NOT SO COSE • JULES BACK TO NATURE • SED SO • MAILING LISTS IN MAJORDOMO

April 1995

£3.20

EXE

The Software Developers' Magazine

Embedded Inspiration

TNT Embedded ToolSuite review

Embedded Systems in the office

Fuzzy control with C++

VxDs all wrapped up

Career Opportunities
Page 92

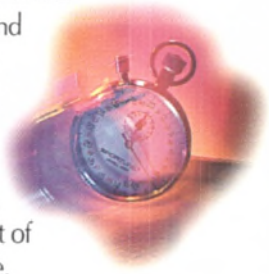
9 770268 687008 04

"They laughed when I sat down to create my Windows to OSF Motif port,

from scratch would easily consume eight months, and a pile of money. Then I discovered MainSoft. They showed me how we could go from Windows to UNIX with no code rewrite."

"We visited MainSoft and brought along our source code. Less than 8 hours later, my editor and GUI were up and running on UNIX. Today, our product has about 250,000 lines of Windows code."

"MainWin also provides a consistent environment across the different UNIX systems. From Sun®, it took us 2 hours to go to HP®. One and a half hours to Alpha. And 3 hours to SGI®. Altogether, we've invested only 5% more effort to support the UNIX platforms, and Windows features like help, printing, common dialogs, print preview, dockable toolbars, DDEML and a host of others all work fine.



But when I had a SUN version just 8 hours later...

...then I laughed, because with MainWin™, I knew I had a competitive edge."

"Our product was a C and C++ programming environment created in Windows™ 3.1, but our larger customers were also asking for UNIX® versions. To rewrite our GUI code

The same code base that runs this Windows application...

...also runs this UNIX application. The secret – MainWin.

"And our UNIX version has exactly the same code except for about 100 lines."

"Now that we've standardised on the Windows API and MainWin, we're enhancing our product via a single code base, and compiling on Windows 3.1, NT, and multiple UNIX systems for every engineering release. We even ship the PC and UNIX versions on the same day."

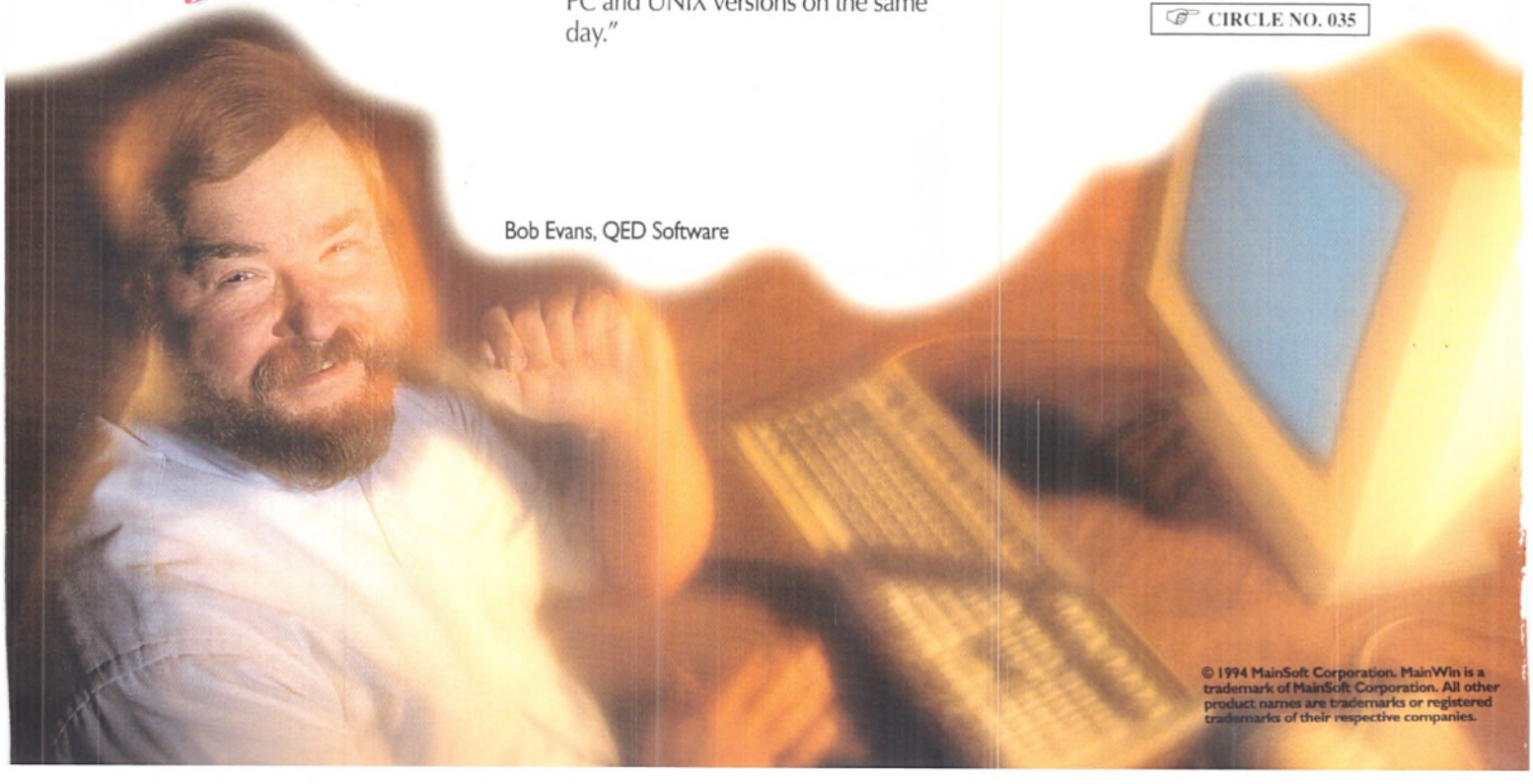
For a full information pack on developing cross-platform with the Microsoft Windows API or MFC contact Personal Workstation Software **0171 231 0333**.

MAINSOFT CORPORATION

PERSONAL WORKSTATION SOFTWARE

5 Lloyds Wharf, Mill Street, London SE1 2BA
Voice +44 (0) 171 231 0333
Fax +44 (0) 171 231 8688
Email sales@perwork.co.uk

CIRCLE NO. 035



Bob Evans, QED Software

EXE
The Software Developers' Magazine

CONTENTS

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

(01364) 654100

FAX: (01364) 654200

C & C++ FOR DOS

Comms

Async Prof for C/C++ 2.1	£170
Greenleaf CommLib 5.1	£235
MagnaComm/DOS	£215
SilverComm "C" Asynch 4.0	£195

Database

c-tree Plus 6.4b	£565
D-ISAM	£495
DataBoss 3.5 (Special Offer)	£79
SoftFocus Btree/ISAM	£75

Graphics

3D-Ware Prof	£249
Essential Graphics Toolkit	£320
Fastgraph 4.0 (Ted Gruber)	£195
GFX Graphics/Fonts & Menus	£300
GX Graphics 3.0	£155
GX Images	£399
MetaWINDOW-DOS 4.4	£170

GUI

4Sight	£300
MEWEL/GUI Std 4.0	£390
object-Menu w/Source 2.7	£510
Zinc Engine & DOS Key 4.0	£505

Math & Scientific

C/Math Toolchest & Grafix	£45
Huge Virtual Array & NAT 3.0	£215
Science, Eng & Graphics Tools	£140

Memory Managers

MegaHeap	£115
MoreHeap 3.23	£135
VMData for DOS	£665

Printer

BD/CX (DOS)	£300
BGI Printer Driver Toolkit	£105
GX Printer 1.5	£240
SLATE	£260

Screen

C-Scape 4.0	£360
C/Windows Toolchest	£45
Greenleaf Datawindows 3.0	£225
Panel Plus II	£260
TCXL UI Dev Sys for DOS/Win	£140

General & Systems Libraries

GX Sounds	£200
Interwork	£205
MTASK	£215
Multi-C w/Source	£39
TTSR Ram Res Dev Sys 2.0	£105
TSRs and More	£130

Tools

C-DOC 6.0	£175
C-Vision for C 3.1	£95
CC-Rider Prof	£235
Clear+ for C	£135
CodeCheck (Professional)	£475
PC-Lint for C/C++	£139
SourcePrint+ 5.0	£210

C & C++ FOR WINDOWS

Asynch Pro for Win	£135
COMM-DRV/LIB	£140
Greenleaf Comm++	£180
Greenleaf CommLib Pro	£365
Borland Database Engine 2.0	£205
CodeBase 5.1	£259
CXBase Pro	£500
DataBoss for Windows	£410
DataDirect Developeris Toolkit	£450
DataTable	£145
DBTools.h++ for ODBC	£340
Greenleaf Database Library 4.0	£180
Integra VDB/C++, Desktop	£209
List & Labels for Win (Pro)	£395
POET SDK 2.1	£360
Raima Database Manager	£380
Spread/VBX++	£175
Charting Tools for Win	£245
Essential Chart for Win	£320
GraphiC/Win 7.0	£360
Graphics Server 4.0	£245
Real Time Graphics Tools	£480
Accusoft Image Lib/Win 5.0	£670
Ad Oculos (Image Analysis) 2.0	£325
ImageMan	£399
LEADTOOLS Prof 4.0	£675
PCX Toolkit for Windows	£155
Borland Visual Solutions Pack	£59
Control Palette 2.0	£135
Crusher! Win DLL w/Source	£235
Diamond Toolkit	£310
Greenleaf ArchiveLib	£210
HeapAgent	£420
KPWin++	£600
Lapack.h++ w/source	£675
M.4	£660
Math.h++ 4.1	£340
Money.h++	£340
MS Visual Control Pack	£69
PKWare Data Comp Lib for Win	£175
ProtoGen+	£220
RWCanvas.h++	£340
SafeWin 3.0	£190
TCOMP/Multi-Platform	£105
TG-CAD Prof 5.0	£795
Tools.h++ 6.0	£340
WinMaker Pro 6.0 (intro)	£360
WinWidgets++	£240
Zinc 4.0 for Windows & NT	£505

BASIC COMPILERS

BBC BASIC-86 Plus	£75
CA-Realizer 2.0 (OS2&Win)	£179
GFA-BASIC for DOS	£80
GFA-BASIC for Windows	£80
GFA-BASIC for Win Compiler	£54
PowerBASIC 3.1 (DOS)	£109
PowerBASIC Pro (Win&DOS)	£225
TrueBASIC Std 3.0	£88

DELPHI

Delphi	£185
Mobius Business Builder (Intro)	£199
Mobius Draw Kit	£99
Mobius WinG Sprite Kit	£99
Orpheus	£135

PROGRAMMING tools

Ada	Assemblers
Basic	C/C++
Comms	Cross Dev
Custom Controls	Database
Debuggers	Dos Extenders
Editors	Fortran
Graphics	GUI
Linkers/Locaters	Lisp
Modula-2	Multi-tasking
Pascal	Prolog
Smalltalk	SQL
Version Control	Visual Programming
Windows	Xbase

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

VISUAL BASIC FOR WINDOWS

Visual Basic 3.0 for Win Std	£96
Visual Basic 3.0 for Win Prof	£230
Comms Lib/Win	£115
Distinct TCP/IP - Visual Edition	£265
NetWare Client SDK for VB	£169
Network Library/Win	£80
PDQ Comm for Windows 2.1	£99
ADE/VBX	£350
CodeBasic 5.1	£140
Controls for Btrieve	£185
Crystal Reports Pro 4.0	£295
DataDirect Multilink/VB	£270
Grid/VBX	£75
Integra VDB/VB, Desktop	£209
List & Labels for VB	£279
TrueGrid Pro 2.1	£105
VB/ISAM MU	£150
VBtrv for Windows	£185
Visual/db Network 3.0	£235
Accusoft Image Lib/VBX 5.0	£500
ImageKnife/VBX 1.3	£175
ImageMan/VB 3.0c	£239
ImageStream/VBX	£180
JPEG Image Compressor VBX	£169
PCX Toolkit for Windows	£155
3-D Widgets	£89
ChartBuilder for VB	£75
Custom Control Factory	£49
Data Widgets	£99
Designer Widgets	£99
First Impression	£180
Formula One 2.0	£180
Gantt/VBX	£195
HighEdit 3.0	£199
MS Visual Control Pack	£69
Real-Time Graphics Tools for VB	£390
QuickPak Pro for Win 3.0	£145
Spread/VBX	£175
SpyWorks-VB	£120
TAB/PRO	£75
TX Text-Control Standard	£180
VB/Magic Controls	£120
VBAssist 3.0	£135
VBGraphix	£270
VBTools 4.0	£115
VB/Rig Professional	£79
WinWidgets/VBX	£160
3D Graphics Tools 4.0	£130
Code.Print Pro for VB	£85
Compression Plus for VBWin	£175
d-Barcode VBX/DLL	£94
DynaZIP VBX	£129
FXTools/VB Std 1.5	£149
MediaKnife/VBX	£244
Microhelp Muscle/Win	£125
QuickPak Scientific for Win	£105
TMS Tools	£99
VB Compress	£99
VB Language Manager	£130
VERSIONS/VB 1.1	£80
Visual Developeris Suite	£290
Visual Instrument Panel Controls	£150
VisualSpeller	£120
VisualWriter	£180
VSView	£105

