look at the block. Slicing and dicing is often performed along a time axis in order to analyse trends and find patterns. Figure 1 shows four different views of a database.

Finally, OLAP servers have the means for storing multidimensional data in a compressed form. This is accomplished by dynamically selecting physical storage arrangements and compression techniques that maximise space utilisation. Dense data (ie data that exists for a high percentage of dimension cells) are stored separately from sparse data (ie a significant percentage of cells are empty). For example, a given sales channel may only sell a few products. In this case the cells that relate sales channels to products will be mostly empty and therefore sparse. By optimising space utilisation, OLAP servers can minimise physical storage requirements, thus making it possible to analyse exceptionally large amounts of data. It also makes it possible to load more data into computer memory which helps to significantly improve performance by minimising physical disk I/O.

RDBMSs lack multidimensional

Relational database management systems and their associated front-end tools have many limitations when it comes to online analytical processing. First, relational databases are fundamentally incapable of managing multidimensional data structures. Instead, data must be stored in interrelated two-dimensional tables which represent flattened out multidimensional structures.

Flattening out is analogous to taking a three dimensional model of a building and flattening it out into numerous architectural drawings. In order to reconstruct the model it is necessary to find drawings that are connected to each other and combine the drawings to form a two-dimensional picture of the three-dimensional model. Reconstructing the model by linking two-dimensional drawings is difficult, resource intensive and prone to error.

A large collection of relational tables representing multiple dimensions is difficult for end-users to understand and use. Users must figure out which data is located in which table, how tables are linked to each other and how to construct the correct SQL query using either a front-end tool or SQL itself to link the tables together.

Crippling performance

Multiple joins and poor performance become an issue since users must also be concerned with optimising queries in order to obtain reasonable performance. A Sales database might contain the tables and data elements given in Figure 2. The normalised database saves space because each product, sales office, district office, and region appears only once in the database. However, a query that needs to summarise and compare data by sales office, district, and regions has to join the four tables together. Relational joins are an exhaustive process. Each record must be matched against each row (or an index entry for each row) in every other table. In this database the join might require up to one trillion matches (1,000,000 * 1,000 * 100 * 10). A database consisting of 10 or more tables would take several times more than this. The matching process must be performed for every OLAP query. This is why many users performing many OLAP queries can bring an RDBMS to its knees.

Table 1	Product Sales/Sales Office	1,000,000 rows
Table 2	Product Description	1,000 rows
Table 3	Sales Office/District Cross Ref	100 rows
Table 4	District/Region Cross Reference	10 rows

Figure 2 - Normalised data penalises performance

Because multiple joins absorb expensive machine resources and are difficult for users to understand, most relational vendors advise that users create special denormalised databases for decision support processing. Denormalised tables are tables that are pre-joined (ie all tables are combined into one table). This has the advantage that it avoids time consuming joins but there are several disadvantages to this approach. The disadvantages are that it is expensive in terms of both performance and system resources. Figure 3 illustrates the differences between normalised and denormalised database designs.

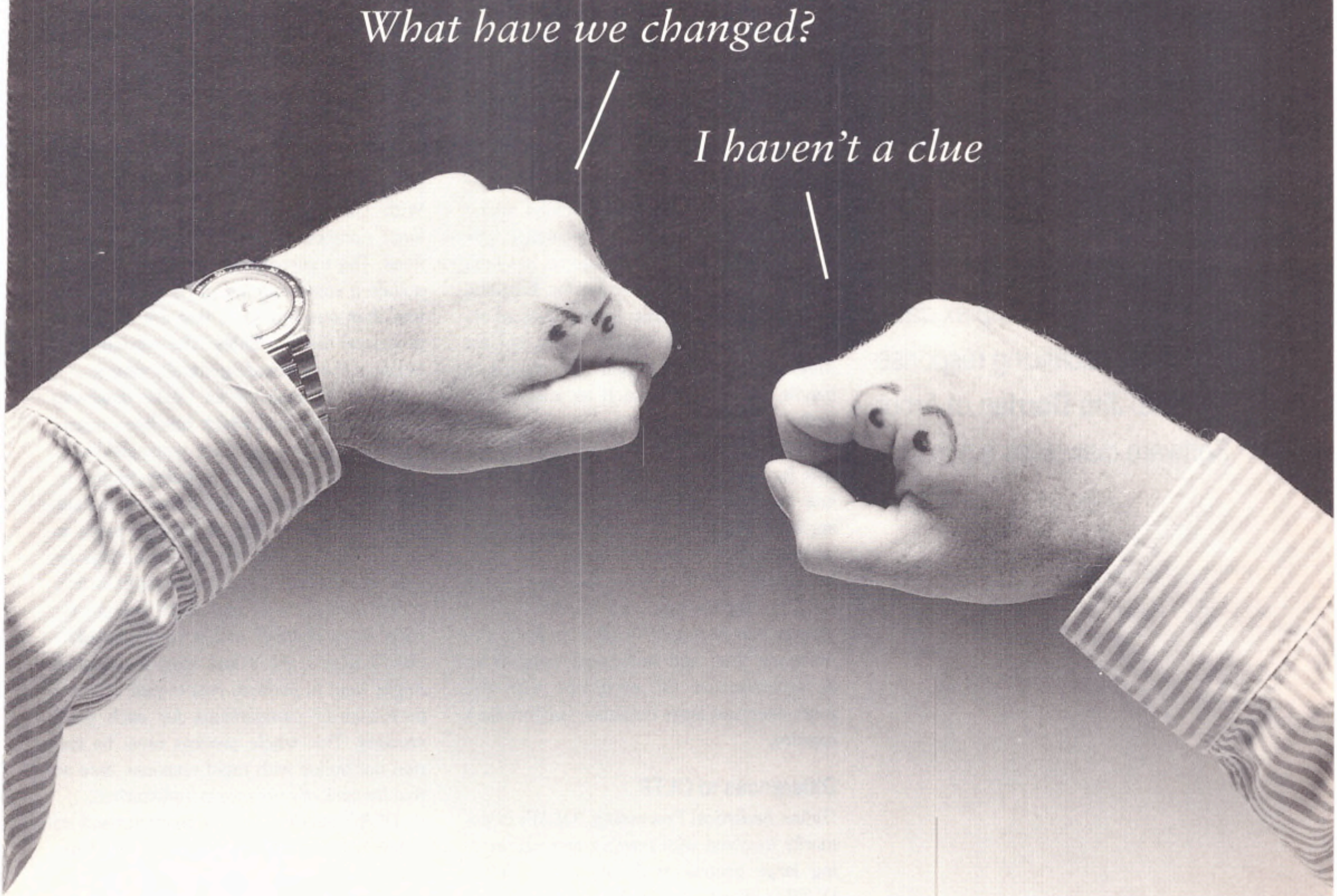
No alternatives...

Users are caught in a no-win situation when using relational databases to service online analytical queries. If the database is broken up into smaller tables then performance suffers because of complicated table joins. If users combine tables into a denormalised table then they must deal with extremely large tables which also impact performance as well as consume space.

The only improvement RDBMS vendors can recommend is to upgrade to more powerful and expensive hardware. Because RDBMSs do not handle online analytical database efficiently, major RDBMS vendors have developed parallel query technology which uses a 'brute force' approach to addressing the query performance of large analytical databases. Parallel processing allows many CPUs (and machines) to scan a database simultaneously. The results are then merged and sorted together.

While parallel processing can be used to speedup certain types of queries (eg queries that scan an entire database) it is not suitable for most other types of queries which cannot be broken up into parallel processes. For example, parallel query options do not improve performance when searching for specific ranges of data. Parallel query options require very expensive symmetrical multi-processing hardware

Does your left hand know what the right is doing?



Get your hands on Intasoft's Configuration Management tools and it will!

SMS is our full-featured version control package: perfect for preventing users overwriting each others' changes and for automating the build process – as well as telling you *Who?*, *What?*, *When?* and *Why?*

Full configuration and change management can make anyone all fingers and thumbs. Not with **AllChange**, Intasoft's brilliant package which provides facilities for product design, baselines, life-cycles, user roles, change requests, configuration build and release management.

Our products will help you with standards (e.g. ISO9001, BS5750) and are backed by our outstanding support.

With all of this available for MS-DOS, Windows and UNIX at prices from as little as £490, we think you'll agree Intasoft are handy people to know!

INTASOFT, Tresco House, 153 Sweetbrier Lane, Exeter, EX1 3DG.
Tel 0392 217670 Fax 0392 437877

INTASOFT

Quality tools for professional software developers

NEW *SMS Version 4 for Windows Now available*

NEW *AllChange – Change control & configuration management system*

➤ CIRCLE NO. 587

which grows even more expensive as the size of the database grows.

Limited intrinsic Analysis

RDBMSs do not have inherent capabilities to manage basic OLAP operations such as consolidations, drill-down, slicing and dicing and complex inter-table and intra-table data relationships. They have a limited set of built-in functions which only operate in a single direction. For example, SQL can only `SUM` and `AVG` the same field in all records but cannot `SUM` and `AVG` several columns in a single row. Users can create an expression that involves several fields in a record in a single table, but these expressions cannot reference other rows in the table nor can they reference other tables unless the tables are first joined in a view. These limitations are intrinsic in the relational model because it only works with flat two-dimensional tables.

As a result, any analytical operation which goes beyond the basic relational selection and update commands has to be handled within a front-end application. These usually require a great deal of customisation and programming in order to achieve good performance and requisite functionality eg complex cross-element calculations, specialised reporting etc.

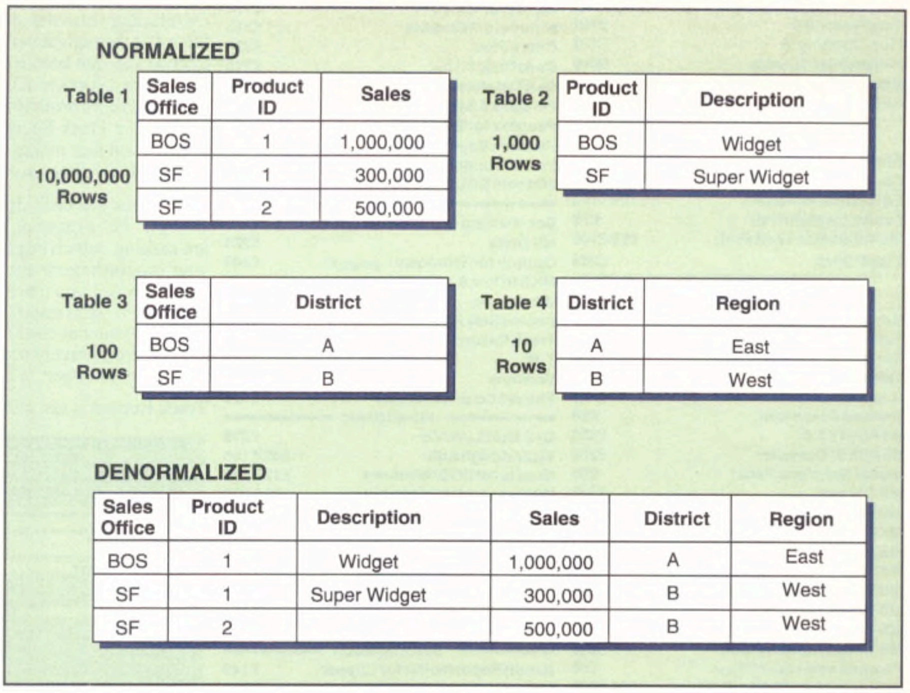
Hogging the network

The front-end applications that run on client PCs have first to extract information from the relational database then move the data from the RDBMS to the front-end application over a network. Since OLAP applications involve very large databases, this means that it is necessary to move large amounts of data from the RDBMS to the front-end application before the front-end tool can perform necessary consolidations, drill-downs, or slice and dice operations.

Transferring large amounts of data not only severely degrades application performance it also quickly saturates the network which in turn impacts the performance of all other users on a network. It is not uncommon for analytical queries to take many minutes or hours to complete because of the loads they place on the network. In contrast, OLAP servers can directly perform analytical operations in the server which means only selected data result sets have to be transmitted to the front-end application for display and further analysis.

Poor consolidation

In some cases it may be possible to consolidate and summarise data in the RDBMS by using the `SELECT` command in conjunction with built-in functions such as `SUM` and `GROUP BY`. These are simple summarisation which cannot be rolled-up or drilled down.



NORMALIZED

Table 1
10,000,000 Rows

Sales Office	Product ID	Sales
BOS	1	1,000,000
SF	1	300,000
SF	2	500,000

Table 2
1,000 Rows

Product ID	Description
BOS	Widget
SF	Super Widget

Table 3
100 Rows

Sales Office	District
BOS	A
SF	B

Table 4
10 Rows

District	Region
A	East
B	West

DENORMALIZED

Sales Office	Product ID	Description	Sales	District	Region
BOS	1	Widget	1,000,000	A	East
SF	1	Super Widget	300,000	B	West
SF	2		500,000	B	West

Figure 3 - Denormalised database take up unecessary space

However, even though these are simple operations they are still quite expensive. In order to summarise data it is usually necessary to run a program that reads all of the rows in the database. Several applications, all submitting summarisation queries against a relational database at the same time, will quickly bring the system to a halt.

Besides being very expensive, users may have to wait several hours or overnight for processing before they can get answers to their queries. To get around the problem of running expensive and time consuming summarisation queries, organisations often create summary tables that store frequently used summarised data. These must be predefined by users. Any summaries not predefined are therefore not available to end-users. This is a severe limitation to online analytical work.

The proliferation of summary tables - one for each type of consolidation - results in a very complex and confusing environment. Users must be aware of which summary tables exist and which information is contained in each summary table. Summary tables must be periodically refreshed in order to keep them synchronised and current. Extracts and refreshes must be scheduled so that they do not interfere with regular processing and so that they do not overly stress the RDBMS.

Since there is a lag time between extract and refresh programs for each table, tables may not be all synchronised at a particular point in time. This can delay analysis or lead to inconsistent results.

Whether users are using summary or detail databases, complex analytical queries have unpredictable effects in a relational environment. Because it is difficult to predict the performance degradation impact of complex relational queries, users are rarely permitted ad hoc access to relational databases. They are usually limited to using preconstructed queries which are designed to have minimal system-wide impact. Users are allowed to modify these slightly by changing search and sorting criteria but cannot build their own ad hoc queries. As a result, business analysts are heavily restricted in the types of online analysis that they can perform.

Summary

In summary, the key feature of OLAP servers which set them apart from other types of data management servers is that they logically organise data in multiple dimensions. This permits users to analyse complex data relationships quickly and easily. The database itself is physically organised in such a way that related data can be rapidly retrieved across multiple dimensions. OLAP servers are very efficient when storing and processing multidimensional data. Even on low-cost platforms, OLAP servers can provide quick response time while accessing large multidimensional databases. ■

Jim Dorrian is the CEO of Arbor Software which has implemented an OLAP server called Essbase. Further information can be obtained from Arbor on 0101 408 7277166.

All over the world software developers are looking for solid solutions to piracy protection and revenue enhancement for their software applications. They're looking for secure solutions that are reliable, compatible and transparent. And that's why they call Rainbow Technologies — the worldwide leader in software and revenue protection.

Technologically Superior

Sentinel keys are supported by the largest R&D department in the software protection industry — renowned for technological innovations like the world's smallest key. And they guarantee support for virtually any language as well as future technologies like Chicago and even Cairo.

Why did 10,000 software developers make this call?

Most Widely Used Keys

It's true, more than 10,000 software developers have called on Rainbow Technologies for the Sentinel® Family of hardware keys. More than any other in the world. In fact, there are over 4,000,000 Sentinel keys in use today. That's because they are the most technologically sophisticated. Take the SentinelSuperPro™, for example. Featuring next generation ASIC technology, this key provides the highest security and most flexibility available. Anywhere. It's just one in a family of keys that support virtually any computer platform including Windows, NT, WIN32S, UNIX, OS/2, open systems and Macintosh.