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Access 2.0	**£289**
Microsoft Fortran 5.1	**£138**
Microsoft Fortran Power Station	**£230**
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 Win Std	**£96**
MS Visual Basic for Win Prof	**£230**
Microsoft Visual C++ Std	**£65**
Microsoft Visual C++ 2.0	**£299**
Microsoft Windows 3.1	**£78**
Borland Database Engine	**£205**
Borland C++ 4.5	**£280**
Borland Pascal with Objects 7.0	**£250**
dBase 5.0 for DOS	**£405**
dBase 5.0 for Windows	**£280**
dBase Compiler 2.0	**£245**
Delphi	**£185**
Turbo C++ 3.0	**£66**
Turbo C++ Visual Ed. for Win	**£94**
Turbo Pascal 7.0	**£94**
Turbo Pascal for Windows 1.5	**£99**
Paradox for Win 5.0	**£280**

WITH FULL TECHNICAL SUPPORT

DATABASE & CLIENT SERVER

Clarion for Windows	£450
DataFlex for Win	£390
Delphi Client/Server	£830
Equinox	£279
Idealist for Win Read/Write	£105
Microsoft Access 2.0	£289
Microsoft FoxPro Pro 2.6	£360
dBase 5.0 for Windows	£280
ObjectView Desktop	£190
OpenInsight 2.01	£345
Paradox for Win 5.0	£280
PowerBuilder Desktop 4.0	£469
SQLWindows Corporate	£2755
SQLWindows Network	£1835
SQLWindows Starter	£945
SQLWindows Solo	£105
VisualAge	£1415
VisualAge Team	£2815
Watcom SQL for Win	£230
DataDirect ODBC Pack for Win	£320
Forest & Trees for Win Std	£615
InfoModeller Desktop	£355
Monarch for Windows	£365
Quest	£500
Quest Reporter	£255

News & Views

ATTENTION ALL C/C++ PROGRAMMERS! – THE BEST IS NOW BETTER & CHEAPER!

WINMAKER PRO 6.0 - ONLY £360

WindowsMaker Professional from Blue Sky has always enjoyed a superb reputation as the leading prototyper/code generator for C/C++. Now there is a new version, renamed WinMaker Pro, and a special price until the end of April.

A new Project Manager lets you access and edit all your objects & resources. 3 new floating palettes have been added - the Toolbox lets you drag'n'drop resources & controls into an app; the Property Sheet lets you set styles, characteristics & event options of any controls; and the Colour & Font Palette lets you set colours & fonts for any control from one place. It also supports VBX's (OCX's to follow), application templates, and includes a Multimedia module (AVI & WAV). It can target MFC, OWL or raw API on Windows, Win32, Windows95 & Windows NT and can integrate right into the Visual C++ Visual Workbench or Borland C++ IDE.

What are you waiting for? Call us for full details and order it before the price goes up to £725. If you have been dreaming of a C++ version of Delphi for your front- ends (no, there won't be one!), WinMaker Pro is as close as you'll get.

DELPHI Price Correction

Last month's pricing was all wrong!! Delphi is actually on special offer until the end of May for £185.

VisualAge from IBM - Fully OO Client/Server

If you are looking for a fully object-oriented client/server development system based on a visual construction-from-parts architecture, look no further. Both OS/2 & Windows versions, single or team versions, based on IBM Smalltalk, suitable for complex, industrial strength development. Add-ons for comms, multimedia, access to Oracle & Sybase. ONLY £1415.

OS/2 Personal Pascal

If you want to learn Pascal on OS/2, then this is the ideal package. On-line docs, samples, tutorials, full 32-bit native code compiler, debugger, etc. From Cabot. ONLY £69

C & C++ COMPILERS

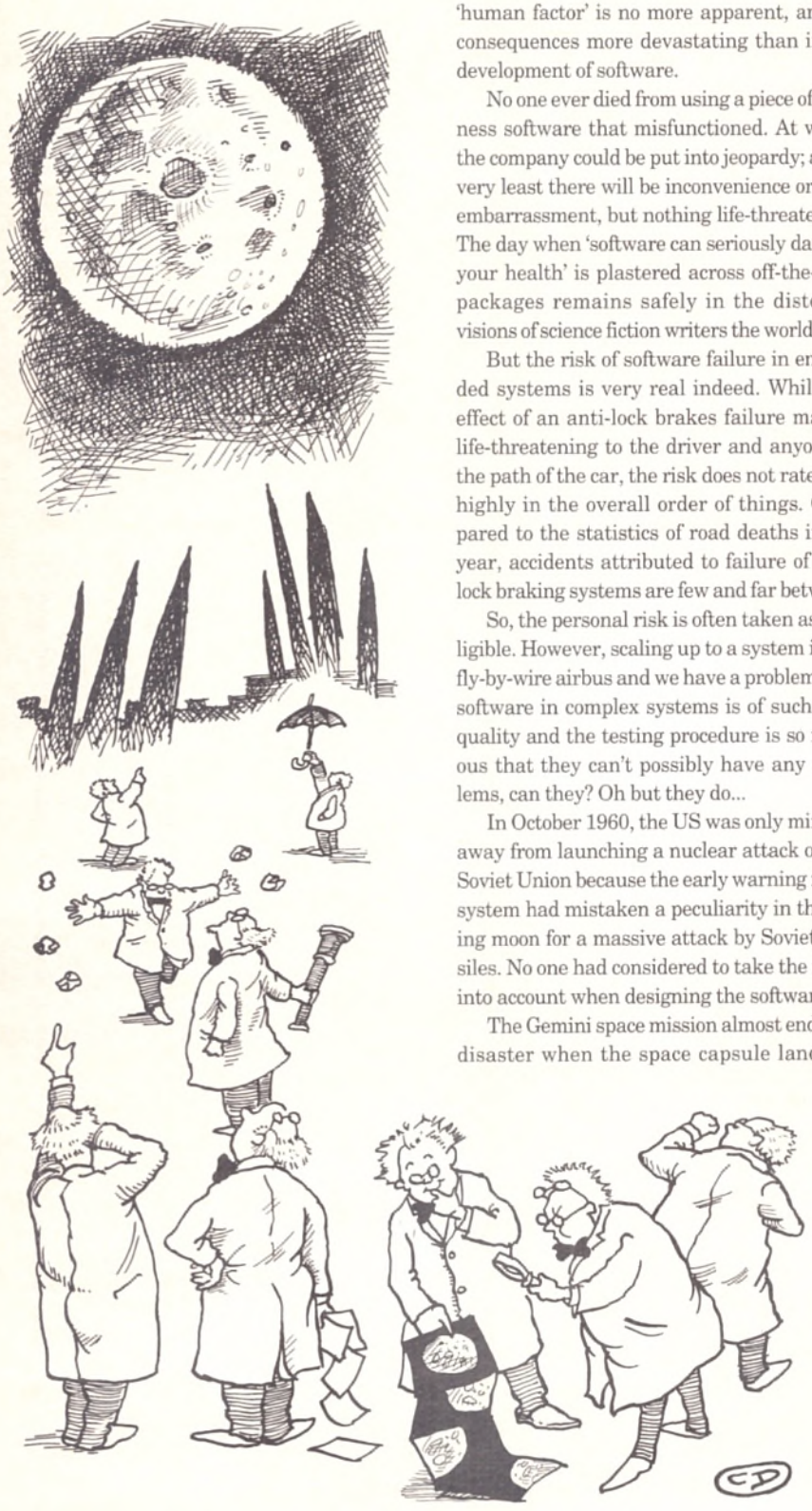
Borland C++ 4.5	£280
Borland C++ 4.5 & DB Tools	£405
High C/C++ 386 3.3	£620
High C/C++ for Win NT	£445
Microsoft Visual C++ Intel Ed	£299
Salford C/C++ DOS Developer	£195
Salford C/C++ Win Developer	£395
Symantec C++ 6.1 Prof	£325
Watcom C/C++ 10.0	£155
Aztec Embedded C86	£290
Microsoft Visual C++ Std	£65
Symantec C++ 6.1 Std	£85
TopSpeed C/C++ Prof	£215
Turbo C++ 3.0	£66
Turbo C++ Visual Ed. for Win	£94

VISUAL BASIC FOR DOS

MS Visual Basic for DOS Std	£96
MS Visual Basic for DOS Prof	£230
NetPak Pro for DOS	£135
PDQ Comm	£65
QuickComm	£129
QuickNet	£79
RS232 Library	£130
db/Lib Prof 3.0	£195
VB/ISAM MU	£150
Graphics QuickScreen	£105
Graphics Workshop	£105
GraphPak Pro	£99
Compression Plus	£79
Compression Workshop	£110
PDQ	£110
Printer Plus	£130
ProBas 5.6	£169
QuickPak Prof 4.19	£145
QuickPak Scientific 3.0	£105
QuickScreen 4.0	£105

Poor workmanship

As software extends into more facets of our lives, quality is becoming increasingly important.



It is often said that we learn from our mistakes. But somehow the same mistakes are never banished completely. Infallible as we may think we are, in spite of countless warnings, often we are stumped by the same mistakes as our peers. Such is human nature, and perhaps we should pay more attention to the 'human factor' in everything we do. The 'human factor' is no more apparent, and its consequences more devastating than in the development of software.

No one ever died from using a piece of business software that misfunctioned. At worst, the company could be put into jeopardy; at the very least there will be inconvenience or even embarrassment, but nothing life-threatening. The day when 'software can seriously damage your health' is plastered across off-the-shelf packages remains safely in the distorted visions of science fiction writers the world over.

But the risk of software failure in embedded systems is very real indeed. While the effect of an anti-lock brakes failure may be life-threatening to the driver and anyone in the path of the car, the risk does not rate very highly in the overall order of things. Compared to the statistics of road deaths in the year, accidents attributed to failure of anti-lock braking systems are few and far between.

So, the personal risk is often taken as negligible. However, scaling up to a system in the fly-by-wire airbus and we have a problem. But software in complex systems is of such high quality and the testing procedure is so rigorous that they can't possibly have any problems, can they? Oh but they do...

In October 1960, the US was only minutes away from launching a nuclear attack on the Soviet Union because the early warning radar system had mistaken a peculiarity in the rising moon for a massive attack by Soviet missiles. No one had considered to take the moon into account when designing the software.

The Gemini space mission almost ended in disaster when the space capsule landed a hundred miles off target and nearly sank before the recovery ship could arrive. Here the engineers forgot that in the three days of the mission, not only had the Earth spun on its axis three times but it had also moved three days further into its orbit around the Sun.
These were design flaws. The specs had not considered the rising moon or the orbit of the Earth when they were drawn. The Shuttle Discovery suffered a peculiar implementation error that caused it to miss a rendezvous with a satellite by 700m. The bug was put down to using single-precision rather than double-precision floating point maths in a situation which called for the latter.

Another catastrophe attributed to an implementation fault occurred at AT&T, when six states on the East coast of the US were without telephones for eight hours. The bug was put down to a missing default condition in a `switch` statement. The cost to the company of that single line of code was estimated at $1.1 billion.

How many of us have used single and double floating point declarations interchangeably? How many of us ignore the default case in a `switch` statement? At least we know, or should know, the behaviour of the `switch` statement without a default case. It is defined in the ISO standard, so there should be no excuse for getting it wrong. What is worrying is the 197 or so issues in ISO C that remain unresolved.

As to C++. For a language that was designed for compatibility with C, there are already 31 differences. A standard is not here yet. In its present state the language is too poorly defined for high end systems.

NASA is an organisation often cited for its high standards in software quality and is regarded as the foremost producer of software in the world. Yet it has only managed to reduce the proportion of bugs in its software to six percent. What hope have we? Six percent of a million lines of source adds up to an awful lot of bugs.

There are a number of commercial development tools which go a long way to guiding us towards producing robust code. To some extent, even a compiler can help. Annoying as they can be, we should not ignore warning messages. They are there because the compiler 'thinks' the programmer has done something, that while valid, is unintentional. Ignoring warnings is rather like driving blindfold. We can go only so far before we crash. ■

Cliff Saran

Hang out with Borland

The trendy thing to do these days is to get onto the Web. Now no one would accuse Borland of being untrendy. It has announced Borland Online, a Web site for developers at `http://www.borland.com`. The site contains Borland press releases, lists of programmes and services run by the company and a 'feedback' section containing discussion groups. There is a corporate section with the company's background, financial figures, strategic directions and partners and a posting of current job vacancies. Presumably those who read the financial pages will figure out the reason for the vacancies. Borland says much of its KnowledgeBase CD is available by a link to the Borland ftp site. As expected, white papers and fact sheets on all Borland products are available. There is also a section called 'The Files' which Borland describes as 'a non-linear environment for discussing new information', and recommends it as 'a place to hang out'.

Good news for old-timers

They say the days of the mainframe are numbered. The question is, how many days precisely? Actually, a recent report for Softlab found that the life span of a mainframe varies from 7.3 *years* to infinity. Furthermore, only 30% of the companies asked said they were actively moving away from the mainframe. And given that 80% of all code being maintained is written in Cobol or PL/1 the need for traditional mainframe skills will be with us for the foreseeable future. Applications that are making the move from mainframe to client/server tend to be the newer ones. These are often built in 4GLs.

Surge

Software developers should brace themselves. Well, that's according to the quarterly Lombard Business Investment Survey. The survey conducted by NOP between December 1994 and January 1995, predicts that the IT sector is set for a surge in sales. The reason it gives is that planned investment by British businesses is expected to reach its highest peak for 18 month. As many as two-thirds of all British companies are planning to increase their IT expenditure. 78% of firms said they were planning to acquire at least one major asset over the next three months. This is an increase of 27% since September 1993. The survey found that while there was still uncertainty in the mainframe/midrange market, PC manufacturers were enjoying unprecedented growth.

Winners and losers and raw data perusers

To many traditional developers, it may come as a bit of a surprise to learn that MS Access appears to have the largest chunk of the Windows database market. In fact, Microsoft claims that Access is the first Windows database to sell over 3 million copies since its launch in November 1992. According to Romtec, MS Access has averaged over 50% of sales within the Windows database market. Microsoft is obviously pleased with the result, especially given the fact that the database market has seen a growth of 250% compared with 1993.

Given that Borland has often been considered king of the xBASE world, it appears that it has lost out in market share with proven dBASE and Paradox technology against Microsoft's new kid in town. What's more, the introduction of dBASE for Windows has done little to regain lost ground.

Extraordinarily, the success of Access, which Microsoft has pushed primarily as an end-user tool, has resulted in increased demand for Access developers Mike Grennon of Greythorn PC recruitment believes the pendulum has swung. '12 months ago the advertisements would have been for dBASE and Paradox.' Certainly, there's a shift from these traditional tools towards Access. US-base *VarBusiness* magazine reported MS Access rated third in the corporate market, behind Visual Basic and C/C++. Surprisingly, it beats established corporate development tools from the likes of Oracle, PowerSoft, Lotus and Gupta.

OpenDoc on Windows: Novell ships its first beta version

At the Brainshare conference held in the US last month Novell shipped its Windows implementation of OpenDoc to 3,000 developers. The Developer Release 1 (DR1) provides Win32 developers with IBM's System Object Model (SOM) and ComponentGlue technology from Novell. ComponentGlue provides a mechanism by which OpenDoc and OLE 2.0 can work transparently together. It allows OpenDoc components to be used in the same way as OLE 1.0, OLE 2.0 and OCX components. Support for drag and drop, clipboard and data linking of OLE and OpenDoc objects can take place within and between documents.

The built-in SOM support complies with OMG's CORBA standard for communication between distributed objects. With SOM it should be possible for developers to create OpenDoc components that not only can work on a single desktop, but also via a gateway to cross-platform targets. The DR1 provides automation and scripting services which implies that developers will be able to control OpenDoc using OLE automation in tools such as Visual Basic. An OSA (Open Scripting Architecture) interface is provided to enable third party developers to produce their own scripting tools.

Novell says that the next release (DR2) will be available in May. It is expected that this will include support for additional scripting. The Windows 3.1 version of OpenDoc will be available during the second quarter 1995.

Developers can request a copy of the OpenDoc Windows DR1 and online technical documentation by sending email to `opendoc@wordperfect.com`, or by posting a message in the OpenDoc Forum on CompuServe. The developer kit release of OpenDoc for Windows can be downloaded from `ftp.wordperfect.com`.

Sun takes the enterprise

Sun makes its move on the enterprise marketplace this month with a multi-user version of Solaris. Versions of the Solaris Base Server and Solaris Network Server have been optimised for text-based applications for the x86 platform. The underlying operating system is Solaris 2.4 x86 which Sun says will support many Unix v3.2, Interactive Unix and SCO Unix applications directly. On top of this, there are an estimated 500 or so native Solaris applications available. The new version of Solaris offers improved virtual terminal support and a character-based user interface. The minimum system requirements is a 386 processor or above, with 8 MB of memory and 45 MB of hard disk space. Sun says it will continue to support Interactive Unix and will provide a migration path for existing users.

Delphi Introductory offer £199 + VAT

RAD → ROI

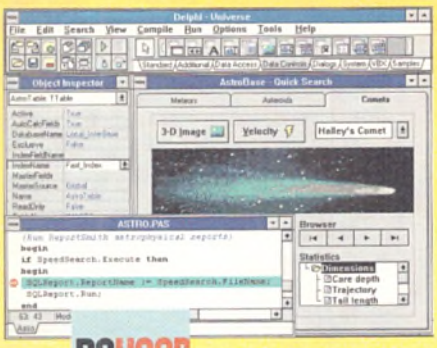
RAPID APPLICATION DEVELOPMENT RETURN ON INVESTMENT

"... Delphi is really very good indeed - no, make that 'excellent'..."
Windows User, March 1995

Delphi is the faster way to high performance applications

"Delphi is Borland's best product for years."
Tim Anderson, Personal Computer World April 1995

If you program . . . you have to see Delphi™. You'll quickly see how its breakthrough combination of the world's fastest optimising native code compiler, visual Rapid Application Development tools, and scalable database technology will give you a tremendous Return On Investment. And sets you on the fast track to Client/Server and Windows 95.



Delphi's Visual Component Library lets you build your complete client/server applications quickly and easily.

RAD: prototype to production

Delphi's sophisticated visual design and debugging tools help you develop faster. They make it easy to work interactively with your customers, so your final .EXE will fit their needs, increase their productivity, and make you a star! But faster development cycles are just the beginning. The Delphi compiler produces applications that run 10-20 times faster than P-code interpreters. Royalty-free executables let you deploy fast, so you can crunch through the applications backlog in record time. And just wait until your users experience the speed of your delivered apps-they won't want anyone's apps but yours.

pcUSER GOLD AWARD

Smooth scaling to Client/Server

Delphi has a built-in ANSI SQL-92 server, Paradox®/dBASE® access, and ODBC support. So you can rapidly develop full Client/Server applications on your desktop. Then, by upgrading to Delphi Client/Server, you can quickly deploy applications across the entire enterprise with high-performance native drivers for Oracle, Sybase, MS SQL Server, Informix,® and InterBase.® And all the work you do in Delphi today will be fully compatible with Delphi's forthcoming 32-bit, Windows 95 release. Put RADical performance to work on your bottom line today. **Delphi usual RRP £349 + VAT**
Delphi Client/Server RRP £999 + VAT

Reuse your components

Delphi's Visual Component Library (VCL) helps you minimise coding, accelerate project completion, and simplify corporate standardisations. You get a gallery of reusable forms, project templates, dialogs, buttons, multimedia controls, and more. Or add your own reusable objects, DLLs, VBX controls, or OLE 2.0 servers into Delphi's open, integrated, component-based development environment.

Visit your dealer, or for further information call

0800 212727

or return the coupon

DELPHI INTRODUCTORY OFFER
Purchase Delphi before 31st May 1995 for only £199 + VAT

☐ **Please send an information pack.** ☐ **Please call with more detailed info.**
☐ **Please send me more information on the forthcoming development seminars.**

Name_____
Position_____
Company_____
Address_____

Postcode _____ Tel:_____
Numbers of PCs in your company _____
☐ I am responsible for purchasing software. My interest is ☐ business ☐ personal

Please return to: Borland International (UK) Limited,
FREEPOST, PO Box 9, Stroud, Gloucestershire GL6 8BR. **AS/9513/CT**

Borland

Hangover at Hanover

On location at this year's CeBit, **David Mery**, our Eurotrekking Features Editor, scouts out the best software development titbits...

CeBIT is a very long and large show. The statistics are mind-boggling. The show has 6,088 exhibitors, spread out over 321,314 square metres of space and that's without counting all the space between the 22 halls! For those who have never been there, it's an experience. The complex, located in Hanover, is like a small town for the duration of the show. Without ever leaving the premises you have access to (besides the exhibition halls) not less than 32 restaurants, four flower shops, a hairdresser, nine banks, seven food shops, a jewellers and much more. All that's missing is a hotel and an airport. The ones in Hanover itself are not enough. During the show all the hotels in a radius of about 100 miles were booked.

Bigger, better

There's no IT show in the world as big as this one. Last year the number of visitors reached 682,000 including 106,000 foreign visitors, ie non-German. Since the number increases every year, more are expected this time. To that you must add all the staff, about 6,000 journalists and the 6,088 exhibitors from 59 countries. New this year are exhibitors coming from as far as Bangladesh, Egypt, Lithuania, Malta, Mexico, Nepal, Pakistan and Sri Lanka. 54 are coming from Great Britain, or Grossbritannien as they say.

There are four and half halls dedicated to software, but some software vendors are also located in other halls like the PowerPC dedicated hall. And that doesn't include the unofficial pavilions, where companies like General Magic are hiding away from the crowd.

Not so AIMless

Apple, IBM and Motorola, AIM for short, had booked one hall dedicated to the PowerPC. It wasn't as full as the other halls but had some interesting products on show. David C Nagel, Senior VP and General Manager of AppleSoft said that Apple was running a prototype of a PowerPC 604 machine in its labs, on which applications written for 68k Macs were running faster than on any 68k powered Macintosh.

According to James Norling, President of Motorola EMEA, 'even though PowerPC is in its early stages, the key operating systems, such as AIX, MacOS, Windows NT, OS/2, Solaris and NetWare are being made available.' The first two are commercially available. NT and OS/2 are in beta and were shown at the CeBIT. The first version of Solaris is expected for mid '95. The beta of OS/2 Warp for the Power PC was running on the as-yet unannounced IBM Power Personal computers. The beta shown was at the same level of code as Warp for Intel but didn't include the support of Windows even though the icons were there. Also shown was SMART, short for Source Migration Analysis Reporting Toolset. SMART ports automatically Windows 16-bit, 32-bit or OS/2 16-bit source code to OS/2 32-bit code. The current version doesn't support MFC or OLE, but both are planned for a future release. SMART is developed by OneUp but IBM has exclusive distribution rights for one year.