Call For Yourself

It's no wonder seven out of the ten largest software companies use Sentinel keys from Rainbow. With direct support from offices and distributors in 41 countries worldwide, no one else can match Rainbow's reputation for service. Want to know more? Call Rainbow today. Ask for a free copy of *The Sentinel Guide To Securing Software*. And while you're at it, ask about our complete line of Sentinel Evaluation Kits.

SENTINEL
Securing the future of software

CALL
44 0932 570066
FOR YOUR FREE GUIDE
TO SECURING SOFTWARE

RAINBOW TECHNOLOGIES

USA/ASIA/LATIN AMERICA 714/454-2100 ■ **RAINBOW LTD./U.K.** 44 932 570066 ■ **RAINBOW MICROPHAR/FRANCE** 33 1 47 38 21 21 ■ **RAINBOW GmbH/GERMANY** 49 89 32 17 98 0

Worldwide distributors: **ARGENTINA:** Agri-Aid, S.A. 54 1 46 8364 • **AUSTRALIA:** Loadplan 61 3 690 0455 • **BENELUX:** Introcom Security Key 31 74 430 105
BRAZIL: Mips Sistemas Ltda. 55 21 507 1839 • **BULGARIA:** KSIMETRO 3592 79 14 78 • **CHILE:** ChileSoft Ltda. 56 2 639 8892 • **COLOMBIA:** Construdata 57 1 610 8549 • **CROATIA:** FELMET 385 41 272 477
CZECHOSLAVAKIA: ASKON International 42 231 66 775 • **GREECE:** Byte Computer S.A. 301 924 17 28 • **HONG KONG:** Computers and Peripherals 852 515 0018
HUNGARY: Polyware Kft 36 76 481 236 • **IRAN:** Gam Electronics 982 122 22374 • **ITALY:** BFI IBEXSA SPA 39 23 31 00535 • **ITALY:** Siosistemi 39 30 24 21074
JAPAN: Giken Shoji Co., Ltd. 81 52 972 6544 • **KOREA:** Genesis Technologies Corporation 82 2 578 3528 • **MEXICO:** Impex Computacion, S.A. de C.V. 91 800 66 466
MIDDLE EAST: Hoche International 44 81 459 8822 • **MOROCCO:** Futur & Soft 212 2 40 03 97 • **PHILIPPINES:** Techno Products and Services 63 2 732 5174 • **POLAND:** DAGMA SP.ZO.O 48 32 525 200
SCANDINAVIA: Perico A/S 47 22 49 1500 • **SOUTH AFRICA:** EPSYS (Pty) Ltd 2711 886459 • **SPAIN:** MECCO 34 3 422 7700 • **SWITZERLAND:** IBV AG 41 1 741 2140
SWITZERLAND: Safe Compaid S.A. 41 2 421 0385 • **THAILAND:** BCS International 66 2 319 4451 • **TURKEY:** BIMEKS, Ltd. 90 216 348 3508 • **VENEZUELA:** HRT-M Osers 58 2 261 4282

➤ CIRCLE NO. 588

Migrating dBASE apps to Windows

With the announcement of dBASE for Windows, **Mike Lewis** looks into porting from dBASE for DOS to the new environment.

Borland has finally announced dBASE for Windows. With this one product it hopes to take control of the Windows database arena. If it fails, Borland will fail. The company's future is tied intimately to its success. The stakes are high. IDC estimates there are in the order of 6.7 million dBASE applications 'out there'. Just think. There are billions of lines of dBASE code in the world...

Compatibility is therefore a key issue. In this article I have been investigating the level of compatibility between dBASE for DOS and dBASE for Windows. The information in this article is based on a beta version of dBASE for Windows. Some of the details might change in the final release.

When you make the move to Windows, you can continue to work with the same data files, use the same programming language and generally apply the same principles and techniques that you have always used. In practice, though, things are not that simple. To get the best from dBASE for Windows, you really need to get to grips with some new ways of working.

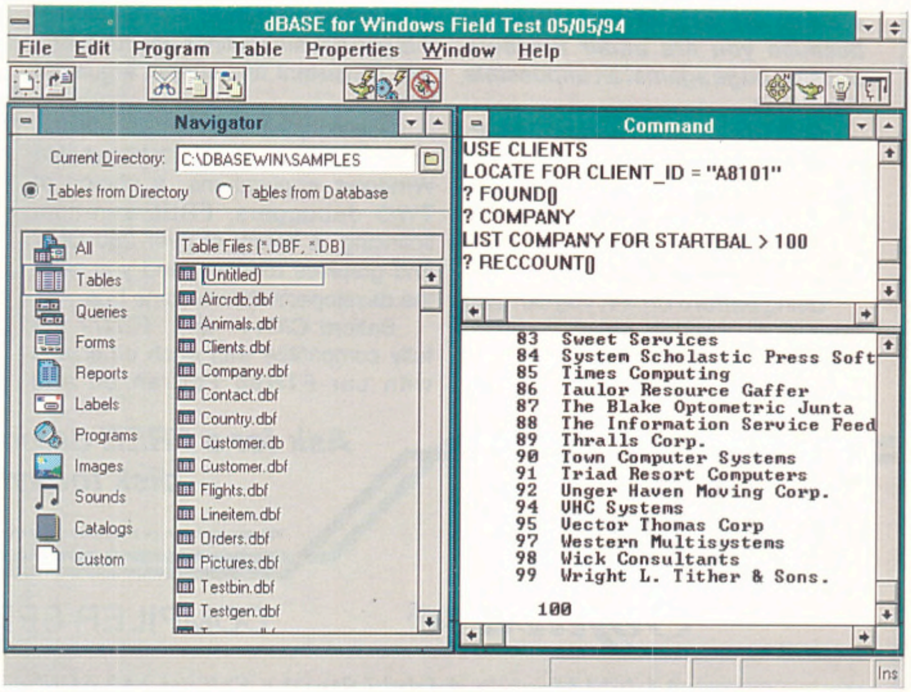
It runs DOS, but it's ugly...

For the most part, Borland has done a good job of porting the dBASE language to Windows. The front end comprises a two-part interface as shown in Figure 1. The Navigator (left) serves a similar function to dBASE IV's Control Centre; the Command window replaces the dot prompt. When you type DO MYAPP in the Command window, dBASE for Windows will cheerfully run your dBASE IV PRG file, which will function just as it did before. Several dBASE IV commands are no longer supported (see text box), but these are simply ignored when encountered in a program.

When you run a dBASE IV (or dBASE III Plus) program in this way, dBASE for Windows opens a 'virtual DOS' window, that is, a movable, resizable window which simulates the 25 x 80 DOS screen. All the program's screen output goes to this window. If the program opens its own menus or windows, these appear within the DOS window.

Although it's easy to move your existing

The virtual DOS window is a picture of ugliness, not least because of its unpleasant monospaced font



```
USE CLIENTS
LOCATE FOR CLIENT_ID = "A8101"
? FOUND()
? COMPANY
LIST COMPANY FOR STARTBAL > 100
? RECCOUNT()
```

```
     83  Sweet Services
     84  System Scholastic Press Soft
     85  Times Computing
     86  Taulor Resource Gaffer
     87  The Blake Optometric Junta
     88  The Information Service Feed
     89  Thralls Corp.
     90  Town Computer Systems
     91  Triad Resort Computers
     92  Unger Haven Moving Corp.
     94  UHC Systems
     95  Uector Thomas Corp
     97  Western Multisystems
     98  Wick Consultants
     99  Wright L. Tither & Sons.

       100
```

Figure1 - Navigator and Control window

COMPARE OUR PRICES

Diagrammers

EasyCASE Prof. for DOS/Win	£345/£645
EasyCASE Des. for DOS/Win	£555/£925
EasyFlow v.8.0	£149
Flow Charting III	£120
InfoModeler Desktop	£399
MetaDesign for Windows v.4.0	£239
RFflow v3.0 - *lowest UK price*	£65

Editors

BRIEF	£115
Codewright Lite/Win. or NT	£109/£175
Ed for DOS/Windows	£129/£165
Evolve for Multi Edit	£75
Multi Edit/Multi Edit Prof.	£69/£109
Visual Slick	£205

Languages

Borland C++ v4.0	£269
Turbo C++ Visual Edition	£79
Turbo C++ v3.0 for DOS	£55
Turbo Pascal v7.0	£75
Borland Pascal	£229
Borland Assembler	£69
dBASE IV 2.0	£385
dBASE IV Compiler	£235
Visual Solutions Pack	£55
MS Access	£249
MS Fortran 5.1	£95
MS FoxPro DOS/Windows	£269
MS MASM 6.11	£105
MS Visual Basic DOS/Windows Std.	£95
MS Visual Basic DOS/Windows Prof.	£229
MS Visual C++ Std.	£75
MS Visual C++ Prof. 1.5	£289
MS Visual Control Pack	£65
PowerBasic - *lowest UK price*	£85
Symantec C++ Std 6.1/Prof.	£69/£259
Smalltalk V DOS/Windows	£65/£249
Watcom C 32 for DOS - *lowest UK price*	£119
Watcom C/C++ 10.0 CD - *special offer*	£175
Watcom C/C++ 10.0 CD - Docs	£299
CA-Realizer	£69

Best Sellers

Database	
Agility for VB/VC++	£179
Btrieve for Windows	£315
C-tree Plus	£329
CodeBase 5.1/++	£215
Q+E Database Library	£320
Paradox 4.5 for DOS	£385
Paradox for Windows	£99
Paradox Engine & DFX	£115
PowerBuilder Desktop - *special offer*	£199
Watcom SQL Developers Ed.	£265
Tools	
Doc-To-Help	£209
MS Delta	£225
Optlink for Windows - *special offer*	£249
MKS RCS v.6.1/Toolkit v.4.1	£229/£169
WinScope - *special offer*	£49
SourceSafe for DOS/Windows	£185/£255
Track Record	£110
T-lib	£89
Versions	£149
Phase3 Code Generator	£129
Visual Basic	
Q+E MultiLink/VB	£239
VBXref/High Edit	£65/£165
Muscle for DOS/Windows	£129/£95
VBTools for DOS/Windows	£95/£85
VBAssist	£85
Data Widgets	£89
Designer Widgets	£89
PDQComm DOS/Windows	£75/£109
QuickPak Prof. DOS or Windows	£149
CodeBasic 5.1	£119
Integra VDB	£369
Miscellaneous	
Bandit Reportwriter for Clipper	£149
Crystal Reports v.3.0 Std./Prof.	£85/£275
Hijaak for DOS or Windows	£109
Image Format Library	£345
Multimedia Toolbook - *special offer*	£99

*All prices exclude £7.50 carriage and VAT.

Watcom C/C++ 10.0 - only £175*

*CD-ROM with online manuals

PowerBuilder Desktop only £199

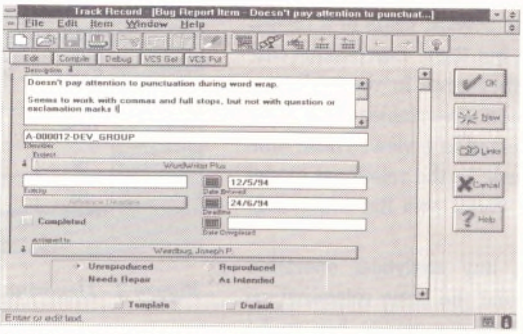
Skill shortage in PowerBuilder!

Track Record from the developers of BRIEF ®

Developing software means juggling hundreds of details. Let just one of them fall through the cracks, and you're in for headaches. Now with Track Record you can keep track of practically everything relating to your work - from bugs to beta test sites. It lets you quickly and easily plan tasks, store and retrieve information and keep a detailed history of who did what, when. As you use Track Record, it contructs an intricate web of interrelated information that makes tracking down details a snap. *What bugs are outstanding ? What are my deadlines ? Who found the 'memory' bug ?*

With Track Record's dynamic database you keep up to date as your project changes. For example, Track Record's reports can show you the progress you are making, which bugs have been found-*and fixed*, or even what part of your program needs the most work - the possibilities are endless ! With Track Record you can hone in on "todays bug reports" or take a look at the big picture "beta testers listed by company". And with its configurable database you can customise Track Record to fit your needs. It's a whole new kind of productivuty tool for Microsoft Windows, designed specifically for software developers. Call for the Track Record 'White Paper'.

Track Record is not only the best bug tracking system but also the best value.



Track Record - [Bug Report Item - Doesn't pay attention to punctuat...]

Doesn't pay attention to punctuation during word wrap.

Seems to work with commas and full stops, but not with question or exclamation marks !

A-000012-DEV_GROUP

WordWriter Plus

Date Entered: 12/5/94

Deadline: 24/6/94

Assigned to: Weedbug, Joseph P.

Enter or edit text

Track Record costs £110* per user, £520* for 5 users and £990* for 10 users

Once you're added a new item to Track Record's database, it becomes part of an interrelated "web" of project information.

Only £110*

SoftWerk UK Ltd.

50 Lion Meadow, Steeple Bumpstead, Haverhill CB9 7BY
Telephone : (0440) 730121 Facsimile : (0440) 730911

➤ CIRCLE NO. 589

Rogue Wave "the Portable" C++ Class Libraries

* * * * NEW * * * *

Heap.h++ - a new, memory management library to *increase the performance* of your C++ and C applications *today!*

Heap.h++ is a memory management library that enables C and C++ programs to *run faster* and *use less memory*, with very little work from the software developer. Simply link existing programs to the ready-built library and run upto 3 times as fast - *immeadiately.*

Designed to increase speed and reduce memory used Heap.h++ significantly reduces the overhead of global operators new and delete as well as malloc(), calloc(), realloc(), and free(). Object interface also provides user control of the memory policy.

Heap.h++ is available now for most UNIX workstations, and if you place your order **before the 1st, July, 1994** you can purchase at our launch price of £699 saving - £150 on your purchase.

* *

We have C++ class libraries available for GUI, Drawing, Charting, Database, Mathematical, Scientific, Engineering, and Financial applications.

Tools.h++ - the portable foundation class library available for DOS, Windows 3.X, Windows/NT, OS/2, MAC, UNIX, and others...

An industry standard library which is a complete toolbox of over 100 C++ classes. It includes all the fundamental data structures and functions needed to do virtually any C++ programming task - includes template classes. All classes support persistence, and the library now includes full internationalisation support and is multi-thread safe.

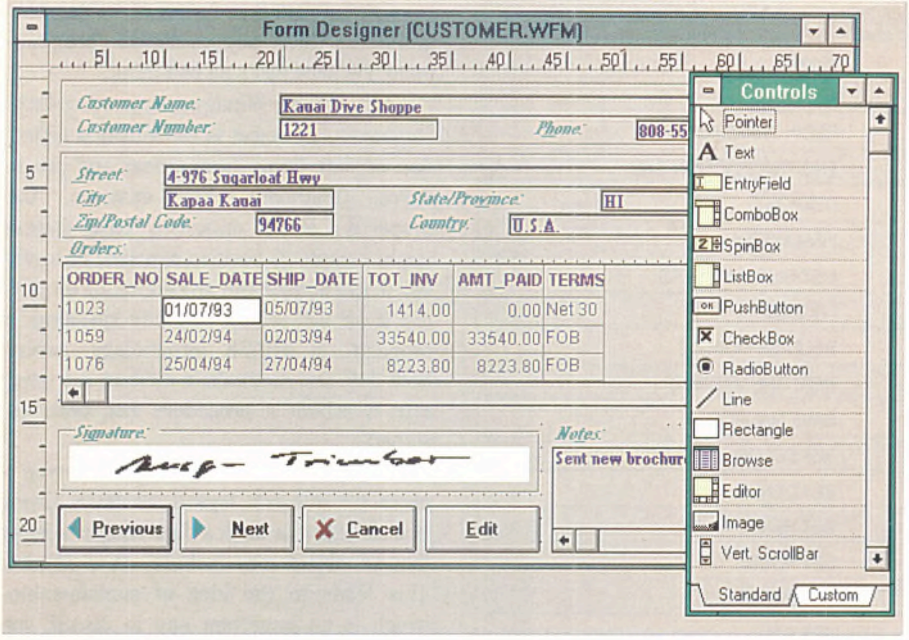
* *

For more information on our C++ class libraries CALL US NOW!