OS/2 for the PowerPC, SMART and many other development tools are available on a CD-ROM updated quarterly. To receive it you have to subscribe to *The Developer Connection for OS/2* ($199). For more information call +45 48101500 for an English speaker. ■

Hot of the press

On the first day of Cebit a news release was issued by Reuters stating that 18 companies, instead of 15, were involved in defining a new standard for linking office equipment (see the article further in this issue). This future standard has been called SmartOffice. Its first goal is to link printers, copiers, fax, PC and a 'new generation of laptop'. The technical development has started under the name of SmartLink Architecture. No officials from the companies involved in the consortium would comment on the announcement but without being specific, Peter T van der Kaag, General Manager of the Corporate Communication Division of Ricoh confirmed that the current offering wasn't complete enough and that Microsoft's At Work is not for low end fax machine. Ricoh's At Work compliant fax machine is at the very high end of the market and even includes a hard disk.

General Magic in Europe

Though General Magic's (see EXE February, p45) developer programme is still at an early stage and supposedly not yet open for developers outside USA, one German company has succeeded in being admitted amongst the happy few. Berner & Mattner, a Munich based technical software development company has finished a demo of a navigation system accessing a public transport database. It received two training sessions about Magic Cap and Telescript from General Magic's partner Philips PACE last December. With the current tools available, browser, compiler and debugger, it wrote the Magic Cap part on a Unix workstation and the Telescript part on a Mac. Berner & Mattner is on +49 896080900 and Philips PACE on +31 40758617.

xBASE für OS/2

The German company Alaska is finishing an xBASE compiler for OS/2. XBase/2 is fully compatible with CA-Clipper 5.1 and automatically generates multi-threaded code. The language has been extended further to take advantage of OS/2 specific features like exception handling, the OS/2 help system or REXX. Once compiled, an application runs in a character windows or full-screen but Xbase/2 also includes some tools to help modify the code to change the user interface to a full Presentation Manager one. The beta programme is complete and the program should be available in May in German and August in English for about £700. Alaska can be reached on +49 69439646 or by email: 100436.1375@compuserve.com.

True GRIT

GFT has released the third version of its multiplatform GUI builder. In addition to Windows 3.1, NT, OS/2 and some Unix support has been added for Windows 95, OS/2 Warp and VMS. The basic product GRITplus has been enhanced with a device concept to write some OS independent code that accesses device drivers, templates and new controls such as notebooks and folders. A report writer and an OO Case tool additional module were present in the previous version but a spreadsheet and a chart module have been added. It is a direct competitor to tools like XVT. GRITplus costs $5000 and is available from GFT (+49 77247065).

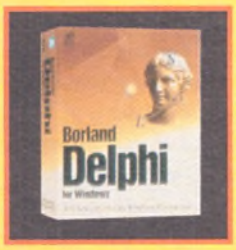
Develop to advantage

Build better applications.

Add performance, functionality & versatility

Borland Delphi

NEW Delphi is here! Delphi facilitates rapid application development (RAD. Delphi has an optimising native code compiler with a visual design environment, and integrated graphical browsing, debugging and project management tools, along with ReportSmith. With the local InterBase Server, Delphi provides for local SQL development, and data access via the Borland Database Engine, facilitates easy scaling to client/server development with Delphi Client/Server. **£199**



DemoSHIELD for Windows Build Interactive Application Demo

DemoSHIELD is a visual development tool for creating interactive software demos and CBT which is quick and easy to learn. Add to your demos, special effects buttons and controls - import video (.AVI) elements and sound (.WAV) to louden the impact. Includes a fast and small royalty free Demo engine, and disk builder for easy distribution. **£425**

Microsoft Test Automate the Testing of your Windows Applications

Microsoft Test is the ideal tool for automating the testing of applications and components (DLL's,OLE) for Windows 3.x, Windows 95 and Windows NT™ . Easy to learn and use, Microsoft Test speeds up testing, reduces costs, and ensures the efficient testing and delivery of high-quality software. All Test Scripts are developed in Test Basic. This is the test every application should take. **£475**

Heap Agent from MicroQuill

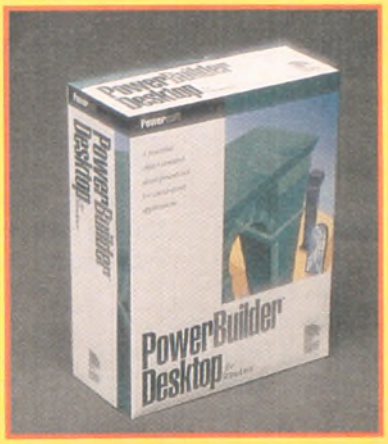
The new Heap Agent for Windows provides complete heap error detection for C/C++ programmers on Windows. Use Heap agent to catch those runtime errors like - overwrites, leakage, double-frees, invalid parameters and references to free or uninitialised memory . Heap Agents browsers and agents help diagnose the errors once they are located. **£325**. Special offer bundle with SmartHeap Windows. **£525**

Microsoft SourceSafe The safest place for all your source code

SourceSafe was acquired in the recent merger of Microsoft with One Tree Software. Source-code control prevents users from accidental code loss, allows back-tracking to previous versions, branching, merging, and the managing of releases. SourceSafe provides robust versioning and configuration management functionality plus security and audit-trail capabilities. SourceSafe supports an open approach, for use with development tools from Microsoft and many other vendors. **£325 per user**

PowerBuilder Desktop for Windows 4.0

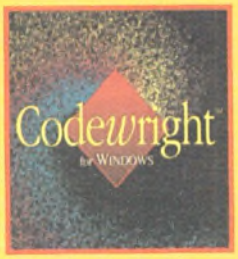
A powerful Windows desktop application development environment with an integrated client/server architecture which includes powerful object-oriented programming tools. With PowerBuilder Desktop, you can easily create robust Windows applications that incorporate data from most desktop databases. Built in is the Watcom SQL database for development and deployment out of the box.



The PowerBuilder family of products includes the Advanced Developer Toolkit (adds database stored procedures, an Install disk builder and more), Lotus Notes Library and the FUNCky Library of useful routines. **Call for information on competitive upgrade pricing for dBASE and Clipper users.**

CodeWright Professional 3.1 Programmer's Editor

CodeWright is a professional quality programmer's editor designed to greatly increase code editing efficiency and provide powerful programming benefits for Windows based development. With emulation for both CUA and Brief, Codewright supports C/C++, Assembly, xBase, Pascal. Key features include Tabbed Output Window, VDOS Command Shell, Help Indexer, User Defined ChromaCoding, and File Based Search and Replace. Available for Windows or Windows NT. **£159**



- PLEASE CALL IF THE ITEM YOU ARE LOOKING FOR IS NOT LISTED
- CALL FOR OUR NEW 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 ACCEPTED WITH PHONE ORDERS

SYSTEM SCIENCE

To Order call - 0171 833 1022
fax 0171 837 6411
3-5 Cynthia Street, London N1 9JF

Your first choice for better development tools

CIRCLE NO. 038

New flavours of Crisp

According to its manufacturer, it's a 'high productivity editor,' it's keystroke compatible with brief and it's Crisp. Pacemaker Software has a new version which boasts support for more platforms including DOS, Windows, NT, Solaris 2, SCO Open Desktop, SVR4, Linux and Linx OS for the PC. It is also available on HP, Sun, IBM, DEC Alpha and Silicon Graphics machines. Crisp provides colour syntax highlighting and built-in. templates for several languages including C, C++, Ada and Fortran. It offers multiple buffers and windows, column cut and paste and a fully customisable macro language. Pacemaker says Crisp can handle data files greater than 100 MB in size. The cost of the Windows version is £135. Crisp on Unix is priced at £350. Pacemaker Software is on 01666 840333.

Golden retriever

Information has never been simpler according to DataWare Technologies. It has produced a set of software libraries called Advanced Search Technologies. The Natural Language Object library enhances traditional Boolean search methods. The Semantic Network Object library offers a conceptual facility with a thesaurus for refining search criteria. Both libraries are compatible with Borland and Microsoft C++ and are available on Windows and NT. They require Dataware's CD Author Development System version 3.21 in order to run. Dataware is on 0181 4240382.

Better Btrieve

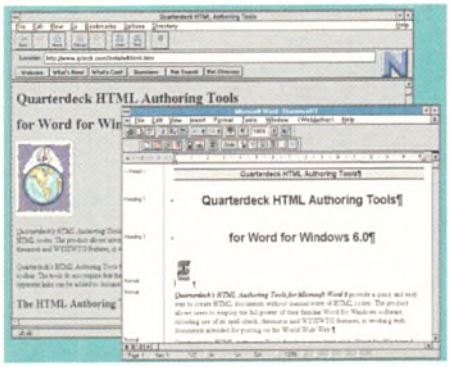
Data Access has worked on improving the performance of its Btrieve engine in DataFlex. The new release of DataFlex for Btrieve contains client runtime code for DataFlex 2.3b, 3.01b, 3.05 for DOS and 3.01 for Windows. The 32-bit DOS platform has been optimised to produce faster and larger applications. According to Data Access, the DOS client included with DataFlex is between 7 percent and 118 percent faster than the 3.05 release. DataFlex Btrieve revision 2.1h is priced at £250 per four user block. Data Access is on 01923 242222.

NFS for NT

Beames and Whiteside has added BW-Connect NFS Server for Windows NT to their collection of connection products. It transforms an NT machine into an NFS file server so that any Unix or other NFS client can access files and printers. It also includes INET server functions such as `telnetd`, `ftpd` and `talkd`. It's available at $295 for a client and a server from Beames and Whiteside (+1 919 831 8989 or sales@bws.com).

Link to Normandy

Normandy is the codename for Quarterdeck's family of WWW tools. The first one, currently in beta test, WebAuthor is for creating HTML 2.0 documents within MS-Word 6.0. It enhances Word's functions with specialised toolbars and menus. It works in both direction. HTML documents can be converted to Word format in order to use all the word processing facilities like the spell checker. In fact during the creation phase, drafts can be saved in Word format. Only at the last stage are they saved as HTML. WebAuthor also provides a custom Internet dictionary of 400 words. Of course, it integrates an HTML 2.0 parser to verify the syntax of existing HTML pages. Other Normandy products are also planned for the first half of 1995. These include a Web server for Windows, a new Mosaic Web browser and other client software. For more information you can call 01245 496699 or look at the URL `http://www.qdeck.com`. Betas of WebAuthor and Quarterdeck's Mosaic Browser can be FTPed from `ftp://qdeck.com`.



A Quarterdeck's Web page in Netscape and its editing with WebAuthor

Completely Warped

The final, completed version of OS/2 Warp is now available. The version currently shipping installs and runs in conjunction with an existing copy of Windows. The announcement is for the 'fullpack' which includes WIN-OS2 containing a copy of Windows optimised to run within OS/2 which is integrated into the operating system. The minimum system requirements are the same as for plain vanilla Warp, that being a 386SX with 4 MB memory.

Included in this version of the operating system is WebExplorer, a native OS/2 web browser. Existing Warp users can also obtain this for free by clicking on the 'Receive Software Updates' icon in the Internet Connection for OS/2 folder. The dialler can now connect to service providers other than IBM and includes Point to Point Protocol support.

Other enhancements include support for Ontrack Disk manager, used on IDE hard drives with capacities greater than 528 MB and more drivers for IDE CD-ROM drives. OS/2 can also read diskettes created using Microsoft's diskette compression technology (DMF) as in MS Office.

The OS/2 Warp fullpack with WIN-OS2 is priced at £125. The standard version costs £85. Existing customers can upgrade from OS/2 2.1 or 2.11 to OS/2 Warp with WIN-OS2 for £70. For more information call the IBM customer support centre on 01329 242728.

VB excess

VB developers get a helping hand from MicroHelp this month. There is a new version of Communications Library which contains not only a VBX, but also 16- and 32-bit OLE custom controls. A DLL is also provided which, apparently is 'optimised for maximum 386 enhanced mode performance'. The toolkit provides 12 terminal emulations, several file transfer protocols and more than 150 modem initialisation strings. File transfer occurs in the background and connection speeds up to 256,200 bps are available. The company claims that Communications Library is the only tool of its kind that can bind to MS Access databases.

Now to help them with their spelling, MicroHelp has a new version of SpellPro and Thesaurus. As its name suggests this is a custom control that adds spellchecking to edit fields. The tool is shipped with both 16- and 32-bit OLE custom control together with a DLL and VBX. Spell-checking dialogs are built-in. The thesaurus contains over 50,000 synonyms. Up to 1,000 user-defined words can be cached. There is also support for multiple dictionaries/thesauri. The company claims that its Dictionary utility gives 50% compression and faster exporting.

Both tools are available from Contemporary Software (01727 811999). Communication Library 3 costs £115 and SpellPro 2, £110.

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**
  54 different custom controls (VBX's) compatible with all Windows development environments that support the VBX extensions.
- **COMPRESSION PLUS**
  A comprehensive set of ZIP - compatible data compression routines for DOS and Windows. Compress files, arrays and screen images to a fraction of their original size, decompressing them only when needed.
- **FAX PLUS**
  Send and receive faxes using almost any class 1, 2 or 2.0 fax modem in ASCII text, BMP, PCX and DCX file formats.
- **MUSCLE**
  Over 600 functions to enhance the speed and capability of your applications.
- **SPELL PRO & THESAURUS**
  A spellchecker and dictionary of synonyms DLL that can be used by almost all Windows development environments.
- **VBVIEWER**
  A file viewer custom control that can display over 20 different file formats.
- **COMMS LIB 3**
  Reliable, easy to use communications routines with a wide variety of terminal emulations and file transfer protocols.
- **NETWORK LIB**
  A DLL that provides access to several Novell, Lantastic and NetBios related services.
- **VBXREF**
  The best project management and cross reference tool. Work with up to 10 projects at one time, and view 20 different forms or modules and 20 file viewers.

SHERIDAN

VBASSIST 3
Powerful design tools and new data access extensions for Rapid Application Development.

DATA WIDGETS
Design database application front ends without time consuming coding. Take full advantage of VB's data access features.

DESIGNER WIDGETS
Create apps that look, feel and act like top selling commercial packages.

3-D WIDGETS
12 custom controls to give a professional 3-D look to your application.

FARPOINT

AWARE/VBX
Build flexible front ends with full data entry controls for the Access database engine.

TAB/PRO
A design time interface for creating customised tabs which give applications a truly professional and organised quality.

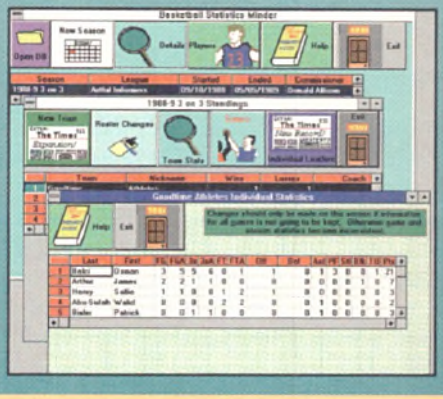
SPREAD/VBX
Over 250 properties make this the most powerful combined spreadsheet and grid control for the Access engine.

GRID/VBX
The best grid for the Access engine with 2 billion rows and columns capacity.

CRYSTAL

CRYSTAL REPORTS 4
The standard edition lets you create reports using data from the major PC databases easily and simply.

CRYSTAL REPORTS PRO 4
Pro 4 gives even more reporting functionality, a full copy of TrueGrid 2 + access to SQL databases.



HOT OFF THE PRESS

ALL THE VB TOOLS YOU'LL EVER NEED!

- 20 of the best Visual Basic programs are featured in our new colour brochure. Whether your applications are geared to database design or report generation; whether you want to add spell check, communication routines or mailing labels, project management or 3-D effects - there's something for you.
- Top products from Crystal Services, Sheridan, MicroHelp, FarPoint and INTERSOLV are explained clearly and concisely. Prices start from just £75 plus discounts for 2 or more products.

ONE-STOP SHOPPING AT ITS BEST

For more information call us on 01727 811999

CONTEMPORARY Software

Abbey View, Everard Close, St Albans AL1 2PS Tel 01727 811999 Fax 01727 848991

CONTEMPORARY SOFTWARE A DIVISION OF INTERSOLV PLC

CIRCLE NO. 039

Maths in a DLL

The Numeric Algorithms Group (NAG) has produced a DLL version of its maths library. The Numerical C DLL contains over 250 routines including functions for minimisation, ordinary differential equations, Fourier transformation, quadrature and linear algebra. It also provides routines for solutions of systems, curve and surface fitting and statistical functions. The library is priced at £495. NAG is based in Oxford and can be contacted on 01865 511245.

Get it Write

Add a complete word processor to your applications with the latest version of HighEdit. Enhancements in version 3.0 include a fully functional printer facility with print preview, RTF support, undo and support for tables. The word processor provides an IconBar, a Ruler and a Status bar. The product is shipped with both a VBX for Visual Basic and DLL interfaces. It can be used with applications built in PowerBuilder, dBASE for Windows and Visual Objects as well as Visual Basic and C/C++. According to the distributor, HighEdit has 'more scope' than Microsoft Write. HighEdit 3.0 costs £199 and is distributed in the UK by Bits Per Second on 01273 727119.

Fit and healthy

Now developers can check the health of their databases with a tool called WizRule. This is a new database auditing tool that determines errors in database files. File formats supported include Paradox, FoxPro, dBASE and Clipper. Other databases can be accessed via ODBC. WizRule apparently uses a 'unique mathematical algorithm' which allows it to determine qualitative and quantitative rules in a database. This can help to identify errors in record clusters before the data is used. WizRules costs £325 and is distributed by SoftKlone on 01628 819200.

OS/2 Pascal

Prospero Software has released a new version of its 32-bit Pascal compiler for OS/2 which now provides a full implementation of the Extended Pascal standard. The compiler includes a Presentation Manager hosted IDE called Workbench, a 32-bit symbolic debugger called Probe and the ProMake utility for building applications. Prospero is also providing a Pascal interface to the OS/2 API. Prospero extended Pascal for OS/2 is available at a special introductory price of £160. Prospero can be reached on 0181 7418531.

Sound's a bound

The penguin folk at Nu Mega have produced upgrades to two versions of the Bounds Checker debugging tool. Version 1.5 for Windows NT is now able to use the .PDB, native debug files of Visual C++. Furthermore, the new release performs validation of parameters and return code for calls to the ANSI C library and the Windows common dialog API. DLL files loaded with the `LoadLibrary` call can also be checked to verify that they correctly free memory and resource that they allocate.

Version 2.5 of Bounds Checker for Windows is the other upgrade. This tool for Win16 debugging now provides validation checks on the 1700 API calls of MFC. Parameters are checked on entry, return status on exit. Bounds Checker's TView tool keeps a log of all Windows events. Calls to the Windows API and MFC can subsequently be browsed by developers.

For developers of VBXs, Nu Mega has provided a facility to detect memory leakage, resource leakage and heap and data corruption either within C/C++ or within Visual Basic. It will also check the VBX's conformance to the VBX programming API. Both versions of Bounds Checker are distributed in the UK by System Science (0171833 1022).

Cadre moves into testing and debugging

Through a joint agreement with Silicon Graphics, Cadre Technologies has produced a suite of platform independent C and C++ development tools. ObjectTeam/ProDev debugger is a C and C++ debugger which Cadre says offers 'complete syntactic understanding' of both languages and 'allows programmers to debug large, complex applications...' Allegedly, it can handle over 100 MB of source code and executables. Multiple views of the application are available and these can be dynamically updated during program execution. As expected, there is tight integration with other tools in the suite.

ObjectTeam/ProDev static analyser produces a graphical representation of code allowing developers to navigate through a program's structure. Multiple views are available including one for actual text, call tree, C++ class libraries and file dependencies. Two modes of operation are available. Fast scanner enables the developer to see the structure quickly. There is also a more detailed parser mode.

Integrated with the static analyser is ObjectTeam/ProDev C++ browser for navigating, analysing and understanding complex class hierarchies. Hierarchies can be viewed by inheritance, friend or containment relationships, or by call reference or method implementation.

The ObjectTeam/ProDev product suite is available on Solaris 2.03 and HP/UX. Cadre is on 01344 300003.

Test for the rest...

You too can produce bug free code like Microsoft's. Never mind the little glitches in VB or Excel or the general protection faults in Windows, MS Test 3.0a, used internally by Microsoft's own developers, is now available. As well as Windows 3.x and Windows NT, the new release supports the Windows 95 look and feel including task bars, sliders, list views and tree views. The front-end itself provides colour syntax highlighting movable and dockable toolbars, popup menus and context sensitive help.

The SmartEvent and SmartQuery features allow developers to simulate events, extract information and record editable scripts which Microsoft says are self-documenting. There is an autosynchronize facility which causes the application under examination to wait up to five seconds to finds controls unavailable due to delays caused by, for example, network or client/server processing. On the client/server front MS Test offers redistributable scripts to up to 250 machines for regression coverage. This feature can also be used to test client workstations in a multimode client/server application. There is an improved bitmap comparison tool with zooming, user-defined image names, toolbar and support for all Windows' video configurations.

NT support extends to RISC based platforms. That means MIPS, Alpha AXP and a new PowerPC version. Microsoft Test 3.0a costs £495. Developers with Test 1.0 or Test 2.0 can upgrade for £149. Microsoft is on 01734 270000.

Tired of Juggling Your Team Development?

Want more info?
2 free White Papers?
PVCS demo diskette?
.....call.....
01727 812812
EXT. 255

Team development adds its own set of challenges to the development process–communicating the right information at the right time to the right team members. How do we keep all the configuration management balls in the air at the same time? Can we control access to protect the code? Did all the changes make it into the release? Will applications build properly?

PVCS–The Standard for SCM

INTERSOLV PVCS adapts to your environment with automated software configuration management (SCM) for team development. The standard for SCM on the LAN, PVCS lets you and your team work smarter, increase productivity, and accelerate delivery. PVCS is used by more developers than all other LAN-based SCM tools combined. And, PVCS covers more operating systems, development environments, and object types than any other SCM tool.

So stop juggling and focus on what you do best–development. We'll show you how. Call us today at **01727 812812, Ext. 255.**

INTERSOLV

CIRCLE NO. 040

Announcing a Major New Upgrade of CommLib

Greenleaf CommLib 5.0

Over 350 public functions and macros, providing powerful tools for all aspects of asynchronous communications programming for Microsoft Windows 3.1 and MS-DOS.

Now we provide even more of the power and features you've asked for!

Some features of CommLib 5.0:

- 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
- NEW! Support for WIN-32 API and 32 bit DOS extenders
- NEW! Support for Windows NT
- XMODEM, YMODEM, ZMODEM, Kermit, ASCII, many options
- NEW! Support for Compuserve B+
- Unlimited number of ports for ISA, EISA, MCA
- Modem controls, keyboard, screen, handshaking and much more!
- Supports all popular C and C++ compilers

What's included:

- Free source code
- All memory models supplied
- Examples that compile and run included for every function
- Professional documentation
- Free technical support and access to our BBS

Greenleaf CommLib 5.0

Only £259 excluding delivery and VAT

No royalties

90-day money back guarantee

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

0566 86037

FAX 0566 86147

BBS 0566 86925

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

Painless Data Compression and Archiving

Introducing Greenleaf ArchiveLib

Experience the power to compress and store data from anywhere, to anywhere in your C/C++ applications.

Until now your development choices for data compression and archiving in C or C++ applications (and other languages too!) were severely limited. Professionals – don't be stymied any longer!

ArchiveLib includes a fast, efficient compression engine and architecture for addition of other compression methods. You control source and destination (file, memory, port, etc.) of all data. Complete control over format of archives.

Many extras for Windows programmers.

ArchiveLib Major Features:

- 100% C++ class library plus C interface
- Supports Microsoft Windows 3.1, MS-DOS, and Watcom 32-bit extended DOS
- Portable data compression is fast and efficient
- Platform and language independence
- Coded for Win32 – smooth path to 32-bit Windows
- Extensible through plug-in compression classes
- Five compression levels – auto or manual selection
- Versatile archiving – files, memory objects
- File information automatically included in archive
- Release and debug library versions included
- Tagged objects flag allocation or pointer errors
- Extensive Windows support includes compression/archiving progress meters and message options
- Wildcard expander class is also useful for general purpose application
- 25 classes – over 100 functions in C and C++

What's included

- Free source code
- All memory models supplied
- Examples that compile and run included for every function
- Professional documentation
- Free technical support and access to our BBS

Greenleaf ArchiveLib

Only £229 excluding delivery and VAT

No royalties

90-day money back guarantee

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

0566 86037

FAX 0566 86147

BBS 0566 86925

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

SOAPBOX

There is certainly a case for a policy of 'try before you buy' in retail software. But how can we protect our investments from being plundered by software pirates asks **Melanie Welsh.**

Software developers often seem to spend a great deal of time discussing the various merits of the tools and platforms with which they create their wares. Obviously, this is an important issue. After all, putting your shirt on the 'wrong horse' can have disastrous financial effects. But there are also other issues which can and will effect the economic situation of the software development world. The issue of copyright is one of them.