Hypersoft Europe

Northern Office: **0602-376550**
Grove House, 13 Main Street, Keyworth, Notts., NG12 5AA
Southern Office: **0273-834596**
PO Box 901, Hassocks, West Sussex, BN6 9ZS
email: hypersoft_euro@cix.compulink.co.uk

➤ CIRCLE NO. 590



Form Designer [CUSTOMER.WFM]

Customer Name: Kauai Dive Shoppe
Customer Number: 1221 Phone: 808-55

Street: 4-976 Sugarloaf Hwy
City: Kapaa Kauai State/Province: HI
Zip/Postal Code: 94766 Country: U.S.A.

Orders:

ORDER_NO	SALE_DATE	SHIP_DATE	TOT_INV	AMT_PAID	TERMS
1023	01/07/93	05/07/93	1414.00	0.00	Net 30
1059	24/02/94	02/03/94	33540.00	33540.00	FOB
1076	25/04/94	27/04/94	8223.80	8223.80	FOB

Signature:

Notes: Sent new brochur

Figure 6 - To create a form, you drag objects from a palette

cases we are creating a new form object, based on dBASE's built-in Form class. A class is a sort of master object - a skeleton set of properties on which actual objects are based. Every object is an instance of one or other class. dBASE comes with 18 built-in classes and also allows programmers to create classes of their own.

Having defined the form object, we proceed to create a push button object, which is contained within the form. We then assign values to the button's properties. Finally, we open the form by executing its `Open<n> method`. The method actually draws the form on the screen, along with all its contained objects.

In general, executing a method is simply a matter of calling it, just as you would call any other procedure or User Defined Function (UDF). In addition, some methods get executed automatically, in response to events. A push button object, for instance, has an `OnClick` method which is executed whenever the user clicks on the button. If you want to give users an easy way of, say, running a search, you could assign the search routine to the `OnClick` method. Then, when the users want to run the search, they simply press the button.

Creating methods

How do you go about writing a method? In exactly the same way as you write any other procedure or function. And having done so, you assign it to the appropriate property using an ordinary assignment statement. That's right - you can now assign a procedure or function to a variable just as if it was a character string or date. Figure 5 shows an example of this.

Alternatively, you can use an entirely new dBASE concept: the code block. A code block is a bit like a UDF in that it performs an action or returns a value. It is also a bit like a data item in that it can be stored in a variable or passed as a parameter. Essentially, it consists of a series of expressions or commands, separated by semicolons, contained in curly brackets. Yes, Clipper programmers, this is the same code block that you have been using for several years now - but watch for differences in syntax.

What makes the form designer unusual is that it is a two-way street

Two-way RAD

If all this sounds too complicated, there is an easy way of avoiding it. Instead of cutting your own object-oriented code, you can get the dBASE for Windows form designer (see Figure 6) to generate it for you. Now, I know all the arguments against using code generators, but they really don't apply in this case. For a start, the form designer doesn't merely generate data-entry forms that you then have to integrate into your application. It generates the application's entire user interface.

A dBASE for Windows form is an executable program in its own right. When you finish designing the form, the designer generates dBASE source code and stores it in a file with the extension WFM. You run this code just as you would a PRG file, that is, by typing `DO` in the Command window, by double-clicking on the file in the Navigator (which is roughly equivalent to dBASE IV's Control Centre), or by specifying the filename as a parameter when you launch dBASE for Windows.

What makes the form designer unusual is that it is a two-way street. As with other code generators, you can open the generated program in a text editor and hand-modify it. But unlike other code generators, your modifications are still visible to the designer. If you edit the code to alter the position of a field, the field will keep its new position the next time you open the form in the designer. If you add a chunk of custom code, the code will still be there when you regenerate the program. This demolishes the main argument against using code generators - that hand-changes are lost each time you regenerate the code.

Form drawing

The mechanics of the form designer are straightforward. You drag Windows controls - edit fields, radio buttons, scroll bars, etc - from the 'object palette' to the form. All the built-in dBASE form objects are visible in the palette, as are any custom classes that you have defined. Incidentally, dBASE for Windows also supports Visual Basic-compatible custom controls; you can place these on the form designer's object palette or call them directly from your code.

Having placed objects on the form, you set about assigning properties to them. To do this interactively, you right-click on the object, then select `Object Properties` from the resulting menu (see Figure 7). This opens the object inspector. The object inspector is simply a dialog in which you specify the object's colour, font, help ID and other properties.

Next comes the most important part: assigning methods - especially event methods - to the objects. If you are using the form designer to create your application's user interface, your methods will embody the full functionality of the application. You would write these methods as procedures or UDFs, which you would probably store in a procedure file. You would then assign the methods to the appropriate objects. You do this simply by typing the procedure or function names against the relevant properties within the object inspector.

I'm sure you will agree that this is a fairly painless way of building a large application. You still have to code the heart of the app by hand, but you do so using the standard dBASE language - object-oriented or procedural, as you wish - and you can forget about programming the user interface, which is handled entirely by the form designer.

Introducing EasyCASE® 4.0

- Provides Full CASE Tool Capabilities
- Powerful Analysis and Design Tool
- Easy to Use
- Affordable

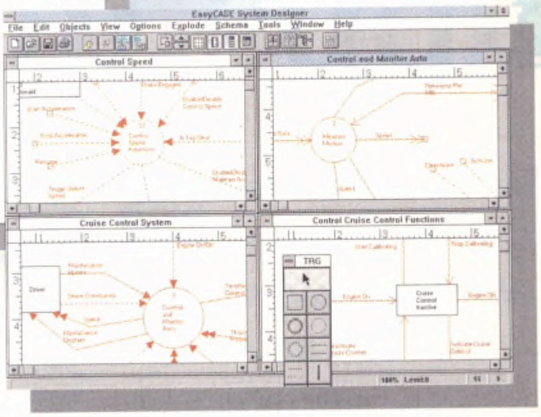
Supports Multiple Methodologies

- Yourdon/ De Marco/ Constantine
- Ward-Mellor/ Hatley
- SSADM
- Gane & Sarson
- Shlaer-Mellor
- Martin/ Jackson
- Chen/ Merise/ IDEF1X
- Bachman

Features

- MS-DOS and Windows Versions.
- Object-Oriented Editing.
- Extensive Documentation.
- Rule-based Methodology
- Monitor.
- dBase Compatibility.
- Methodology Guide.
- Instructive Tutorial.

New workgroup for networks edition now available.



GREAT WESTERN INSTRUMENTS LTD

Tel: 0272 860400

Great Western Instruments Ltd

Redwood House Bristol Road Keynsham Bristol BS18 2BB England

© 1993 Great Western Instruments Ltd. All rights reserved. EasyCASE is a registered trademark of Evergreen CASE Tools Inc.

Evergreen CASE Tools

➤ CIRCLE NO. 591

BLEEP ... BLEEP ... BLEEP ... BLEEP ...

Is this the sound of your PC as it reboots yet again when you run your "fully tested" program? Or is it a case of expletives deleted because you are under pressure chasing bugs against an impossible deadline?

Salford's 32-bit C/C++ and Fortran compilers have unique compile-time options that can ensure you write 100% watertight programs by AUTOMATICALLY pinpointing those hard-to-find errors.

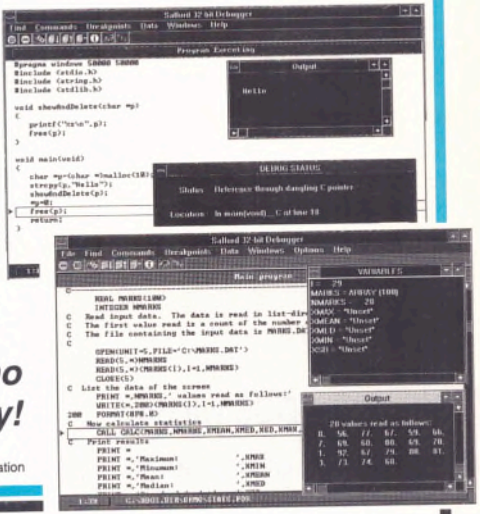
Using ***Salford C/C++***, you can stop references through unset pointers; never access unallocated memory and always allocate/deallocate memory consistently.

With ***Salford FTN77 Fortran***, you can locate arithmetic overflows and array bound errors; home in on the use of undefined variables and be certain that you call subroutines and functions with consistent and correct argument lists.

Now couple these benefits with the ability to create not only DOS but also Windows applications, a choice of TWO debuggers, FREE run-time licencing, a library of over 250 DOS and graphics routines, and you have the developer's dream come true.

Salford C/C++ and FTN77 are fully compatible with each other and with our FTN90 Fortran 90 and PASCAL compilers, so mixed language programming is a cinch!

Call or fax us right now and find out for yourself. Five minutes now could put you months ahead.



```
#pragma window 50000 50000
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void showAndDelete(char *p)
{
   printf("%s\n",p);
   free(p);
}

void main(void)
{
   char *p=(char *)malloc(10);
   strcpy(p,"Hello");
   showAndDelete(p);
   *p=0;
   free(p);
   return;
}
```

```
Hello
```

```
Status   Reference through dangling C pointer
Location   In main(void)  C at line 18
```

```
C
      REAL MARKS(100)
      INTEGER NMARKS
C   Read input data.  The data is read in list-dir
C   The first value read is a count of the number 
C   The file containing the input data is MARKS.DA
C
      OPEN(UNIT=5,FILE='C:\MARKS.DAT')
      READ(5,*)NMARKS
      READ(5,*)(MARKS(I),I=1,NMARKS)
      CLOSE(5)
C  List the data of the screen
      PRINT *,NMARKS,' values read as follows:'
      WRITE(*,200)(MARKS(I),I=1,NMARKS)
200   FORMAT(8F8.0)
C   Now calculate statistics
      CALL CALC(MARKS,NMARKS,XMEAN,XMED,XSD,XMAX,
C   Print results
      PRINT *
      PRINT *,'Maximum:               ',XMAX
      PRINT *,'Minumum:               ',XMIN
      PRINT *,'Mean:                  ',XMEAN
      PRINT *,'Median:                ',XMED
```

```
I =    29
MARKS = ARRAY (100)
NMARKS =    20
XMAX = "Unset"
XMEAN = "Unset"
XMED = "Unset"
XMIN = "Unset"
XSD = "Unset"
```

```
20 values read as follows:
 8.   56.   77.   67.   59.   66.
 7.   69.   60.   88.   69.   78.
 1.   92.   67.   79.   88.   81.
 3.   73.   74.   60.
```

```
1:39   C:\BOOI.DIR\DEMO\STATS.FOR
```

Ask for a FREE demo disk today!

Windows is a trademark of Microsoft Corporation

Salford *Software*

COMPILER SPECIALISTS SINCE 1978

Adelphi House • Adelphi Street • Salford M3 6EN • United Kingdom
Tel: (+44) 061 834 2454 • Fax: (+44) 061 834 2148 • Facts-by-Fax™: (+44) 061 834 2153

➤ CIRCLE NO. 592

Accessing DBFs

So far, we have only considered DOS/Windows compatibility from the point of view of application development. An equally important issue is that of data files. How easy is it to use your existing DBF data in your Windows applications?

Essentially, all dBASE IV data files can be used directly by dBASE for Windows, without requiring any form of conversion. This applies to DBFs, single- and multiple-index files and memo files. What's more, you can continue sharing the files with dBASE IV users, as the same protocols for file and record locking are employed by both programs.

You can also create files in dBASE for Windows and access them in dBASE IV. dBASE for Windows supports two new data types - binary and OLE - which are not backwardly compatible. But even if your DBFs contain these data types, you can still open and read them in dBASE IV. Of course, you won't be able to work with the binary or OLE data - you'll just see seemingly arbitary numbers in these fields.

In fact, dBASE for Windows offers a good deal of flexibility in accessing data, thanks to Borland's database connectivity system, IDAPI. In a nutshell, IDAPI gives compliant applications a way of getting at data from a variety of sources, including DBF files, Paradox tables and various SQL database servers such as Oracle and InterBase. Within dBASE, you can access all thee data sources using familiar commands like USE and SKIP.

Using IDAPI with dBASE

Although IDAPI is more or less transparent to the dBASE programmer, there are a couple of important points which need to be watched. The first is one of terminology. In line with standard industry usage, dBASE now uses the term 'table' to mean a collection of similar records and 'database' to man a collection of tables. A dBASE table is therefore a DBF file, while a dBASE database is a collection of DBFs *held in the same directory.*

```
USE CUSTOMER ORDER CUSTNUM
TYPE PARADOX
SEEK "A01"
&& go to a specific record
SaveRec = BOOKMARK()
&& save current bookmark
GO TOP
&& go to another record
GO SaveRec
&& go back to saved record
```

Figure 8 - Using a bookmark

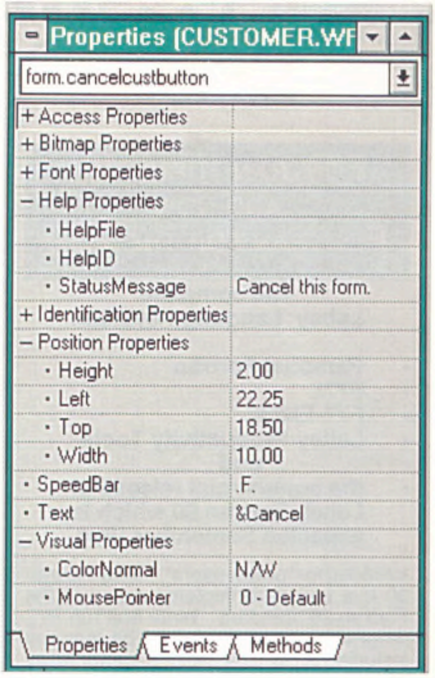
You use the IDAPI configuration utility to define the database. So if all your application's DBFs were held in a directory called `C:\COMPANY\ACCOUNTS\1994`, you would specify that directory as the database. To avoid having to hard-code a long directory path, you can assign a shorter logical alias to it - `ACCOUNTS`, for example. This might sound like a lot of unnecessary hassle but it does have an important benefit. If you subsequently move your data from a local directory to a back-end server, or *vice versa*, you only need to change the definition of the alias.

> **If you are an existing dBASE programmer who has been stuck for too long in a character-based world, now's the time to break free.**

Once you have defined the database, you can issue an `OPEN DATABASE` command to establish a connection to it within your program. You can also use the `SET DATABASE` command to switch to a different (open) database, or you can use the database name to qualify a table name. Thus `USE :ACCOUNTS:CUSTOMER` would open the Customer table within the Accounts databse. Remember, none of this is essential. If you want to go on accessing dBASE files in the traditional way, you can ignore these new commands completely.

Another point to watch for with IDAPI is to do with record pointers - or, rather, the lack of them. The problem is that neither Paradox nor SQL databases supports the concept of record pointers. This does not affect your ability to process records sequentially - you can still use commands like `SKIP` and `SCAN/ENDSCAN` with these tables - but if you try to go to a specific record number, you will see an error message.