In 1987 Ryoichi Mori, head of the Japan Electronics Industry Development Association pioneered the idea of 'superdistribution' - the nemesis of copyright. The basic premise of copyright law is that anything worth copying illegally, ie stealing, comes in some kind of physical format. In UK law it is impossible to copyright an idea. Superdistribution, however, works on the fundamental principle that preventing people from copying information is time-consuming, expensive and ultimately impossible to enforce universally. Instead Mori and other Superdistribution advocates suggest that we encourage free distribution of our products and concentrate our time on monitoring their usage. While making software count how many times it has been used is easy, counting how many times it's been copied is a lot trickier.

So how do developers gain any financial benefit from this initial burst of generosity? The answer lies in the hardware. Superdistribution software could only work on computers that were equipped to register and pay for that software on a usage basis. Mori's prototype is a silicon chip that plugs into a coprocessor slot. Because the hardware is relatively uncomplicated it should also be cheap to manufacture, which is of course, a key consideration. Data on each computer owner's software usage could then be sent via modem, smart card or even good old-fashioned floppy, for billing. The advantage over more traditional dongle-base protection schemes is that, should the chip become a standard component of personal computer hardware, every software manufacturer would be in a position to take advantage of its capabilities.

Shareware, on first appraisal, sounds quite similar to Superdistribution. The developers of shareware encourage users to hand out free copies, asking them to send money if they use the software. Unfortunately, shareware developers have no way of ensuring that the people who use their programs will pay for them. And, although millions of pounds in annual revenue are made from this form of software, it is estimated that, at the most, only 5% of the people who regularly use shareware programs will actually cough up the readies. Superdistribution also has the potential to cater for the dawning age of the 'Information Superhighway'. Any other copyrighted information, books, songs, quicktime movies could all be used and paid for through Superdistribution.

Or, we could just hold on, in belligerent determination, to the principle of copyright. There seems to be no reason why we couldn't prolong the present situation indefinitely.

But keeping copyright may not actually be the best financial decision. Superdistribution has the potential to encourage users to buy more software than they do now. Because of the present copyright laws, software demos are often less than satisfactory, which can deter people from buying. Many users would like to use some software packages every now and then, but because they can't justify the cost for such infrequent use they don't buy anything at all. Superdistribution would mean that these users could get hold of the software they wanted without resorting to using pirated copies. The increased revenue for software companies from thousands or millions of 'one-time' users could be considerable.

And, although there is the temptation to think that most users would still persist in pirating software, on second analysis this seems unlikely. Most consumers are very fond of convenience and quality. For example, although pirating music onto audio cassette is considerably cheaper than paying for a legal copy, the record industry is still going strong, because while home taping may save you money, it just isn't that convenient.

The software industry could be like this too. At the moment, packages that the home-user would like to buy can cost as much, if not more, than the computer itself. Individual users feel far more justified in appropriating such costly software than they would a £12 CD. And, telling them that the software industry loses billions of pounds each year in pirated copies only makes them think of the many more billions that are made.

If users were to buy their software on a 'metered' basis, it is unlikely that many would persist in using illegal copies. The benefits of full documentation and technical support, not to mention the fact that they would no longer be committing a crime every time they turned the computer on, would be too tempting for the majority to resist.

As the Internet gains in popularity, and ways of copying software illegally become more widespread, we have to give this issue some serious consideration. Superdistribution obviously has its setbacks, but it puts forward a number of interesting considerations. Naturally, we want to protect hard-earned investments in software development, but equally, we need to ensure that the end product is exposed to the maximum number of people. There should be no reason why both these aims cannot be achieved together. And who knows, maybe there'll be some money in it as well... ■

Phar and

After years of supplying linkers and assemblers to embedded systems developers, Phar Lap takes the plunge with a complete suite of tools and an all-new 32-bit kernel. **Cliff Saran** gets his mitts on a copy...

In the past, embedded systems developers ran their development tools on the PC and their applications on custom hardware. But a recent trend in the industry has seen a shift from more traditional target platforms towards PC-based hardware. The reason for this shift is that with over 100 million PCs in the world, there is no shortage of low cost, quality development tools available. And of course, PC-based hardware is relatively cheap. The situation has come to the fore recently as both Intel and AMD have announced embedded versions of their 386 microprocessor families. These are the Intel 386EX and the AMD386Em.

For application areas that require 32-bit code, developers have traditionally taken the 68k route along with the associated cost of specialised, embedded systems development tools. Often they would purchase the embedded operating system from one supplier, obtain a compiler from someone else and the debugger and monitor tools from yet another supplier. Inevitably, problems with integration do arise. However, the embedded systems developer must now speak to several suppliers to get an answer. Now, with the processors from Intel and AMD it is possible for them to build and even test systems on their host PCs. TNT Embedded ToolSuite is Phar Lap's 'one stop' embedded systems package, designed to take advantage of this emerging niche in the embedded system tools market.

Phar Lap aims with ToolSuite to provide all the ancillary tools necessary to write software for embedded systems using standard, off-the-shelf PC-based compilers and debuggers. The review software worked in conjunction with the 32-bit compilers from Borland, Microsoft and MetaWare.

Not only are the tools to build embedded systems provided, but also the embedded operating system itself. The company's decision not to redistribute an existing embedded operating system at first appears somewhat misguided. After all, there are a handful of well known suppliers of embedded systems. But Phar Lap is not one of them. More often than not the name 'Phar Lap' is synonymous with DOS extenders.

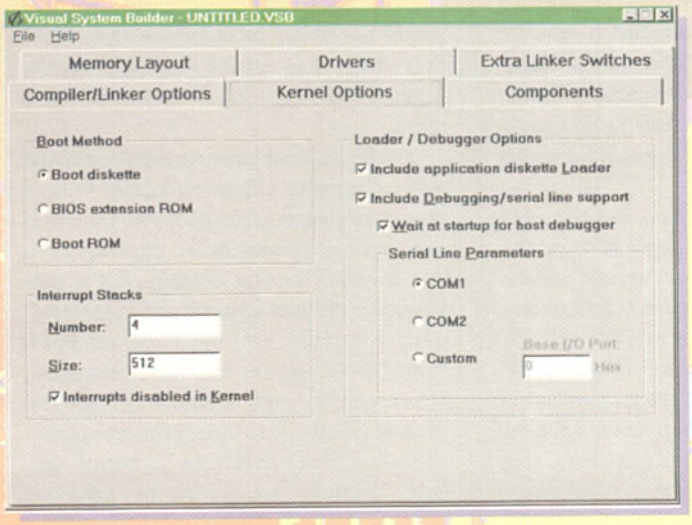


Figure 1 - Changing kernel setting with Visual System Builder

A head for embed

However the company has, by its own admission, been in the embedded systems market from the late eighties. Previously it offered two tools for embedded systems developers, LinkLoc and 386|ASM. LinkLoc performs linking, locating and output formatting in one step. It supports both Microsoft and MetaWare compilers and is compatible with several in-circuit emulators, PROM programmers and embedded debuggers. 386|ASM is 32bit cross assembler. Both these tools find their way into the ToolSuite product.

Phar Lap intended ToolSuite to contain all the support tools apart from actual compilers and debuggers, necessary to write embedded systems software. Those tools include a 32-bit embedded system. The advantage to embedded systems developers is that Phar Lap should be able to tackle any configuration problems with the software. The drawback is that what happens if a developer doesn't wish to use the embedded operating system shipped with ToolSuite?

Sworn to compile

For the purpose of this review I decided to run Visual C++ 2.0 with ToolSuite, hosted under Windows 95. But there should be no reason why the software should not run under Windows NT either. This is the easiest way to get up and running. However, Phar Lap provides a batch program called BINDVC20.BAT for rehosting the 32-bit Visual C++ tools under Windows 3.1 and DOS. The compiler, linker, librarian, resource converter, CodeView packer utility, NMAKE and the browser information maintenance utility can be rehosted this way. All that is lost is the ability for Visual C++ to pre-compiler headers and incremental linking.

It isn't even necessary to install Visual C++ under NT or Windows 95. Phar Lap's GETVC20.BAT utility installs the compiler from DOS. The only drawback from being

away

DOS-hosted is that Visual C++ looses its ability to run from CD-ROM. Given that a full installation of Visual C++ gobbles up acres of hard disk space, DOS-hosting may not be such a good idea, especially if disk space is at a premium. Still, the GETVC20.BAT installation script can install selectively. Developers can choose whether they want MFC, help file or samples along with the basic compiler tools.

With the compiler and ToolSuite installed, all that's left to do is set up `path`, `include`, `lib` and `helpfiles` DOS environment variables. Less work is needed to get going with Borland C++ as it is not necessary to rehost the 32-bit compiler. The BCC32.CFG file needs to be updated with the include path for ToolSuite files and `lib` and `path` DOS environment variable must be set up. For MetaWare's High C/C++, BINDHC32.BAT is provided for rehosting the NT-only compiler to DOS. Then `path`, `ipath`, `lib` and `helpfiles` DOS environment variables should be set.

Fully loaded

While Phar Lap intends ToolSuite to be used to create embedded systems, developers do not need custom hardware in order to run embedded applications. Out of the box, ToolSuite comes with a 1.5 metre LapLink null modem cables for connecting development and target PC together. Of course, in my office, the two machines were at opposite ends of the room. But that's a minor niggle.

As it ships, the tools are configured to use COM1 on both host and target. Although the source for the embedded operating system is provided, in most cases it is not necessary to build the operating system in order to run application code. The operating system is also supplied on a DOS-readable diskette. DISKCOPY can be used to create backups. Even if the configuration of the operating systems must be changed, often there is no need to rebuild it. Phar Lap provides a command line configuration utility called CFIGKERN for changing such parameters as the COM port used on the target system or whether to enable kernel interrupts.

The right signal

Once the target machine has been booted with this diskette, the operating system comes up in a mode where it waits for a signal from the host PC. In a typical set up, the source code for the application would have first been compiled with Visual C++ (or Borland C++ or MetaWare High C/C++) then linked with LinkLoc to create an executable. Now this executable is targeted at Phar Lap's embedded operating system. It should not be run under DOS or Windows. The RUNEMB utility is used to upload the application into the target machine and signal the operating system to run the code. RUNEMB presents a window with a thermometer displaying the amount of code that has been uploaded, which is useful when applications start growing in size. It will attempt to connect to the target at 115,200 then 57,600, then 38,400 down to 9600 bps until a successful connection is made.

A stub DOS executable is bound with the embedded application to prevent anyone from attempting to run it under DOS. If they try, it simply prints a message urging them to run RUNEMB with the name of the executable as the first command line parameter. This is similar to the behaviour of a Windows executable when the user tries to run it under DOS.

Nit-picking

A live connection to the host means that developers can debug their code remotely. It is pretty obvious that, rather than ship a remote debugger with ToolSuite, Phar Lap would prefer to support the existing debugger supplied with the compiler. In anticipation of the inevitable question, to get around the fact that neither Borland's Turbo Debugger, nor CodeView, support debugging over a serial cable, it was necessary for Phar Lap to provide some kind of 'wrapper'. The one for CodeView is CVEMB. As in RUNEMB, code is first uploaded to the target system. On completion, all features of CodeView are available to the developer. It is possible to single-step, step over functions, add break points and watchpoints and display the call-return stack.

Boot-iful

Uploading the application software from a host is an ideal way to make rapid changes to the code without the time consuming chore of erasing and blowing eproms. Of course, unless the final system will involve a host and target PC, at a certain stage in the project, the developer will decide to run the code in isolation on the target system.

There are two choices here. Since Phar Lap's embedded operating system can be loaded from diskette, it is possible to configure it to load the application code once the operating system itself has finished loading. That way, both system and software can be installed an run in one go, from a single diskette. In this case, the application will run from RAM.

The alternative is to blow a PROM with the code. The PROM will contain both the application code and the operating system itself which are linked together with LinkLoc. A further complexity with creating a PROM is that there are two ways to boot the embedded kernel.

On hardware equipped with a PC BIOS, the embedded kernel must be located either between C0000h to DFFFFh or at E0000h. Power On self Test (POST) will search for a three byte signature code. If found, an `int 19h` handler will be installed. When POST is complete control is passed to `int 19h` which boots the TNT Embedded Kernel. While linking the embedded kernel to achieve this can be rather tricky, Phar Lap includes LinkLoc linker response files for the Microsoft, Borland and MetaWare compilers. LinkLoc chucks out a file in the Microsoft Portable

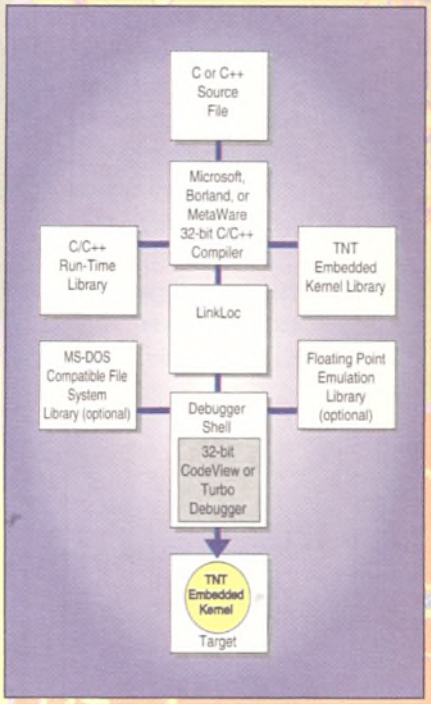


Figure 2 - The whole of ToolSuite is centred around LinkLoc

Navigate to Client/Server the Fast and Secure Way

Btrieve 6 offers a fast and secure route to client/server, avoiding all the traps of a traditional migration. With Btrieve 6, you maintain total control over relational structures and distributed data routines while enjoying the benefits offered by client/server processing.

Take the navigational path to client/server

Navigational client/server allows you to custom design relational structures and maximize performance with directional controls to retrieve, update, insert and delete distributed data. With Btrieve 6, you will have the transaction processing muscle to build multi-gigabyte database servers supporting hundreds of users with sub-second response times. And Btrieve 6 supports the major server operating systems such as NetWare and Windows NT Server as well as the major client operating systems.

Move from standalone to client/server without recoding

With Btrieve 6, you can develop on a laptop using Btrieve 6 workstation engine, then deploy the application on a network using a Btrieve 6 server engine... all without changing your application code !

Avoid retraining problems

Btrieve 6 directional controls integrate with existing application code so you can selectively upgrade your current applications to client/server. Using your 3, 4 and 5 GL tools, merely replace your data management code with Btrieve 6. This way your users can continue working with familiar applications while gaining the benefits offered by client/server.

Add SQL along the way

Btrieve integrates with Scalable SQL, our award-winning relational database. Since both are based on our Microkernel Database Engine, SQL applications work in unison with Btrieve applications - each having current access to all data. Now you have the freedom to write new applications in Btrieve or Scalable SQL and the trip to client/server will be much more manageable.

IT SIMPLY WORKS

BTRIEVE TECHNOLOGIES

Btrieve Technologies, Inc. - 8834 Capital of Texas HighWay North - Austin, TX 78759 - USA
Tél. (512) 794 17 19 - Fax (512) 794 17 78.
CompuServe (GO BTRIEVE).

Hurry! The Btrieve 6 Client/Server Developer Bundle is available for a limited time! Please send me :

- ☐ More information on the Btrieve 6 navigational client/server database and the Btrieve 6 Client/Server Developer Bundle.
- ☐ More information on the Scalable SQL relational database engine
- ☐ A White paper on Btrieve Technologies Microkernel Database Engine
- ☐ A referral to a local distributor or reseller

yes It runs with NetWare

Microsoft SOLUTION PROVIDER

For more information, mail or fax this coupon to :
Btrieve Technologies Inc.
8834 Capital of Texas HighWay North
Austin TX 78759 USA
Tél. : (512) 794 17 19 - Fax : (512) 794 17 18
CompuServe (GO BTRIEVE)

Name : _____
Company : _____
Street Address : _____
City/Province : _____
Country : _____ Postal Code : _____
Fax : _____ Phone : _____

Executable file format. This comprises a real mode stub program and a protected mode executable. The stub is the TNT embedded kernel; the protected mode program is the application code.

Now a custom 386 board will not need a PC BIOS. On power-on the processor jumps to address FF00:F000 and begins executing the code there. By blowing the TNT embedded kernel at this address in the PROM, the kernel will boot immediately after power-on. Obviously, with this method, the chip set of the motherboard must be initialised by the kernel itself.

Kernel cosmetics

Although I haven't mentioned it yet, it is obvious that since the kernel can load from diskette, it supports a local filing system. Of course, the system under development may not need a local filing system, so this feature can be removed from the kernel. The same is true of the floating point emulation library. These are both switches which can be toggled using CFIGKERN.

The other way of controlling the configuration of the kernel is linking or removing optional modules. It is possible to add support for timer, keyboard input and console output using this method. Also, the modules are self-contained and thus can be replaced by alternative code with the same external interface. Modules can be created for controlling custom hardware. This is the only way a developer can create a custom device driver. Both 16- and 32-bit APIs are available. Figure 3 lists all the 32-bit drivers and functions that can be replaced in this manner. But why have 16-bit drivers? Well, an application may need to perform a read from the keyboard. This involves a call to the PC BIOS which is, of course, 16-bit code. At a later date, the developer could choose to write a 32-bit keyboard handler and replace the 16-bit call in the kernel. Figure 4 shows the list of 16-bit functions available.

Run for cover

The good news for developers is that pretty much all of the C Runtime Library is supported by TNT Embedded Kernel apart from those calls, such as `spawn`, that would require features not built into the kernel. This means that developers can prototype application software easily on their development machines. In fact, ToolSuite includes a copy of Phar Lap DOS-Extender for this very purpose. DOS-Extender provides a flat 32-bit address space with a similar environment to the TNT Embedded Kernel. However, Kernel-specific API calls as listed in Figure 3 to 5, are unavailable. Even so, Phar Lap developed its sample terminal application in DOS-Extender first, before porting it to TNT Embedded System. Stub code can be provided for hardware dependent aspects of the application which can be used to emulate the desired effect.

EtsCustomGetKeyboardDriver	Initialise keyboard driver
EtsCustomGetScreenDriver	Initialise screen driver
EtsCustomGetTimerDriver	Initialise timer driver
EtsCustomEmulInit	Initialise floating point emulator
EtsCustomExitProcess	Terminate the process
EtsCustomFSDInit	Initialise target file system
EtsCustomGetCommandLine	Get host's command line
EtsCustomGetEnvStrings	Get host's environment strings
EtsCustomGetMemPool	Return available memory ranges

Figure 3 - Replaceable 32-bit drivers and functions

EkCustomClearNMI	Clear source of non-maskable interrupt
EkCustomCommBreakOff	Turn off line break
EkCustomCommBreakOn	Turn on line break
EkCustomCommClearStatus	Clear latched status indicators
EkCustomCommGetStatus	Get communications device status
EkCustomCommInitialize	Initialise host communications driver
EkCustomCommReadCharacter	Read a message character
EkCustomCommSetSpeed	Set the communications speed
EkCustomCommStartReceive	Set up for start of message reception
EkCustomCommStartSend	Set up for start of message transmission
EkCustomCommWriteCharacter	Write a message character
EkCustomDeviceInit	Initialise other target hardware
EkCustomGetScreenCursor	Get cursor position on target screen
EkCustomMaskNMI Mask	non-maskable interrupt
EkCustomPutChar	Write character to target display adapter
EkCustomRealModeInit	Complete real mode initialisation
EkCustomRunMode	Set kernel run mode
EkCustomSetSceenCursor	Set cursor position on target screen
EkCustomSystemInit	Initialise system-critical hardware

Figure 4 - 16-bit functions that can be replaced

Suite for all

As mentioned above ToolSuite includes a copy of Phar Lap DOS-Extender. There is also the 386|ASM cross assembler and a copy of FrontRunner, Phar Lap's DOS command shell for Windows. While the tools are mainly command line, there is one that runs under Windows. This is a project configuration tool called Visual System Builder. With it, developers can choose which kernel options they need; add or remove optional modules and set up linker and compiler switches. In one place developers can select which compiler they are using, whether to build for ROM or boot disk and specify memory options such as size of the stack. Visual System Builder then creates two linker response files, one specifies the way the kernel should be built. The other is for the application code. Figure 1 is a screenshot of Visual System Builder.

Books, books and more books

ToolSuite comes with 13 printed manuals. Of these, the most important one is the 290 page *TNT Embedded Kernel Developer's Guide* which describes the ins and outs of using the kernel. It covers configurations for the Microsoft, Borland and MetaWare compilers and illustrates sample debugging sessions in CodeView and Turbo Debugger. There are also chapters on writing interrupt and exception handlers and a section on customising the kernel. The API is described in detail and there are several pretty comprehensive sample applications listed in the manual. A separate 255 manual is provided for LinkLoc. The is also a short 70 page guide to using Visual System Builder. The remainder of the tools are covered into the 88 page *ToolSuite Utilities Reference Manual.*

There are three manual on TNT DOS-Extender. These are the *C/C++ User Guide*, a *Libraries and System Call Reference* and the *TNT DOS-Extender Reference Manual.* Other manuals include *CodeView Reference*, *Windows Interface Guide* and a user guide to Phar Lap Front Runner.

Installation

ToolSuite is shipped on nine 1.44 MB diskettes and requires just over 9 MB of disk space. It also includes a separate diskette

with a bootable version of the TNT Embedded Kernel. For the host machine, system requirements are that of the compiler being used. In the case of Visual C++, 8 MB is sufficient. But up to 160 MB of hard disk space may be required if a complete installation of the development system is performed. Target processor is 386SX or above. The kernel itself requires about 25 KB of memory depending on which configuration option is selected. For instance more will be required to add floating point emulation or support for local filing system.

How valuable?

Development tools for embedded systems are notoriously expensive. In some cases, they are not as easy to use as off-the-shelf PC-based development tools. ToolSuite is attractive because it allows developers to work with familiar mass-produced compilers and debuggers which cost very little compared to embedded versions of such tools.

What happens if the developer doesn't wish to use this embedded OS?

EtsCallExitHandler	Call registered Exit function
EtsExitProcess	Terminate program
EtsGetKernelRunMode	Returns run mode configured in kernel
EtsGetSystemInfor	Get configuration information
EtsGetVsbVarsPointer	Get pointer to kernel VSB_VARS structure
EtsQueryFileHandle	Query file handle
EtsRegisterCallback	Register device driver callback
EtsSelectConsole	Select host or local console
EtsSelectFieldSystem	Select host or local file system