To get round this, dBASE for Windows introduces something called a bookmark. You can think of a bookmark as a handle to a record. It's a bit like a record number, except that you cannot examine its value to determine the record's position within the table. You can, however, use it as the operand of a `GO` command. The `Bookmark()` function returns the bookmark of the current recod, which can then be saved in a variable - see Figure 8 for an example.



form.cancelcustbutton

+ Access Properties	
+ Bitmap Properties	
+ Font Properties	
– Help Properties	
· HelpFile	
· HelpID	
· StatusMessage	Cancel this form.
+ Identification Properties	
– Position Properties	
· Height	2.00
· Left	22.25
· Top	18.50
· Width	10.00
· SpeedBar	.F.
· Text	&Cancel
– Visual Properties	
· ColorNormal	N/W
· MousePointer	0 - Default

Figure 7 - Properties can be assigned either by program code or by using the interactive property inspector

A final point about Paradox and SQL tables is that they do not support dBASE's two-stage deletion process. When you delete a record from these tables, it disappears for good. You cannot recall it, and you cannot subsequently pack the table. In fact, `RECALL` and `PACK` both generate errors. This is unfortunate, because it means that you might have to change the logic of your program when moving from DBFs to other data sources (unlike with bookmarks, which work just the same with DBFs as they do with Paradox and SQL data).

And so...

This detail apart, dBASE IV developers should have few difficulties in porting their code to Windows. You will have to get to grips with object-oriented programming if you want to give your apps a true GUI look and feel. But, given the benefits of OOP, it's almost certainly worth doing that anyway. dBASE for Windows also makes it easy to move to a client/server environment, thanks to the almost transparent use of IDAPI. If you are an existing dBASE programmer who has been stuck for too long in a character-based world, now's the time to break free. ■

Mike Lewis is a freelance technical journalist. He writes for computer magazines throughout the world and has written and translated several books on database-related subjects. Contact him on Compuserve (100012,2105). dBASE for Windows costs £349. Upgrades start at £99.

pecode	pscode
pwait	ACCESS()
ASSIST	BEGIN TRANSACTION/
CALL()	END TRANSACTION CALL
COMPLETED()	DEXPORT
DGEN()	ISMARKED ()
LIST/DISPLAY HISTORY	LIST/DISPLAY USERS
LOAD	LOGOUT
PROTECT	RELEASE MODULE
RESET	RESTORE MACROS
ROLLBACK	SAVE MACROS
SET CLOCK	SET COLOR
SET DBTRAP	SET DEBUG
SET ENCRYPTION	SET HELP
SET HISTORY	SET HOURS
SET INSTRUCT	SET PAUSE
SET SCOREBOARD	SET SQL
SET TRAP	USER()

Figure 2 - dBASE IV language elements not supported by dBASE for Windows

apps to Windows in this way, the results look pretty ghastly. The virtual DOS window is a picture of ugliness, not least because of its unpleasant monospaced font. Your program cannot support any of the usual GUI controls, and if it opens its own windows, these appear as simple rectangles rather than as true Windows windows. If you care about the appearance of your apps, you'll want to do something about this.

Unfortunately, that's not as simple as it should be. You cannot merely add a `Font` clause to your `@/Say` commands to introduce proportional fonts, or modify a `Picture` clause to turn a `Get` field into a set of radio buttons. In fact, if you want to create a real GUI application, you have to forget all the time-honoured commands like `@/Say/Get` - these are now flagged as being retained for compatibility only.

Objects all the way

Instead, you have to jump right into the new, object-oriented, way of doing things. The techniques are not that hard to learn, but Borland's approach is nevertheless puzzling. In FoxPro for Windows, you are allowed to continue using commands like `@/Say/Get`, which Microsoft has extended to cater for the graphical interface. Borland originally planned the same scheme for dBASE for Windows, but now seems to have abandoned it in favour of forcing programmers into object-oriented programming. Perhaps that's no bad thing.

In dBASE for Windows, as in other OOP languages, an object is essentially a collection of data items (properties) and procedures (methods). For example, the properties of an on-screen push button might include its position and size; its methods might include the procedure for displaying it on the screen. (Strictly speaking, a property, in dBASE terminology, covers both data and methods; a method is a property to which a procedure has been assigned.)

Properties may be of any data type - character, numeric, logical, whatever. Given that an object is itself a data type, it follows that an object may include other objects. This leads to the idea of containership, which is an important one in dBASE for Windows. You'll often come across it when working with forms. A form object acts as a container for holding other on-screen objects such as edit fields and push buttons.

To make the transition to OOP easier Borland has provided a choice of syntax for use with objects. Existing dBASE programmers can use the **DEFINE** command to create objects, this being analogous to the way that you create windows and menus in dBASE IV. People coming from C++ or Pascal might prefer to opt for the **NEW** operator, using the dot notation to access the properties. The choice is not a hard-and-fast one; you are free to mix the two types of syntax in the same program.

Class definition

To get a flavour of what's involved, take a look at the listings in Figures 3 and 4, which use **DEFINE** and **NEW** respectively. In both

```
DEFINE FORM MyForm FROM 10,20 TO
16,40 ;
   PROPERTY ;
      Text "Sample Form"

DEFINE PUSHBUTTON Search OF MyForm
;
   PROPERTY;
      Text "&Search", ;
      Top  2, ;
      Left 5, ;
      Width 12, ;
      Height 2.5, ;
      FontSize 8


OPEN FORM MyForm
```

Figure 3 - Creating objects with DEFINE

```
MyForm = NEW FORM("Sample Form")
Myform.Top = 10
Myform.Left = 20
MyForm.Height = 7
Myform.Width = 21


Search = NEW PUSHBUTTON(MyForm)
Search.Text = "&Search"
Search.Top = 2
Search.Left = 5
Search.Width = 12
Search.Height = 2.5
Search.FontSize = 8


MyForm.Open()
```

Figure 4 - Creating objects with NEW

```
MyForm = NEW FORM("Sample Form")
&& -- define form properties here
Search = NEW PUSHBUTTON(MyForm)
Search.Text = "&Search"
Search.OnClick = MySearch
&& -- define other button proper-
ties here
MyForm.Open()
&& -- do something with the form


PROCEDURE MySearch
   && -- search routine goes here
RETURN
```

Figure 5 - Assigning an event method

Walking the Web

In this month's Internet column **Paul Richardson** gives a guided tour of that distributed hypermedia info-system known as the World Wide Web...

To put it dramatically, although not very informatively the World Wide Web (*aka* WWW, W3, 'The Web') is the Internet's killer app. To enlighten you a bit more; WWW is an attempt to organise all the world's online information and present it through the use of a single browser utility.

History of the web

WWW was conceived at CERN, the Swiss centre for particle physics research, by Tim Berners-Lee back in 1989. He perceived a need in the field of particle physics for a wide-area hypertext system that would assist the widely-spread community of scientists to keep track of research papers, both present and past. In addition, he was keen to see the information presented in a uniform manner, so that scientists would not need to learn how to access many disparate systems.

A WWW system went into full operation at CERN in 1991. In the same year the server and browser software were made available to the Internet community at large. From this point the growth of WWW has been exponential, as servers popped-up all over the world and browsers for various platforms were written.

An important catalyst to the popularity of WWW has been the involvement of the National Centre for Supercomputer Applications (NCSA), an American institute with a good track record of producing quality Internet applications. NCSA has written servers and browsers for several platforms which are freely accessible to anyone with access to the Internet.

URL	Description
http://www.sun.com/	Sun Microsystems' product details and World Cup information.
http://hypatia.gsfc.nasa.gov/NASA_homepage.html	NASA's home page.
http://apollo.co.uk	Commercial advertising service.
http://www.elpress.com/	Commercial WWW server - space for rent.
http://www.sgi.com/	Silicon Graphics' web server - great graphics.
http://mistral.enst.fr/~pioch/louvre/	Pictures from the Louvre Museum and a tour of Paris.
http://www.cityscape.co.uk/	CityScape products and services. Also Global On-Line Directory.
http://www.ucs.ed.ac.uk/General/uk.html	Clickable map of UK WWW sites.
http://www.bbcnc.org.uk/	BBC Networking Club.
http://debra.dgbt.doc.ca/opengov/	Canadian government server.
http://search.bibsys.no:5001/zpub.html	Norwegian library service.
http://www.cyberspace.org/u/ecafe/www/index.html	Electronic Cafe.

Figure 1 - Interesting Resource Locators

The past year has seen two major developments in the WWW field. First, the NCSA has licensed the source code for its browsers and servers to several companies. And second, the majority of the NCSA development team left to form a company which plans to offer commercial grade WWW software. The NCSA is putting a new development team in place. It is stressing that it will continue to support and develop its applications.

Just browsing, thank you

An important feature of most WWW browsers is that they 'know' how to present various types of information to the user, possibly through the use of an external utility such as a sound tool or GIF viewer. It is this feature that makes WWW a hyper*media* system rather than just a hyper*text* system and reduces the chasm between the much-touted capabilities of the Super Data Highway and current reality.

Hopping around the Web when using a graphical browser is generally merely a matter of clicking on the hotspots. It's the most effortless way of performing intercontinental hops ever devised this side of 'the final frontier'. I do, unfortunately, have to report one form of 'travel sickness' though. I seem to be developing an imaginary case of Repetitive Strain Injury in my right fore finger.

Mosaic is preconfigured with a list of pages to choose from that form a good basis for exploration. Figure 1 contains my own suggestions. These are an eclectic mix that attempts to illustrate the diversity of the Web's application.

The introduction of Universal Resource Locators (URL) scheme is an important step forward for the Internet as a whole since it provides the means for referring to any document or service on the Internet. The syntax of a URL is:

```
protocol://host[:port]/pathname
```

or

```
news:newsgroup
```

You will notice that many of the URLs in Figure 1 simply end with a trailing backslash, this indicates to the browser that the

home page in this directory should be loaded, ie `home.html`.

Browsers are readily available for a number of platforms. Figure 2 contains details of the most popular ones, including how to obtain them.

When a WWW session is initiated, the browser will generally load a 'home page'. Mosaic comes preconfigured to load the NCSA home page, although through judicious fiddling with the .INI file it can be configured to load any page or none at all.

For those with only a dial-up connection it is a good idea to turn off the display of inline images which causes a simple logo to be displayed instead. It's a difficult compromise to make, as it significantly reduces the Web's sex appeal. But the frustration of waiting for those little images to load is similarly unpalatable.

Custom browse

WWW browsers rely heavily on external 'viewers' for displaying graphics, playing audio, etc. At the very least, you should have a utility capable of displaying .GIF files, such as LVIEW.EXE for MS Windows or **xv** for X. The mapping of file type to viewer application is completely configurable, so you can make use of whichever applications you already have. The mapping is performed in two stages, first the file extension is mapped to a 'type' and then the 'type' is mapped to a viewer. As an example, here is a snippet from my Mosaic for Windows .INI file:

```
[Viewers]
TYPE0="audio/wav"
.
audio/wav="mplayer %ls"

Suffixes]
.
audio/wav=.wave,.wav,.WAV
```

Platform			
Package	FTP Site	Directory	Comments
Microsoft Windows			
Cello Mosaic for Windows	ftp.law.cornell.eduf ftp.ncsa.uiuc.edu	/pub/LII/cello/ /PC/Mosaic/	 Very popular.
MSDOS			
DosLynx	ftp2.cc.ukans.edu	/pub/WWW/DosLynx/	
Macintosh browsers			
Mosaic for Macintosh	ftp.ncsa.uiuc.edu	/Mac/Mosaic/	
Amiga			
AMosaic	max.physics.sunysb.edu	/pub/amosaic/	
XWindows			
NCSA Mosaic for X tkWWW Browser/Editor for X11 Viola for X Chimera	ftp.ncsa.uiuc.edu harbor.ecn.purdue.edu ora.com ftp.cs.unlv.edu	/Mosaic/ /tkwww/ /pub/www/viola/ /pub/chimera/	Highly regarded. Supports WSYIWYG HTML editing. For Motif or Xlib. Handles HTML+ forms and tables. Uses Athena. Supports forms and inline images.
Text-mode Unix			
Line Mode Browser Lynx full screen browser perlWWW Emacs w3-mode	info.cern.ch ftp2.cc.ukans.edu archive.cis.ohio-state.edu moose.cs.indiana.edu	/pub/www/src/ /pub/WWW/lynx/ /pub/w3browser pub/elisp/w3	 Written in Perl. Extension to Emacs.

Figure 2 - World Wide Web browsers

HyperText Markup Language

HTML is an SGML DTD! Whoa, let me just run the Macro Preprocessor on that...Whirr...HyperText Markup Language is a Standard Generalised Markup Language Document Type Definition. To put it another way, HTML is a specific implementation of SGML. An HTML document is plain ASCII text marked-up using tags. The hypermedia links in HTML documents can point to other parts of the same document, a different document at the same site or a document at a different site, thus paving the way for an information system that spans the globe.

'But, what about all the information that isn't written in HTML,' I hear you ask. Well, as already mentioned URLs can point to any type of resource. Through the use of the browser's other built-in protocols and HTML gateways, they can all be viewed or retrieved by a WWW browser.

Platform			
Package	FTP Site	Directory	Comments
Microsoft Windows			
NCSA httpd SerWeb	ftp.ncsa.uiuc.edu winftp.cica.indiana.edu	/Web/ncsa_httpd/contrib/ /pub/pc/win3/winsock/	Has most of the features of the Unix version. Basic.
Microsoft Windows NT			
HTTPS	emwac.ed.ac.uk	/pub/https/	For Intel and Alpha.
Macintosh browsers			
MacHTTP	www.uth.tmc.edu	/mac_info/	
Unix			
NCSA httpd CERN httpd GN Gopher/HTTP server Plexus	ftp.ncsa.uiuc.edu info.cern.ch hopf.math.nwu.edu bsdi.com	/pub/web/ /pub/www/bin/ /pub/gn/ /server/doc/	 Serves both WWW and Gopher. No script capabilities. Written in Perl.

Figure 3 - World Wide Web servers

The most important thing to remember when rolling your own pages is that the browser ignores whitespace and performs word wrap when it thinks it is necessary. The only way to force the browser to abide by your formatting is to use the **`<PRE>...</PRE>`** tag. This might be necessary, say, if you wanted to include a C fragment.

The precise graphical style associated with each tag is not defined and hence two different WWW browsers might present the same HTML document differently. Indeed, Mosaic allows, the user to choose the font associated with several of the tags.

Of course the real power of HTML is not the simplicity with which plain text can be formatted, but its ability to specify hypermedia links. Even the inline images can be linked. So, for instance, a set of thumbnail previews could have links to the full-size graphic files.

Tools that help you HTML

Writing HTML by hand is not much fun. Fortunately there are some tools available. These tools can be broadly classified into editors and convertors. Two editors that are in common usage are HTMLASST.EXE, an MS Windows program and CU_HTML.DOT, a set of macros for Microsoft Word. Emacs fans also do well with several extension packages for producing HTML available.

Convertors are important because of the wealth of non-HTML documents already on the Internet. Convertors have been written for a wide range of document formats, such as FAQs, RFCs, RTF , man pages, postscript, Word, WordPerfect, FrameMaker, troff, LaTeX, C, C++, Fortran, LISP... For

C, C++ and BASIC programmers, now you get much more than xBase compatible DBMS power.

Thousands of programmers have already discovered how to get dBASE, FoxPro and Clipper compatibility with their favorite language and hardware platforms. For example, one customer has C programs running on PC and Sun workstations sharing data with concurrently running FoxPro for Windows applications.

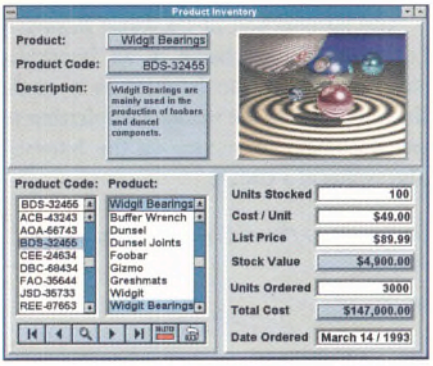
You see, CodeBase technology is simply the best way to add multi-user xBase compatible DBMS power to C, C++, Basic or Pascal.

You still gain speed & small size

CodeBase users really appreciate our small executable size. Unlike SQL engines which are a Meg or so in size, CodeBase 5.1 EXE's can be as small as 45K! You'll also like the speed–with our Intelligent Queries you get the execution speed of C plus stunning query performance from our smart use of available index information.

Now formatted data entry in Windows is as easy as point & click!

Experienced Windows programmers know **formatted** data entry is difficult to program under Windows. But with our new CodeControls, you can simply 'Point & Click' to design data entry windows for date, numeric, and character information–formatted just the way you want it.



Product: Widgit Bearings

Product Code: BDS-32455

Description: Widgit Bearings are mainly used in the production of foobars and duncel componets.

Product Code:	Product:
BDS-32456	Widgit Bearings
ACB-43243	Buffer Wrench
AOA-66743	Dunsel
BDS-32455	Dunsel Joints
CEE-24634	Foobar
DBC-68434	Gizmo
FAO-35644	Greshmats
JSD-35733	Widgit
REE-87653	Widgit Bearings

Units Stocked 100

Cost / Unit $49.00

List Price $89.99

Stock Value $4,900.00

Units Ordered 3000

Total Cost $147,000.00

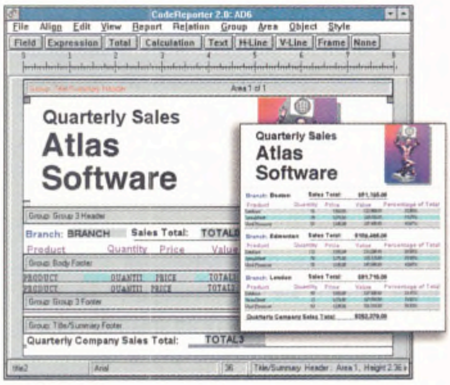
Date Ordered March 14 / 1993

***Introducing the new* CodeControls**, a unique set of data-aware custom controls. Now simply drop them into your Windows applications via your favorite visual interface builder.

NEW–Data-aware controls

Our new custom controls are ***data-aware,*** so now you can easily build a scrolling list box that's tied to a data file, or look up matching combo box entries–even as the user types.

Introducing CodeReporter 2.0



Quarterly Sales
Atlas
Software

Branch: BRANCH Sales Total: TOTALS

Product Quantity Price Value

PRODUCT QUANTIT PRICE TOTALS

PRODUCT QUANTIT PRICE TOTALS

Quarterly Company Sales Total: TOTAL3

***Introducing the new* CodeReporter 2.0** our visual, interactive xBase report writer. We designed it with developers in mind, but end-users will love it.

Create a wide variety of reports –visually, easily, and instantly.

Use ***CodeReporter's*** new *Instant Report Wizard* to create a report–in an instant. To refine your report, simply drag and drop report objects–for data, totals, text or graphics–using the interactive layout screen.

Easily build report queries using our calculator-style expression builder. Then get your reports lightning fast through the built-in Intelligent Query Technology.

And get multi-platform portability.

Once your reports are designed under Windows, you can generate corresponding source code. Use this source code to launch reports under DOS, Macintosh, Windows, NT, OS/2 or UNIX.

Introducing CodeTranslator 3.0

Now you can automatically translate Clipper, dBASE, and FoxPro code into C++. Turbo-charge critical xBase applications, port to new operating systems, and gain the flexibility of C++.

CodeTranslator keeps your variable names and uses the CodeBase++ library–making the translated code easier for you to read and maintain.

Buy One, Get Two FREE.

Now when you buy any one of our xBase compatible library products: ***CodeBase, CodeBase++, CodeBasic*** or ***CodePascal*** (for the language of your choice), you'll get both the new ***CodeReporter 2.0*** *AND* the new ***CodeControls 2.0*** absolutely **FREE –for a limited time only.**

TO ORDER NOW
CALL 071-833 1022

System Science
Your Sequiter Authorised Dealer

SEQUITER
SOFTWARE INC.

FAX 403•436•2999
UK Tel. +44-81-317-4321
P.O. Box 575 Newmarket NH 03857-0575

➤ CIRCLE NO. 595

Letters

We welcome short letters on any subject that is relevant to software development. Please write to: The Editor, EXE Magazine, St Giles House, 50 Poland Street, London W1V 4AX. Unless your letter is marked 'Not for Publication', it will be considered for inclusion in this section.

NT for the masses

I liked the style of your article on setting up NT Advanced Server. The step-by-step report on the event was just what was needed for people contemplating a similar exercise. It gave a global overview of what is required in order to decide in advance: naming machines and domains, the areas where care may be needed to stay within the compatibility envelope. However, I would like to raise a couple of points that installations like yours, where there are 20 users or less, may not encounter.

An unwelcome trend with GUI-based systems is to make administration a manual-intensive exercise: setting up user names and accounts, for example, may involve much clicking on buttons and typing into text-boxes. When there are 2,000 or more users we may look for automated script based systems. Unfortunately, this may often be in vain, because it is now 'user-friendly' !

Alex Telford
Human Genome Mapping Project Resource Centre, Harrow

Oh yeah?

In response to H G Martin's *C and C++ Stink*, it is all too easy to blame the choice of language for poor and often buggy programs. There is also very little relation between quality and world wide acceptance. Gifted programmers can use whatever is available to turn out world beating products. Gifted marketing experts can make world beaters out of dross.

Those familiar with the work of Donald Knuth will know that he originally chose (generic) Pascal for writing TEX and MetaFont because he wanted portability above all else and was willing to tolerate severe deprivation, use of hacks and several global goto's in order to achieve this end. It should also be noted that his recent work has been in ISO C. I doubt that anyone who has used his recent material would accept that this change of language is a sign of developing senility.

Too often the problem faced by programmers is a fixation on a particular language by managers, supervisors and the like. I know of a university Department of Engineering that had to set up its own course in C to support the needs of students working on embedded systems. The university's Computer Science Department categorically refused to teach C, despite its generally acknowledged suitability for embedded system programming.

What we need (along with America) is proper training, leading to practitioners with a sound understanding of the software development process. We need managers who understand that the choice of a compiler is probably the least important decision in the process. Good development and maintenance tools are essential. Unfortunately too many companies in the UK consider providing anything more than a compiler and editor as a luxury their programmers can do without.

Francis Glassborow
Chairman, ACCU

Murder, He Wrote

On 18th May 1994, I booted my computer to find a display of flashing boxes, the sound of various musical notes and a message advising me that my PC had been subject to a 'Morphic Virus Invasion'.

As a Forensic Accountant and Computer Consultant, I was aware that the above message must have been written by someone who was aware of the new strain of Computer Viruses know as 'Polymorphous Viruses'. These are relatively difficult to detect by most Anti-Virus software, as they change their appearance when activated or moved. My attack must have been written to shareware or some recent programme as the Filespec of the respective Batch Files was dated in early May. Two files were timed at 13:01 on a Saturday, and one at 20:38 on the Sunday. Those were the days on which the CRIME was committed.

My course of action was to Boot from a Clean floppy, use a Utility to discover which Files were involved and move them to a floppy. I then printed out the Files and was able, on a further occasion, to rewrite the Batch Files from the Printouts to create a 'simulation' of this Virus which the originator has, as I discovered, named Killjoy.

I am not aware of the origination of the virus, but I believe that it must have been put on more than one floppy, and also that it will surface during the next few weeks on other Computers. I believe that Killjoy may be included on a Shareware Games Diskette in a compressed form. When Decompressed and activated on a Trigger it will 'murder' the Target Computer. It may also be a 'Trojan'. Anyone who introduces this Virus to a Computer is also guilty under the Computer Misuse Act, and could be an accessory to the Crime.

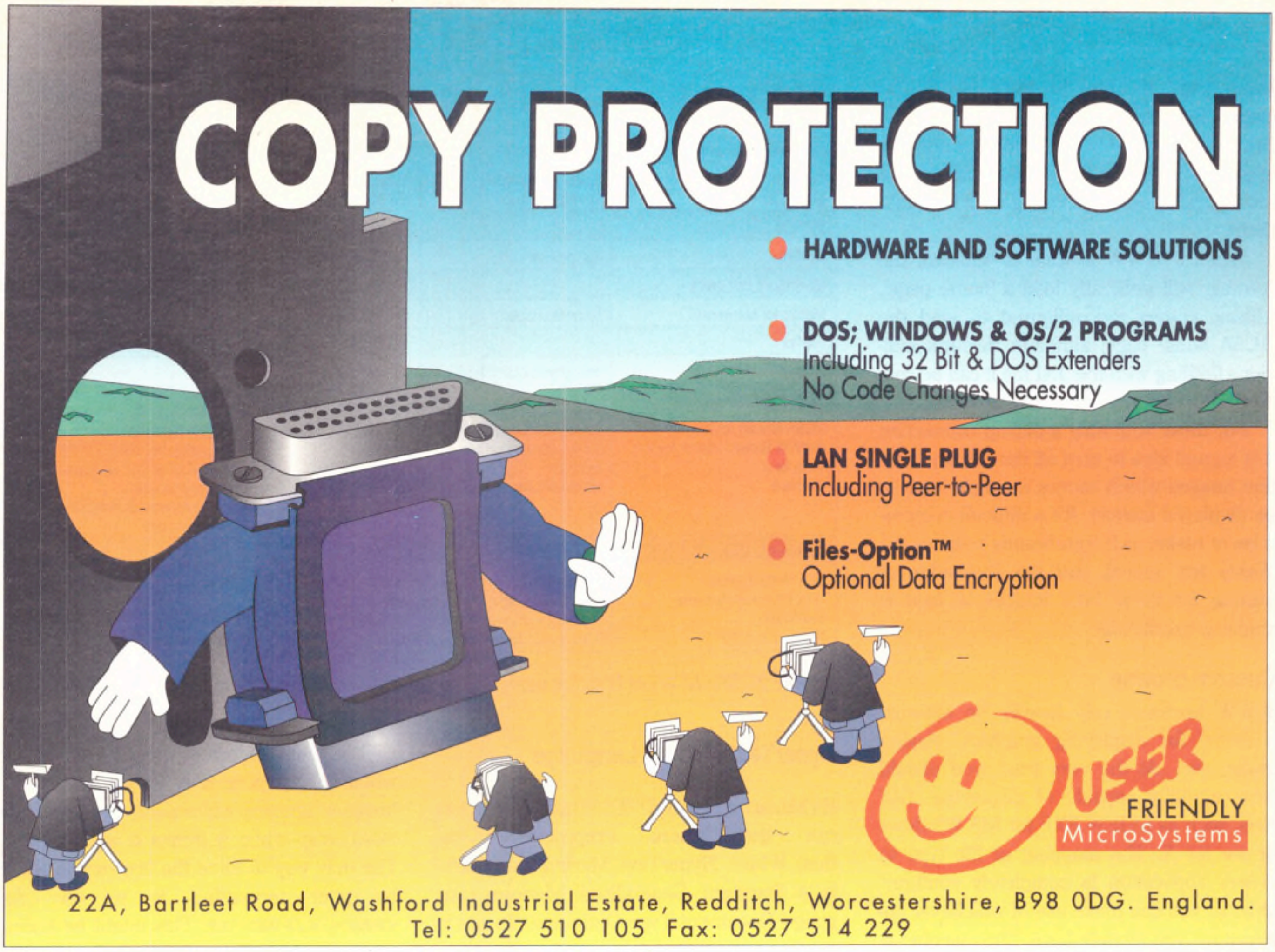
The floppy to which I moved the various files was destroyed soon after I had printed out the contents of the Files. I believe, but cannot prove, that the Diskette was subject to some type of Formatting. The attack copies another File to the Autoexec.bat File on the Root Directory and therefore prevents access to the Computer. I also believe, but am unable to prove that it may Format the Hard Disk.

I immediately contacted the Technical departments of Reflex Magnetics, S&S International and Sophos, as well as the Computer Crime Department of New Scotland Yard, and sent them Faxes of the Printouts and Filespecs. I believe that the general public should be aware of this strain and take all precautions before it is too late.

Jonathon Goldman
FAPA, CEUE

Letter of the Month

The writer of the best letter of the month, as judged by the Editor, will receive a £30 book voucher, courtesy of PC Bookshop, 21 Sicilian Avenue, London WC1A 2QH (071 8310022). The best letter is the one printed first. Please note that letters submitted to this page may be edited.



COPY PROTECTION

- **HARDWARE AND SOFTWARE SOLUTIONS**
- **DOS; WINDOWS & OS/2 PROGRAMS**
  Including 32 Bit & DOS Extenders
  No Code Changes Necessary
- **LAN SINGLE PLUG**
  Including Peer-to-Peer
- **Files-Option™**
  Optional Data Encryption

USER FRIENDLY MicroSystems

22A, Bartleet Road, Washford Industrial Estate, Redditch, Worcestershire, B98 0DG. England.
Tel: 0527 510 105 Fax: 0527 514 229

➤ CIRCLE NO. 596

MICROLAND BULLETIN BOARD SYSTEM

0891 990 505
To Download

0483 725 905
To View

Shareware to download at V32bis, V32, V22bis, V22, V23 & HST etc, 8 data bits, no parity.

INTERNET CONNECTED

Microland is now connected to the world's largest network. We now provide full global mail, news, software archive, and interactive services. Don't miss out, get connected and explore the world of Internet. Download from the biggest archives in the world. Services subject to availability.

Why wait for your software when you can download now direct to your computer. Microland offers high speed connections and most files are compressed to minimise cost. Check it out on 0483 725 905 at normal call charges. So much shareware you are spoilt for choice.

Calls to 0891 990 505 charged at 36p per minute
cheap rate, 48p per minute all other times.
Trevan Designs Ltd, PO Box 13, Aldershot, Hants. GU12 6YX.

➤ CIRCLE NO. 597

SOFTWARE PACKAGING

- Printed boxes and sleeves in full colour to your specification
- Shaped polystyrene packers
- Ring binders/slip cases plain or printed
- Postal cartons and labels
- Disk pockets. Disk labels. Disk duplicating
- Printed copyright envelopes
- Shrink wrapping
- Assembly service

MANUAL PRINTING

- New manuals or reprints. All sizes
- Short run copying service
- Wire-o-binding, tabbing
- Data conversion
- Full artwork/origination service

RAPID TELEQUOTE SERVICE

RIDGEWAY PRESS
TEL: 0734 845331 FAX: 0734 845186

➤ CIRCLE NO. 598

Book Review

Programming for Graphics Files in C and C++ Reviewed by **Edward Kenworthy**

For me, graphics programming has always been a bit of black art. The details of GIF and JPEG and TIFF formats belong to one of the more mysterious and, as yet, unpenetrated branches of the programmer's trade. My previous attempts to clear the mystery surrounding graphics had failed miserably: foiled by texts that either went into the explicit and disgusting detail of how to manipulate the registers on your PC's VGA card, or proved to be hollow sounding shells with little substance beneath the glossy exterior.

John Levine's *Programming for Graphics Files in C and C++* concentrates entirely on the representation of graphics images in files. I find this far more interesting than grubbing around in your PCs VGA card. Especially in these days of sanitised graphics programming provided by MS-Windows and similar GUIs. Levine begins with an overview of what graphics files are and a general description of the different types of format: bitmaps, vectors, metafiles and printer data files. He also covers in principle the conversion methods between each of these formats.

The bulk of the book consists of six sections covering the different graphics file formats and, with the exception of the JPEG format, complete libraries for reading and writing all the bitmap formats described. So the companion disk is probably worth ordering ! One of the most interesting libraries has been written by Jef Poskanzer and named PBM. This is used as a framework for bitmap file processing, which is possible because all bitmap files have a similar structure. Once the core routines have been covered, Levine moves onto specific libraries for monochrome, grayscale and full-colour bitmapped images. Specific bitmap formats are then described, with each of the sections being divided according to compression method: run length (MacPaint and PCX), uncompressed (MS Windows Bitmap and Targa) and dictionary compressed (GIF, TIFF and JPEG).

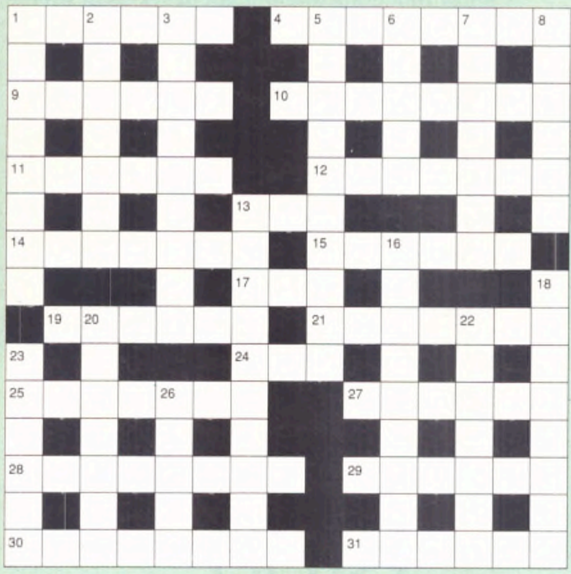
The final section of the book covers conversions between the different graphics file formats: bitmap to bitmap using the PBM library, and vector to bitmap by rasterization. Reliable bitmap to vector conversion, as Levine states at the beginning of the book, is beyond current techniques.

So, what did I think of this book ? I was interested more in the general aspects of graphics file formats: how bitmap files are structured, how 'glossy' compression methods like JPEG worked and similar details: Levine's book covered all this. That it also provided libraries to read, write and convert the graphics files I was most likely to have to work with (bitmaps), was a bonus. On this basis, for me the book was a hit. Overall I found it interesting and well written, with a good depth and breadth of coverage within the topic area.

Verdict: Recommended.

Title:	*Programming for Graphics Files in C and C++*
Pages:	*494*
Price:	*£43.95 with disk*
Author:	*John Levine*
Publisher:	*John Wiley and Sons, Inc*
ISBN:	*0-471-59854-2*

PRIZE CROSSWORD



ACROSS

1 Pity about a king - it needs checking (6)

4 Getting together the final code (8)

9 Input device under the prof (6)

10 Big bird (newly dead) somehow starts & copies (8)

11 Hang about! Work about with hesitation (6)

12 Crystalline array (7)

13 US TLA source of hidden world facts on cd? (3)

14 The program language of the future (7)

15 Basic units of down-market OOL? (6)

17 Moi? A French friend?

19 Whence the graph should sprout with no equipment inside (6)

21 One that goes on top in a multi-layer system (7)

24 Bangladeshi bread from Grandma (3)

25 Not on the top level of objects (3-4)

27 Combust in the design I tendered for (6)

28 Castles for adventurous programmers? (8)

29 Tall fringed dog from Araby (6)

30 Get out key with a lad from the East and get in using a ladder (8)

31 Say it and smile! (6)

DOWN

1 Type of processing for lines that remain the same (8)

2 Task of 9 in England's Silicon Valley (7)

3 About to fall with golf accessory to encircle I hear (9)

5 Causing great excitement without the start of a model (10)

6 Great show in the European Community (latin style) (5)

7 Laying odds on improvement without sign of hesitaiton (7)

8 Chorus from those who always agree (6)

13 Sent along the signal's route (10)

16 Old icon (9)

18 As great as eight bits - a nice mouthful (8)

20 Unemotional aspect of automation (7)

22 Garland certain to improve play-time (7)

23 Mind you take it from a tipsy chemist! (6)

26 Sing à la Suisse (5)

SOLUTION TO JUNE'S CROSSWORD: ACROSS: 1. COMPUTERISATION 9. NOISE 10. OPERATING 11. OBSESSIVE 12. DIGIT 13. EROSION 1 5. GENESIS 17. THERMAL 19 FIDGITS 21. FIELD 23. NORMALISE 25. EXTRACTOR 26. GUIDE 27. SYSTEMS ANALYSIS **DOWN:** 1. CONSOLE 2. MAIN STORE 3. USERS 4. EMOTION 5. ICEBERG 6. ABANDONED 7. ICING 8. NEGATES 14. IMMEDIATE 16. SPECIFIES 17. TOFFEES

'PRIZE CROSSWORD' is compiled by Eric Deeson

July's prize crossword is sponsored by Borland which will be supplying a copy of the new Sidekick for Windows. May's winner was GV Rendell. Please send your solutions to
EXE Magazine, St Giles House,
50 Poland Street, London W1.4AX

more details on these convertors see URL `http://info.cern.ch/hypertext/WWW/Tools/Overview.html`

Through an interface called the Common Gateway Interface (CGI), HTML can be extended to initiate a variety of actions, such as accepting user input and sending email. In fact, you can kick-off scripts to do just about anything.

There are many calls to extend the capabilities of HTML and a draft specification for HTML+ has been produced. HTML+ is a superset of HTML and supports features such as tables, right justified text and centred text. The URL for the draft HTML+ specification is `ftp://ds.internic.net/internet-drafts/draft-raggett-www-html-00.txt`

Serving up some hypermedia

The phenomenal growth of the Web is in part attributable to the easy availability of the server executables for various platforms. Many organisations with information to distribute are benefiting from setting up servers; clubs, academics institutions, charities, government agencies, companies.

There are a number of steps involved in setting up your own server. First, you obviously need to get connected to the Internet. Then obtain the appropriate server for your platform. Next, scribe a few HTML pages (preferably sprinkled with inline images) and fire-up the server. Finally announce your presence by a post in `comp.infosystems.www`. Okay, so I trivialised the matter somewhat, but I'm afraid the more gory details will have to wait for another article. If you're feeling adventurous and want to wade in there, I've put a list of servers in Figure 3.

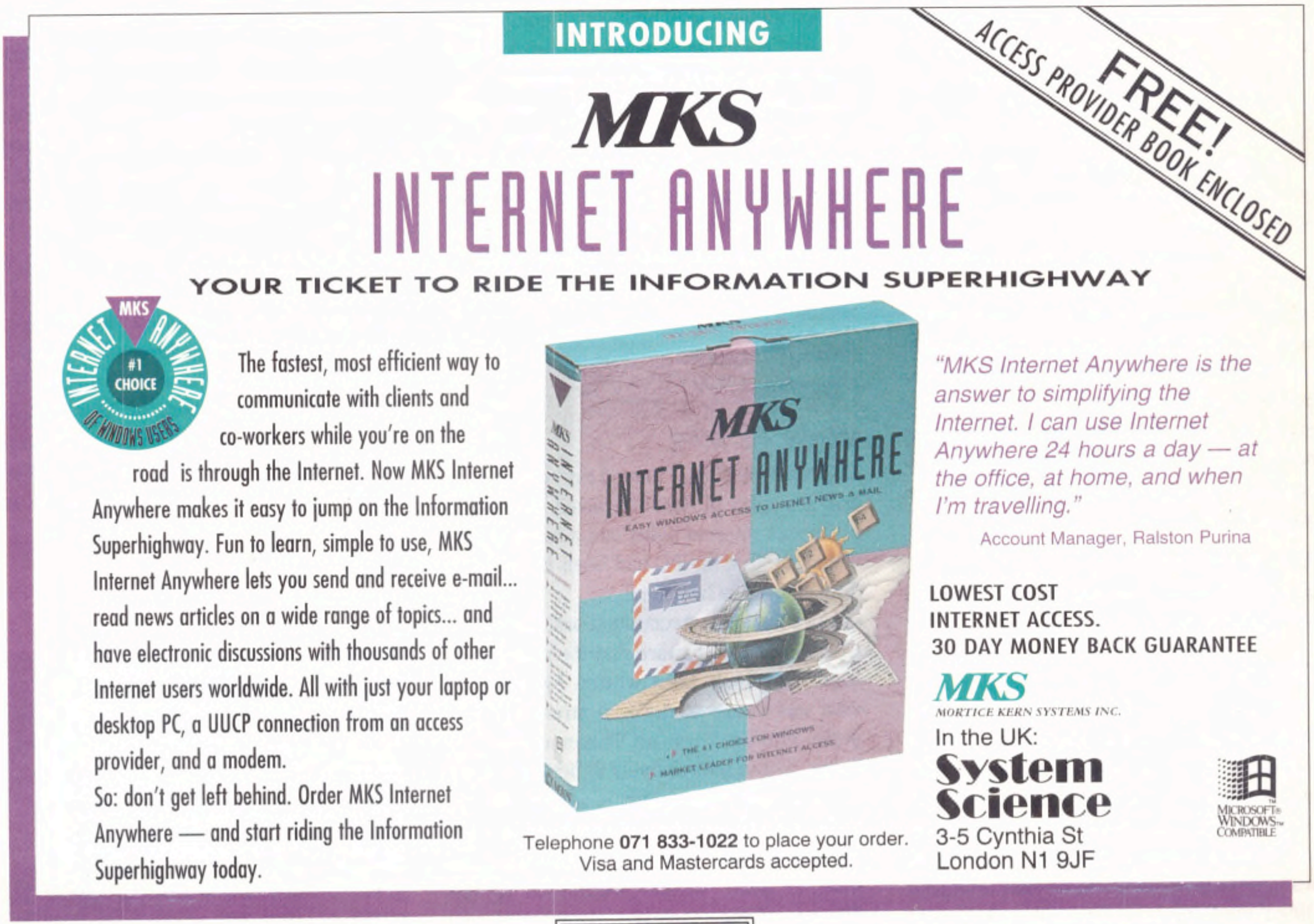
> **It's the most effortless way of performing intercontinental hops ever devised this side of 'the final frontier'**

What the web can do for you

Among the more creative uses are examples of shopping malls, estate agents, classified advertising, art exhibitions and guided tours. An entirely different range of applications present themselves if an internal WWW server is set up and is accessible only to the users of the private network. For instance you could create personnel directory services, floor plans, online forms such as holiday applications or suggestions, organisation charts and company-wide memos.

The ease and speed with which such information can be disseminated using a private Web, in conjunction with a corporate email system goes a long way towards achieving the flatter, more democratic structure which many companies are attempting to put in place. The Web is such a malleable and potentially pervasive media that it represents a great challenge to entrepreneurs and marketing executives. Applications are really only limited by the imagination. ■

Paul Richardson is a Director of Motiv Systems Ltd, a consultancy specialising in Open Systems, interoperability and the Internet. He can be contacted on 0223 576318 or by email at `PaulR@Motiv.demon.co.uk`.



FIRECRUITMENT

AP/DH

SOUTH COAST PROJECT
IBM CONTRACTORS NEEDED

Analyst/Programmers	-	COBOL, CICS, DB2
Analyst/Designer	-	COBOL, CICS, DB2
Analyst/ Programmers	-	COBOL, CICS, IDMS-R
Analyst Designer	-	COBOL, CICS, IDMS-R
Analyst/Designer	-	AFP, EDI
Comms Special	-	RS6000, AIX

TH/JW

NORTH EAST
SUPPORT & MAINTENANCE PROJECT
ICL, VME, DDS with COBOL and SCL

Systems Analyst • Team Leaders • Project Leaders • Technical Managers • Senior Systems Analysts • • Development Manager •

PS **COBOL, CICS, DB2 or TELON - Analyst Programmer**

ORACLE Version 7 - Analyst Programmer

SS **PAXUS, POLISY - Analyst Programmer**

JM **SUPRA/MANTIS - DBA with DCL**

MLT **DOS/Windows Conversion - C and C++**

To find out more, contact Vincent Banks:
Tel: 081-943 2123 Fax: 081-943 5159

F.I. RECRUITMENT, BridgeHouse, Broad Street,
Teddington, Middlesex TW11 8QT

C / C++ SDK and SQL an advantage	**to £25k + car** London
C++ Visual Basic, ODBC and SQL an advantage	**to £25k** M4 Corridor
PROJECT MANAGERS Large scale Open Systems implementation	**to £45k + car** London
ORACLE V6/7. Forms 3/4. CASE desirbale.	**to £28k + car** Berks
POWERHOUSE Any platform	**Package to £32k** City
GEMBASE All levels.	**to £25k** Hants
SYNON & RPG All levels.	**to £30k** Berks
LOTUS NOTES Applications Development.	**£neg + car** Berks

We receive new requirements every day (both contract and permanent) requiring PC Development skills (Windows, Access, Visual Basic, C++, etc) and RDBMS experience (Oracle, Informix, Ingres, Sybase, etc).

If you are interested in any of the above positions, or if you are looking to move soon, then please send your CV to Gareth or Viv at the following address:

Clarke Davies Limited

The Cooper's House, Brewery Court, Theale,
Reading, Berks RG7 5AJ

Tel: 0734 302255 Fax: 0734 302444

the soft corporation

Specialists in Software Development Staff Recruitment

SOFTWARE ENGINEERS - SENIOR SOFTWARE ENGINEERS

Various Client/End Users, Software Vendors and Software Houses dedicated to strategic implementation of leading edge technology and integration of applications across different hardware and operating systems platforms require candidates to degree level with a scientific / technical development bias and 1-3 years' experience. There are two main options:
TECHNICAL DEVELOPMENT: Continued use of UNIX, VMS, MS-DOS, C, Windows (SDK, NT or X-Windows and Toolkits), Networking and Communications with companies offering technology based careers and management responsibility.
COMMERCIAL DEVELOPMENT: Using technical based skills already developed, but offering opportunities to apply analysis and design skills rather than remain "a technical guru" in various environments including finance. Please call to discuss your particular career, growth and potential.

£12,000 - £25,000 + benefits. REF SC/01/EXE

LONDON / HOME COUNTIES WINDOWS SDK / NT DEVELOPMENTS

Senior Development Engineers **Analyst/Programmers**
To £30,000 + benefits **To £27,000 + benefits**

Strong programming skills in C or C++ and Windows 3 are prerequisites for these positions. Experience in some of the following areas is also required: MS - DOS 5.0; MS Windows 3.1; Windows SDK; MS C 7.0; Visual Basic 2.0, Visual C++ and Microsoft NT. Also desirable Windows VxD or networking skills.

REF SC/02/EXE

UNIX / VMS / MS WINDOWS ALL LEVELS

A degree in computer or natural science, 2 years solid C programming experience and a sound understanding of UNIX, VMS or MS-DOS are required to work on large scale programs with user interaction. You will need an intelligent problem solving approach to work and be a quick learner to programme software in an X-Windows, Windows SDK or NT environment, port software to different systems and liaise with customers to drive through product improvements. Excellent career opportunities for the right candidates.

£16 - £28,000 REF SC/03/EXE

C / C++ / VISUAL BASIC / UNIX / MS-DOS DEVELOPERS

Software House and End Users in Finance, Banking, Manufacturing, Commercial, Scientific and Government application environments require excellent C skills. Both Windows development skills W/3, SDK, NT, X - Windows and Visual Basic or strong C, C++ solid operating systems and good application knowledge are again much in demand. Software development experience is the key, and being able to deliver high performance, high quality, well specified software in competitive time scales. Opportunities vary from small to large software companies involved in expert systems, GUIs, Image Processing, GIS, EIS, Communications, Networking and Object Orientated Databases. Graduates through to senior software engineers / team leaders are required. Please call to discuss.

£14,000 - £35,000 REF SC/04/EXE

INGRES / ORACLE / SYBASE ALL LEVELS

Additional experience of: SQL, Forms, C and other 3GL languages are also of interest. We currently have client companies including Management Consultancies, Systems Houses, Systems Vendors, Bank and Finance clients looking for candidates with: Relational Database Design, Database tuning, Systems Administration, DBAs, Pre/Post Sales and solid programming knowledge and expertise. Please call to discuss your particular requirements.

£18,000 - £40,000 + benefits REF SC/05/EXE

INGRES / VMS / C 1 - 3 YEARS £25 - 30K + Bonus

Excellent opportunities for Graduates with 1 - 3 years' experience to join a Banking Organisation. Training will be given in the Derivatives / Financial Instruments market. Motivated self - starters who want to take on responsibility in a progressive organisation where skills are rewarded on merit. **OTHER PLATFORMS/DATABASE CONSIDERED.**

C/06/EXE

UNIX / MS - DOS / VMS / C AND C++ ANALYST PROGRAMMER

Excellent opportunities exist for bright graduates with one year + experience. Personal background requires a solid understanding of the project life cycle and a commitment to high quality coding. you will be trained in all aspects of Investment Banking, relational databases, 4GL's and Object Orientated Design. A good opportunity for a second career move.

£17 - £25K + Banking Benefits REF SC/07/EXE

WINDOWS 3.1/X-WINDOWS BANKING ALL LEVELS

3 city clients require windows skills at any level. Other relevant skills are SQL server, Transact SQL, UNIX, VMS or MS-DOS, C, C++, Open Client (DB and Net library) Open interface and APT. Exposure to analysis, developing user interfaces and rapid development techniques. Full training in Middle Office / Production and Front Office Systems including: Financial and Management Accounting, Treasury, Equity, Fixed Income and Derivatives.

£20 - £25K + Banking Benefits REF SC/08/EXE

OOD / OOP, C, C++, VISUAL C++ ALL LEVELS

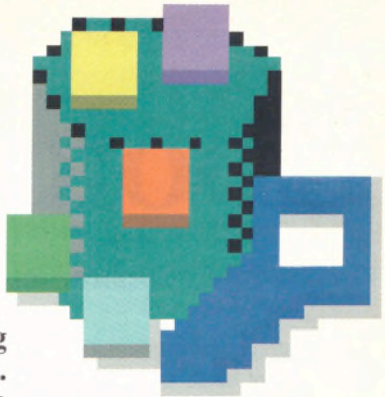
As the market for Object Orientated \skills gathers pace we have a number of clients designing systems in diverse application areas including: **Multi-media, DTP, Telephary, LANs, Electronic publishing, On-line Information Feeds, Finance and Banking** in both a UNIX and DOS environment.
Positions available very from traditional Programmer / Software Engineer and Analyst / Programmers to Designers / Senior Software Engineers in the overall strategic direction for end-user organisations

£17 - £35,000 + Benefits REF SC/09/EXE

VISUAL BASIC SKILLS MUCH IN DEMAND - PLEASE CALL TO DISCUSS

S **the soft corporation**
10 Pakeman Street, London N7 6QN
Tel: 071-609 5501 Fax: 071-700 5787

INTEGRA VDB™



The Visual Database Builder for Microsoft Visual Basic, Visual C++ and Borland C++

Developers will find that they score dramatic productivity gains in building snappy, complete, and visually appealing Windows database solutions. Integra VDB offers an event-driven application development paradigm based on high level reusable objects, a low level function call interface for additional flexibility, and access to multiple database management systems through ODBC for Microsoft's Visual Basic, Visual C++ or Borland's Visual C++.

Integra VDB consists of Database Objects (Custom Controls), a Visual Data Manager, Database Functions, ODBC compliant SQL engine and optional ODBC drivers to other databases. Database application templates with sources are also provided to help developers jump-start the application development process.

INTEGRA VDB FEATURES

Database Architecture Controls

Help organise the structure of a database application, generate forms automatically, and allow interactive binding of visual objects to database columns.

- Application, Data Source, Query, and Form Defintion Controls.
- Form Builder Control. Autmatic generation of simple or master detail forms that contain appropriate visual objects and operations to be performed on your database.

Visual Query Builder

Display powerful queries interactively

- Use Query by Example and add Triggers
- Use Drag/Drop features; Relationship Editor; Expression Builder

Database Support.

SQL and non-SQL databases supported through the ODBC interface.

- dBASE IV; Paradox 3.5:
- DB2;DEC Rdb; SQL/400; Oracle; Sybase/Microsoft SQL Servers; Informix; Ingres; WATCOM SQL:
- Free drivers for Microsoft FoxPro, Access and Btrieve:
- Other databases supported:

Database Class Libraries

A set of C++ classes for database operations to build applications in Mixed or Manual Mode.

Visual Controls

Display information and allow database actions.

- Database Operation, Grid (read only), Check box, Combo Box, Frame, List Box, Radio Button and Text Controls.
- Grid Control, Fully editable; optimised data access; cell level control; set attributes like colour, bitmaps, fonts, headers, grid lines on/off etc.
- Mask Edit Control. Automatic data validation.

High-Level Database Functions

A set of functions to build applications in Mixed or Manual Mode.

Free Built-In Integra SQL Engine.

Based on the ANSI SQL 92 standard and ODBC-compliant.

Visual Data Manager.

- Create/drop tables and indexes; Alter and Browse tables; Run Visual Query Builder; Execute SQL command files; Set date formats; Import/export data; Create databases; Create schemas; Create/drop views.

Attention: Users of Borland's Visual Control Pack which includes the Integra VBX Controls can now upgrade to the full Integra VDB package for only £325 (normal price £399)

QBS

QBS SOFTWARE LTD

10 Barley Mow Passage,
London W4 4PH, UK
Tel: +44-81-994-4842 Fax:+44-81-994-3441
BBS: +44-81-747-1979

➤ CIRCLE NO. 600

West Yorkshire and the North

Programmers, Analyst Programmers,
Software Engineers,
Support (Novell, Unix),
also Computer Field Service Engineers.

FOR YOUR NEXT CAREER MOVE
AROUND WEST YORKSHIRE

Telephone Vincent Atherton on Leeds
(0532) 504560 or write to:

AIREDALE RECRUITMENT

Realtex House, Micklefield Lane, Rawdon,
Leeds, LS19 6AX

Airedale Recruitment

LONDON — UNIX/C DEVELOPMENT — TO £20,000 + BENS

Located in a number of offices around London, our client is a successful systems house dedicated to the bespoke development of a range of print management systems. With an increasing number of projects in the pipeline an urgent vacancy now exists for a Programmer with 2-3 years' UNIX/C experience and the ability to work on multi platform systems. The role will involve in house development and systems enhancements as well as specification and support. Ideal candidates will have TCP/IP, NFS and RDBMS exposure as well as the versatility to work under DOS/VAX/VMS/Alpha systems. Superb opportunity for good low level programmer looking for a step up the career ladder!

HERTS — TOP FLIGHT GRADUATES - TOP LEVEL SW HOUSE — £16-22,000

Highly successful British company specialising in the development of forth generation software tools - sales expanding at over 50% per annum, both in the UK and abroad (especially USA and Australia). Vacancies exist for ambitious graduates (1st or 2:1 in Computing/Maths or Scientific based degree) to join the software development team. In addition to a good degree, candidates should offer at least 1 year's C development experience gained on a UNIX platform. There are few organisations who will offer more challenging and stimulating technical work, a commitment to formal training and regular merit based salary reviews.

C,C++ GUI, UNIX — WORKING AT THE LEADING EDGE — £18-28,000

Are you currently working in an environment that challenges you technically? Does your current employer have a commitment to working with the latest UNIX, Client Server software? If the answer to these questions is no then now could be the time to move on! Our client , a major UK computer services organisation has urgent requirements for graduate calibre UNIX developers who have upwards of 1 year's UNIX and C and /or C++. Knowledge of Graphical User Interfaces and RDBMS (SYBASE, Oracle, Ingres, Informix) would be advantageous but training will be provided to successful applicants. Locations include the South East, West Country and the Midlands.

CITY — UNIX/C/C++ BANKING — TO £26,000 + BANK BENS

Leading international banking organisation has 2 urgent vacancies for Analyst Programmers. The successful candidates will work within the Group Treasure Division as part of a small team developing UNIX based trading systems in C and C++. Applicants must have at least 2 years' relevant technical experience and possess the interpersonal skills (and patience) to liaise closely with dealers in both development and support situations. Excellent salary packages on offer in this exciting financial environment.

LONDON — SOFTWARE ENGINEER - DEALING ROOM — TO £27,000

It's the over 14 years' specialist experience, the very latest technological environment and the development of leading financial/front office systems which makes this opportunity so very attractive! And while the standards set by our client are high, the reward is a career which will utilise all your skills and offer you are a rare chance to work with an extremely talented development team. With at least 3 years' UNIX (SunOS) experience, candidates will have strong C/C++ and X-Windows/Motif/GUI. Your design experience and your development skills could now lead you into another career dimension!

9am-7pm Monday-Friday call 081-780 1694
evenings/weekends 0895 422131 or Fax 081-780 9844.
PROGRESSIVE COMPUTER RECRUITMENT, EUROPA HOUSE,
266/276 UPPER RICHMOND ROAD, LONDON SW15 6TQ

ASH ASSOCIATES

DORSET

This small software house developing bespoke Windows applications for clients across the South is expanding. They seek two Programmers/Analysts with at least 1yrs Visual Basic and Windows design experience to be responsible for the development of applications for Manufacturing Financial and Retail markets.
Salary to £20K + Benefits.

BUCKS

Our client is expanding. They seek software design engineers to work on projects including Compilers and Device Driver development for both Graphics and Network Hardware. Applications are targeted for the PC, Sun and Apple Macintosh environments. You will be young, degree qualified 2i or higher with at least 3yrs C and Assembler experience. *Salaries to £35K.*

PARIS - FRANCE

Our client a designer of Windows based applications software for the Oil Exploration Industry seeks a software design engineer to work in Paris. You will be young highly qualified with at least 3yrs C and 80X86 assemblers gained in a PC, Real Time development environment. Experience of Data-Acquisition/Recording or Windows an advantage as will be French.
Salary c£25K+.

BUCKS-LONDON-SURREY

These three companies design and develop both software and hardware Network Products for the PC and Workstation markets. They seek software engineers at all levels. You will need a minimum of 3yrs C experience gained commercially in either a PC or DOS/Windows or Sun/Unix environment. The ideal candidates will be highly qualified and have previous Network development experience (NOVELL, TCP/IP). *Salaries to £28K.*

HERTS

This company a designer of Communications Products are experiencing excellent growth, seek a software engineer with in depth OS/2 Windows (Presentation Manager) and C++ experience. You will be highly qualified with the ability to manage a Project and direct the work of Junior engineers. *Salary to £30K.*

SURREY

My client seeks Degree Qualified (MSc) software design engineers with both C and Assemblers gained commercially in a Real Time development environment. You will be working as both an individual and as part of a Team on High Technology projects including Imaging, Simulation, Industrial Control and 2D/3D Graphics.
Salaries to £25K.

Call James Hunt, Ron Cook or Jasmin Rio on TEL: (0425) 475480 or Fax: (0425) 480807, write to Ash Associates-(TOGA), First Floor, 39-41 High Street, Ringwood, Herts BH24 1AD.

CATALYST CONSULTING

Search Selection and Advertising
Parade House, 135 The Parade,
Watford, Herts, WD1 1NA
Tel: 0923 240139 Fax: 0923 249436

CAREER OPPORTUNITIES IN THE SOUTH EAST

C++ NT	Analyst Programmers	£38k
C++ UNIX	Analyst Programmers	£38k
C UNIX	Software Engineers	£30k
VMS, INGRES	Analyst Programmers	£35k
INGRES	DBA	£40k
VMS C	Analyst Programmers	£35k
FORTRAN VMS	Real Time APs	£35k
X Motif	Analyst Programmers	£35k
SYBASE	Distributed Database Architect	£65k
SYBASE	Database Designer	£50k
C Windows DLL	Analyst Programmer	£30k
Visual Basic	Analyst Programmer	£35k
VMS VAX Basic	Analyst Programmer	£30k
PC / LAN	TECHNICAL SUPPORT	£35k
SUNOS	Systems Administrator	£35k
C Motif Unix	Analyst Programmer	£35k
Excel VB C	Analyst Programmer	£35k

EXCEPTIONAL OPPORTUNITIES FOR GRADUATES
WITH MIN 2.1 IN A NUMERATE SUBJECT

THESE ARE A SMALL SAMPLE OF OUR CURRENT VACANCIES. IF YOU ARE CURRENTLY LOOKING TO CHANGE JOBS OR ARE CURIOUS ABOUT OPPORTUNITIES AVAILABLE TO YOU, CALL US NOW.

Guidelines for the goodlife...

Les Peck goes through the basics of independent employment.

In most cases, contracting in the IT world will work in our favour. Financially: the daily rate is better, there's no more PAYE. Our tax is deferred, we can register for VAT and claim it back on legitimate business expenses. We can pay ourselves director's fees and dividends. And if we run a company car, almost all our travelling can be set against profits.

As contractors, we can pick the kind of work we want to do - and if the market's right, we can also choose the location. Age is not important, there is still a demand for the older skills: read the specialist adverts; they often want people who know the older languages. And as most contracts only last about six months, the eternal game of 'office-politics' is someone else's problem.

To break into contracting, the first thing you need is a good CV, which needs to show at least two years of practical experience. Ultimately, a clean, clearly laid-out document in an easily legible font is desirable: try to make your CV easier for the client to read *now*, than leave till later. Current thinking recommends that you write for what you *want* to do - not what you may have done. As a technical author, my CV begins with all the wordprocessing software I use, then lists my contracts in reverse order - three to a page. If my client is looking for a Word-for-Windows man who can write User Guides, he need look no further than page two: it's what I want to do - not what I've done. It works for me.

Under the current set of rules, contractors should operate as Limited companies. Basically, this means doing the job and running a business at the same time. It's not as hard as it sounds. Most contractors use accountants. But they have to be the right accountants - ones who specialise in the IT line of business. Some will even handle the day-to-day money matters - like tax and VAT. Check their adverts in the trade magazines and call one.

When it comes to finding work, I prefer to use an agency . Many of the better ones advertise in the IT magazines. Check them out. Find the ones who handle your discipline and call them. Select one large agency, one specialist, a smaller company, and five others picked at random.

Clients often tend to fill their jobs from several sources. If you flood the market, you risk a multiple submission - and if the client is unwilling to offend his favoured agencies, he'll go for someone else. If you're the best available, the job will find you. But call your agencies from time to time, just to keep in touch.

Finally, you should be prepared to travel. The jobs don't come to us - we have to go and catch them. In the past five years, I've worked in Windsor, London, Southend, Birmingham, Manchester and Bristol. Some days, it feels like a lousy life - but it keeps the dog in biscuits. So get out there and good hunting!

ADVERTISERS INDEX

ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE	ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
AL Downloading	Development tools	571	37	Nu Mega	Programming tools	565	24
Asymetrix	Info Modeler Database design	560	11	Poet	Object Oriented Database	569	33
Bits Per Second	Graphics tools	583	53	Pygmalion	Training	578	43
Borland	Development tools	558	7	QBS	Windows programming tool	600	81
Cebra	Dual screen accelorators	579	43	Q & E Software	BASIC database	564	23
Citadel	Comms library	563	19	Quarterdeck	Memory management	566	27
Contemporary III	Development tools	585	57	Rainbow Technologies	Security products	588	65
Contemporary	Programming tools	564	45	Real Techniques & Methods	CASE tools	602	OBC
Crag Systems	CASE tools	575	41	Recital	Development tools	568	31
DES	Software Protection	582	51	Richfords	Training	577	41
Easel UK	Development tools	561	13	Ridgeway Press	Computer Manuals and Printers	598	77
F1	Visual Basic training	573	37	Salford Software	Salford C/C++	592	69
GWI	Development tools	69	591	Scientific Software	Development tools	594	73
Grey Matter	Programming tools	557	5	Sequiter	Development tools	595	75
Hypersoft Europe	Programming tools	590	67	Software Security	Security dongles	586	61
Institute of Analysts/Progs	Institution	576	41	Softwerk	Development tools	589	67
Intasoft	Version control	587	63	Staria	Telephony	584	55
Interactive	Software Development forum	574	39	Systems FX	Development tools	559	9
Knowledge Garden	GUI development tools	556	2	System Science	Programming tools	570	35
Magnifeye	Software Protection device	581	47	System Star SoftTools Ltd	Programming tools	593	73
Mercury Interactive	Testing tools	601	IBC	Trevan Design	Bulletin Board	597	77
Microcosm	Copy control	572	37	User Friendly	Software Copy Control	596	77
Microsoft	Visual C++	555	1	WordPerfect	Development tools	562	15
MKS	Development tools	599	79	Zinc	GUI library	567	29

EXEnders

Yearning for a PC-motif teapot? Never fear: EXEnders has come up with the computer event of the year...

You see, it all comes down to your basic method and medium. Your *average* soap opera, well now, that has at least a couple of camcorders, maybe even some lights and one of those 70s-dusters-on-a-pole for the sound (1). *We*, on the other hand, have half a side of A4, in black and white. Production has been busting a gut for the last month trying to overcome the crippling drawbacks that those who attempt the paper-based Soap are faced with. Our man X has been staunch. Shortly after finishing work on the Missile Launching Mac (2), he devoted his ingenuity and expertise to persuading the Publisher of the need for some kind of media backup. Perhaps a Newton Message Pad with next month's issue - to animate the enthralling and ongoing tale.

But she wasn't having any of it... I know. I know... write in and complain. Still, for the moment then it's back to the old Shanks' Pony of Interactive Media... the imagination.

Picture if you will the scene... a bustling cavern of light. As we pan down from the highest viewpoint we see below us a complex grid of building structures. Workers scurry from Cathedral-like dome, to spatial 'scaffolding', to the more common and outlying white-walled constructs. As yet, the construction work appears unfinished, but who can resist to ask - 'Just what is this awesome conurbation, Ecosphere2 perchance? A vision from the future?'. Nope, this is, of course, the 'LOVE Your Computer' Show. Rows of glistening cars in the grounds outside proudly display their 'I ♥ My Computer' (3) stickers along with little fluffy PCs dangling from the mirror. You thought you'd seen it all didn't you? But until now, nobody had come up with a show that catered for the complete Computer Lifestyle. Here we see pizza delivery boys scurrying through the crowds (4), as happy Compuphiles wrestle their way through the exciting Paper Free Obstacle Course (anyone who manages to leave the building without any promotional information or carrier bags is awarded the special 'LOVE' prize (5). On offer are 1000's of essential items for the Computer-Lover: pens, pencils, mugs, t-shirts, teapots, toilet roll holders... You name it, you can have it with the computer logo of your choice. And nestled amongst the stalls, in the very heart of things is, of course, the subject of this Soap. The Software Developing magazine. The scene is terrifying as snarling developers, programmers and those appreciative of beauty, fight for their free copy. Perched on the top of the stand sit three of the trusty employees. Their faces a picture of woe, as they discuss possible exit routes. Is this the face of shows-to-come we ask ourselves? Well, maybe not. But I'm afraid you're just going to have to wait another month to find out. Yes, we've run out of space already. Start humming the theme tune please, as you wonder to yourself 'Will they make it out alive, is this going to delay next months issue, and where can *I* get some little fluffy PCs from then?' **M**

1) Trades Descriptions have advised against mentioning the possibility of actors.

2) Stops the cleaners playing Tetris.

3) Or the slightly more risqué 'Software Developers Do IT In Front Of A Nuclear Reactor'.

4) Compupizza™ decided to take a risk at this particular show and reduce its profit share to a mere 400%, taking the price of a slice of cheese and tomato with anchovies down to £7.75.

5) A carrier bag full of promotional information, suckers.

VB, C++, FOX SUPPORT

Founded in 1975, Microsoft® has become the worldwide leader in software for personal computers. The company offers a wide range of products and services for business and personal use, each designed with the mission of making it easier and more enjoyable for people to take advantage of the full power of personal computing every day.

Microsoft is committed to making it easy for our customers to get the best from their PC systems through the right combination of software and service.

Personal computer usage is growing ever more diverse, and IT managers, business PC users, students, home computer enthusiasts and many other kinds of user all have widely differing service needs and expectations.

While recognising that specific service requirements are as varied as our customers themselves, we are committed to ensuring that every Microsoft customer has access to first-class services to match those needs.

THE DEVELOPER TEAM

In the Developer Support Team we provide support on all Microsoft Developer Tools products. Our work is extremely varied, ranging from straightforward queries to complex developer cycle issues. We pride ourselves on our expertise, not only with Microsoft products but in all aspects of the developer tool environment, and we apply this knowledge on a daily basis supporting our customers.

Due to continuing success in this area we are increasing the number of individuals substantially over the next twelve months. We will be looking for a range of skillsets throughout the year as shown.

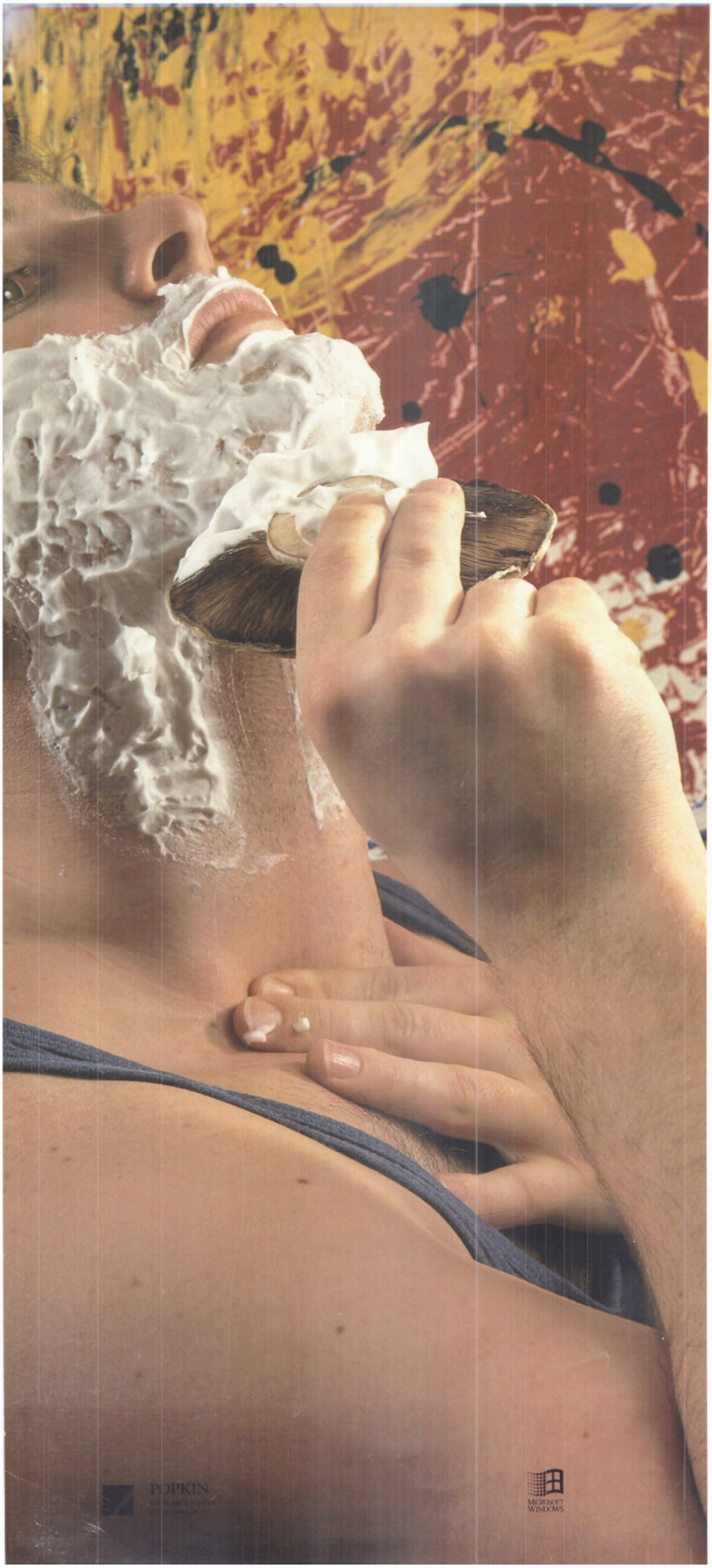
PROFILE - DEVELOPER SUPPORT

- Minimum two years in-depth experience in one or more of the following areas is essential: Microsoft® C++, Visual Basic®, Foxpro® or in software development in an equivalent product
- Excellent interpersonal skills
- Highly motivated
- Focused on teamwork
- Experience in a problem solving environment would be an advantage
- Ideally qualified as a Microsoft Certified Professional
- Strong customer service focus

If you meet the above requirements and are interested, please send your CV quoting reference MS012/EXE to: Recruitment Services, Microsoft Ltd., Microsoft Place, Winnersh Triangle, Wokingham, Berkshire RG11 5TP, or for more information call us on 0734 270416.

Microsoft®

Making it easier



GET THE RIGHT TOOL FOR THE JOB

CASE tools, come in all shapes and sizes and vary in cost from the sublime to the ridiculous. System Architect offers you the features of more expensive CASE tools for a fraction of the cost.

• PC based-Windows and OS/2 • Multi-method support includes: Rumbaugh OMT, Coad/Yourdon, Booch and SSADM • Reverse Data Engineering • Interfaces to client server development tools • Powerful documentation facilities • Multi-user • Customisable • Easy-to-use • Technical Support • Training Courses

RT&M
118-120 WARWICK STREET, LEAMINGTON SPA,
WARWICKSHIRE CV32 4QY. FAX 0926 422165

0926 450858

SYSTEM ARCHITECT

right tool, right price

➤ CIRCLE NO. 602

JOB
C++/MS-WINDOWS

LOCATION CAMBS. — **SALARY** £18-24K

Dynamic Software Engineers are required by this fast-growing Multimedia company. Suitable candidates should offer experience of programming in Borland or Visual C++ under MS Windows. Experience with Multimedia systems, whilst an advantage, is not essential. More importantly, candidates should be bright, enthusiastic individuals who enjoy learning new skills. Working in a small development team, the successful candidates will be liaising with other departments, so good interpersonal skills are essential.

REF:EXE 19

JOB
SDK/'C'/MULTIMEDIA

LOCATION West London — **SALARY** To £20K

An expanding developer of Multimedia titles, our client is looking for two Analyst Programmers The main criteria is a minimum of 12 months' SDK and 'C' experience. If applicants have also developed query tools to access any of the leading PC database engines or have any Multimedia/CD-ROM exposure then this would be of added interest. Successful candidates will be involved in whole project life cycles and will be using the latest development tools.

REF:EXE 21

JOB
C++ / MS WINDOWS

LOCATION Berks. — **SALARY** To £28K

A leading and rapidly expanding developer / manufacturer of laser scanning systems is looking for three Software Engineers with strong C++ skills. Candidates must have in excess of 12 months' C++ experience (gained on a significant project), backed up by a good knowledge of MS-Windows programming. Any CAD / CAM or 3D Visualisation experience would be advantageous, but is not essential. Challenging development projects and highly competitive salary packages are on offer.

REF:EXE 23

JOB
VISUAL BASIC/ORACLE

LOCATION City — **SALARY** To £32K

Our client, a leading City based financial institution, is about to undertake the development of GUI front ends to their existing systems. The ideal candidates for these positions will have strong 'C' / UNIX and Oracle skills, coupled with recent programming experience in Visual Basic. Successful candidates will have the opportunity to work on a variety of projects ranging from Equity to Accounting Systems, while enjoying an excellent team orientated working environment.

REF:EXE 20

JOB
'C' / UNIX / GUI

LOCATION Surrey — **SALARY** To £18-26K

Having just moved to impressive new offices, our client is seeking five Analyst Programmers to work on the development of their networking and communications products. Candidates should have a minimum of 2 years 'C' and UNIX design and development experience and, preferably, have some familiarity with XView, Motif and/or X. Our client is committed to offering their staff excellent opportunities for career development and, therefore, candidates demonstrating management potential will be considered for future team leading positions.

REF:EXE 22

JOB
C++ PROGRAMMERS

LOCATION Herts. — **SALARY** To £30K

These vacancies will appeal to dynamic, young software engineers who enjoy working in a product development environment using the latest Windows technology. Our client, an expanding company, is currently developing its next generation of terminal emulation products and is therefore seeking high calibre Developers.
Suitable applicants should have excellent MS Windows / SDK experience along with a fluency in 'C' or, preferably, Visual C++ / MFC. Excellent salaries are on offer for talented individuals who enjoy a challenging development environment.

REF:EXE 24

CONTRACT VACANCIES - UK WIDE

	Sybase SQL Server	
City	Consultant	6 months
	Oracle 6 / Forms* 3	
Surrey	Programmer	6 months
	Oracle 6 / Forms* 3	
Berks.	Programmer	3 months
	INGRES ABF	
City	DBA	3 months
	GUPTA / 'C'	
City	Consultant	6 months
	Paradox / PAL	
City	Programmer	5 months
	OS/2 / PM	
City	Programmer	3 months
	'C' / Ass/Real Time	
Berks.	Soft. Eng. x 2	6 months
	'C' / DOS	
W.London	Programmer	3 months

REF:EXE / 25

	Windows NT	
City	Developer	6 months
	Windows NT / ’C’	
Middx.	Programmer	6 months
	MS-Windows / SDK / 'C'	
W.London	Programmers x 3	3 months
	MS-Windows / Btrieve	
Herts	Programmer	3 months
	MS-Windows / Banking	
City	Consultant	6 months
	Device Driver / Windows	
Surrey	Soft. Eng.	3 months
	Windows Testing	
Berks.	Soft. Test.	3 months
	X-Windows / ’C’	
Surrey	Soft. Eng. x 2	6 months
	X-Windows / ’C’	
Herts.	Soft. Eng.	4 months

REF:EXE 26

	Visual C++	
City	Developer	3 months
	Visual C++	
W.London	Programmer	3 months
	Visual C++	
Herts.	Programmer x 2	3 months
	Borland C++ / Multi-Media	
S.London	Soft. Eng.	3 months
	C++ / Multi-Media	
London	Developer	3 months
	Visual Basic v3	
City	Programmer	6 months
	Visual Basic v3	
Oxon	Programmer	3 months
	Visual Basic / MS-Access	
City	Programmer	6 months
	Visual Basic / SDK / ’C’	
Surrey	Developer	6 months

REF:EXE 27

Logistix Recruitment Limited
Lamb House, Church Street
Chiswick Mall, London W4 2PD
Tel: 081-742 3060
Fax: 081-742 3061

We have a large number of PERMANENT and CONTRACT opportunities throughout the UK. Please call one of our consultants for further information or, alternatively post/fax a CV to us and we will contact you at a convenient time.

Logistix