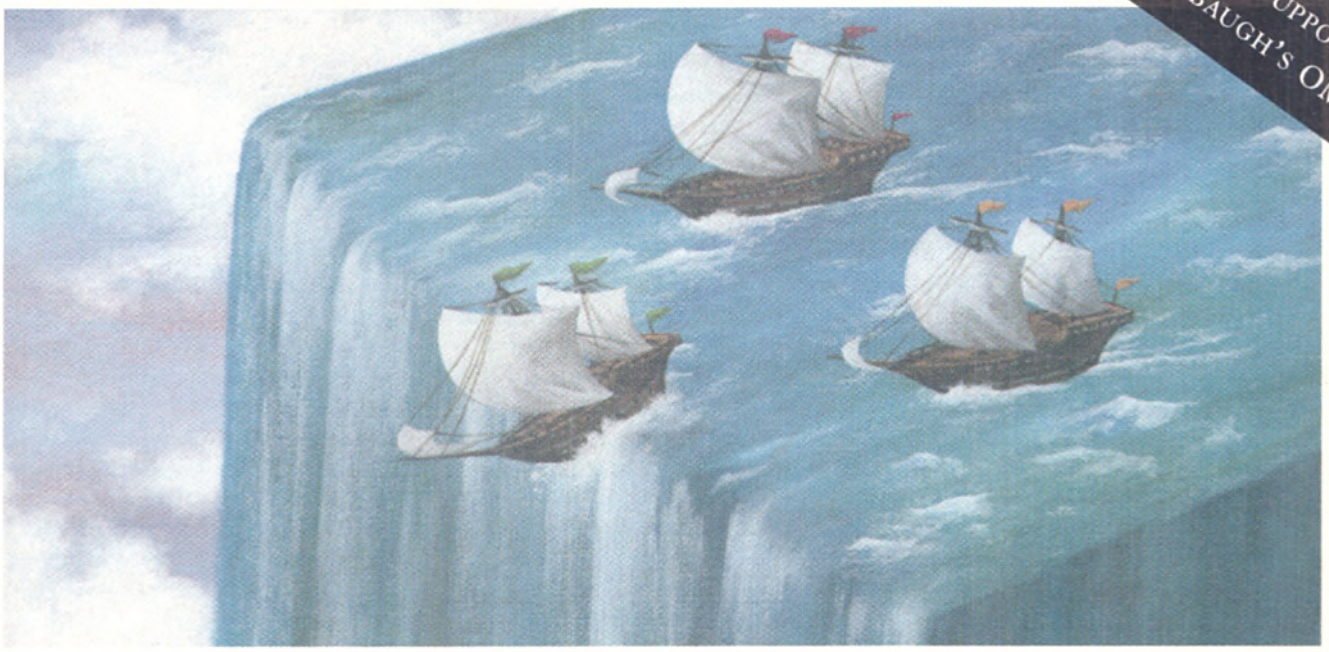
Figure 5 - System calls for TNT Embedded Kernel

Gains in productivity are possible thanks to the ability for developers to prototype systems on standard PC hardware. By including DOS-Extender in the product, Phar Lap enables developers to create 32-bit prototype applications which can be tested on the host development machine itself, rather than target hardware. Furthermore, since the TNT Embedded Kernel is designed to run on top of a standard PC BIOS, the cost of creating complete embedded systems is drastically reduced since developers can base their applications on relatively low-cost PC hardware.

ToolSuite is not Phar Lap's first foray into the world of embedded systems. LinkLoc and 386|ASM have been available for years. Even the TNT Embedded Kernel, which is a new product, owes much to the technology behind Phar Lap's popular family of DOS extenders. For Phar Lap, the only worry, is that in order for the product to succeed as a whole, it must convince embedded systems developers that there is a need for yet another embedded system. ■

Phar Lap Embedded ToolSuite is distributed in the UK by System Science (0171 8331022) and Grey Matter (01364 654100). System Science's price is £1,995.

System Architect NOW Supports Rumbaugh's OMT



Regarding The Matter Of Low Price Meaning Low Functionality

Kiss Another Myth Goodbye.

The Myth

When application development tools first gained acceptance, the products that offered real power were too costly for many companies. As a consequence, people believed if a tool was low priced, it wasn't powerful.

The Reality

When Popkin released the original version of System Architect in 1988, that particular myth was laid to rest. It was the first product to offer high performance at a low price. To this day, SA is the performance leader in it's category and yet, the price remains low. The savings are especially evident when outfitting a project team. For a typical 10-seat configuration, SA costs just a fraction of other repository-based tools working under Windows® or OS/2®.

Simplifying A Complex World

SA makes your job easier by supporting the major structured, BPR (IDEF) and OO methodologies. You get the benefits of the best repository-based data modelling tool, plus the functionality of a comprehensive application development environment. SA's repository is extensible and customisable, so adding new objects and properties is a simple task.

Because SA Reverse Data Engineer is ODBC-compatible, you'll be able to reverse engineer a wide range of DBMS' and RDBMS' including Oracle, SYBASE, Informix and many others.

Using SA Schema Generator, you'll be able to generate triggers and schema for the most popular RDBMS'. Plus SA Schema Generator 2.0 has a new user interface, making it easier to use than ever.

50,000 People Can't Be Wrong

That's the number of satisfied System Architect users worldwide. And it's growing every day.

Don't just take our word for it...

"System Architect from Popkin Software is evolving at a phenomenal rate. This product is without a doubt the best value for developers on limited budgets, or for that matter perhaps on any budget... Comparing System Architect to other CASE tools, it appeared to be the easiest to customise and learn."

- Application Development Trends Magazine, June 1994

SYSTEM ARCHITECT™

Popkin Software & Systems Ltd
118-120 Warwick Street Royal Leamington Spa Warwickshire England CV32 4QY
Tel: (01926) 450858 • Fax: (01926) 311833

CIRCLE NO. 044

Microsoft Windows

PowerSoft Code Partner

Going to work

The computer market is big but still relatively small compared to that for the electrical and electronic appliances. **David Mery** explains how three major software companies are trying to create the standard that will give them the best part of this market.

You arrive at the office, switch on your PC, there's a message on your screen coming from the fax machine saying that you received two faxes. One of them asks you to send 10 copies of a large report to another office of your company but you also need to make five copies for your local office. So you send the document to the local photocopier and instruct it to do the job. After making five copies, the photocopier then, on its own, sends the document to the other office's photocopier to process the remainder. Maybe not today, not tomorrow but soon it will all be reality. Most of the elements needed to make it happen already exist. A network connection and some standard to communicate between the office equipment is all that is required.

Most of us interact daily with many of these *intelligent* devices such as computers, fax, photocopiers at work and washing machines or VCRs at home. But beside computers most of these are isolated devices. There is a growing concern in the industry that equipped with the right standard(s) such equipment could be networked and interact together bringing a whole new range of functions. They would be able to share information and even control each other. They would also be able to give status information in real time to any user on the network. The potential size of this market is immense. According to Novell, in 1994, there were 'approximately 400 million intelligent devices in use worldwide, or about four intelligent devices for every personal computer.' Several standards are emerging and a battle between them will probably happen in this market segment.

Microsoft was first in June 1993 with an architecture it called Microsoft At Work. Novell came in with NEST (Novell Embedded Systems Technology), which has just been released in its first implementation. Then there's IBM. Last February, a Japanese newspaper reported a leak about an unannounced standard coming from 15 companies including IBM, Toshiba, Ricoh and ICL.

As you would expect, these three architectures differ in their respective philosophy. No surprise with Microsoft: it's Windows everywhere through a modified Windows kernel which runs in the embedded system. Novell's system is hardware and operating system independent but is based on, surprise, surprise, NetWare 4. NEST 1.0 includes only an IPX/SPX stack. IBM *et al* have opted to define an interface with no imposed code. So as long as OEMs respect this API they are free to develop what ever code they want.

For At Work and NEST, the list of licensees probably do not reflect a real commitment from all the companies on the list

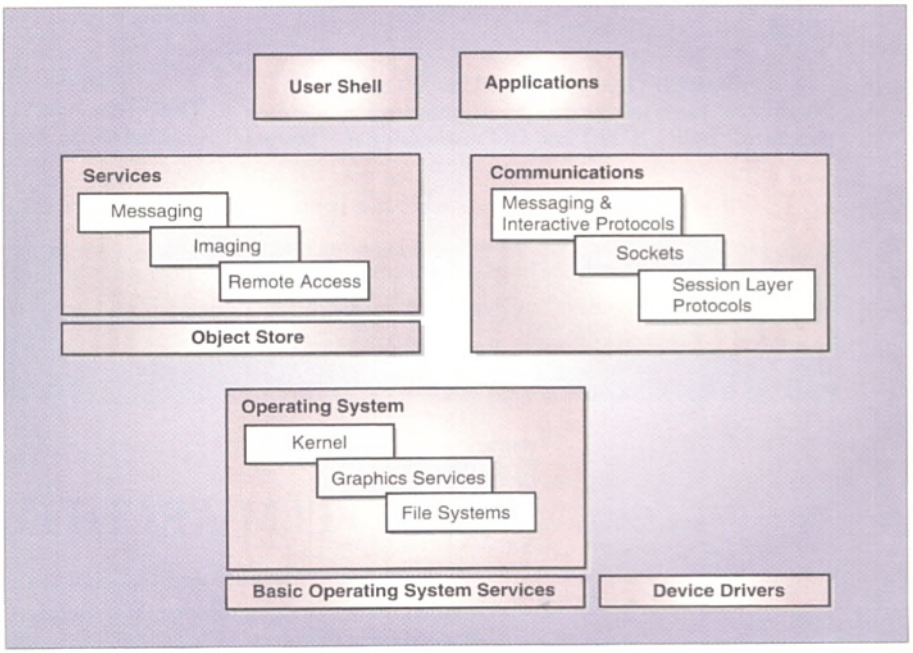


Figure 1 - Microsoft At Work architecture (Source Microsoft)

No Other Client/Server Tool Manages Business Change and Complexity Like Magic

Integrating the Future

Managing the Present

Leveraging the Past

1st Place Software Developers Challenge

Sponsored by EXE

The development path you choose determines your company's success. With business practices changing and business environments being reengineered, you need client *and* server application development that stands the test of time. And that can only mean Magic. Magic leverages your legacy IT resources to provide an economical migration path to client/server. It manages your present development needs with an innovative programming paradigm that ensures extremely rapid application delivery and easy maintenance. And Magic integrates the future with fully scalable open architecture that takes applications from tactical to strategic - incorporating new platforms, DBMSs, multiple client/server models and application partitioning scenarios.

The secret of Magic's success is the Magic Engine, an advanced repository of sophisticated application functionality. Driven by a high level, table-driven interface, the Magic Engine lets the developer concentrate on business rules and logic instead of low-level repetitive tasks.

Isn't it time you put Magic to work for your business? For more information and to attend a free Magic seminar, call:

Tel 01344 303045

MAGIC

Client and Server Development for Managing Change and Complexity

U.S. Tel: (714) 250-1718, Fax: (714) 250-7404 France Tel: (1) 47 73 90 90, Fax: (1) 47 76 44 36 The Netherlands Tel: 3405 71888, Fax: 3405 71766 U.K. Tel: 0344 303045, Fax: 0344 303024
Germany Tel: (089) 74129-135, Fax: (089) 74120-102 International Tel: (972-3) 538 9292, Fax: (972-3) 538 9333

CIRCLE NO. 045

Dongles? Buttons? Plugs?

Get Serious About Software Protection.

When it comes to protecting your software, you need real solutions - not gadgets. You need **HASP**® - *the Professional Software Protection System.*

Quick & Easy to Implement
Getting started with HASP is a snap. Interfaces are available for most programming languages, and a variety of installation options are supplied to meet your needs and schedule.

Technology That's Tough to Beat
ASIC technology, sophisticated software and random encryption algorithms are just a few reasons your products are safer when protected by HASP.

Developed with Users in Mind
Transparency. Compatibility. Connectivity. HASP was developed to perform unnoticed. Whether installed on a PC, Mac or network, your end-users will never know it's there.

Get Serious-Try it for Yourself
Find out why HASP is the fastest growing form of protection in the industry. Order your HASP Developer's Kit today.

01753 622266

ALADDIN

The Professional's Choice

United Kingdom	**Aladdin Knowledge Systems UK Ltd** 16a St. Leonard's Road, Windsor, Berks, SL4 3BU, UK Tel: 01753 622266, Fax: 01753 622262
North America	**Aladdin Software Security Inc.** Tel: (800) 223 4277, (212) 564 5678 Fax: (212) 564 3377 E-mail: sales@hasp.com
Intl Office	**Aladdin Knowledge Systems Ltd.** Tel: 972-3-537 5795, Fax: 972-3-537 5796 E-mail: aladdin@aladdin.co.il
France	**Aladdin France SA** Tel: 1 40 85 98 85, Fax: 1 41 21 90 56

CIRCLE NO. 046

member of spa

yes It runs with NetWare

Available and Ready for OS/2

Microsoft Windows Compatible

■ **Australia** Conlab 3 8985685 ■ **Benelux** Aladdin Benelux 08894 19777 ■ **Czech** Atlas 2 766085 ■ **Chile** Micrologica 2 222 1388
■ **Denmark** Berendsen 39 577300 ■ **Egypt** Zeineldein 2 3604632 ■ **Finland** ID-Systems 0 870 3520 ■ **Germany** CSS 201 278804
■ **Greece** Unibrain 1 6856320 ■ **Italy** Partner Data 2 26147380 ■ **Japan** Athena, 3 58 213284 ■ **Korea** Dae-A 2 848 4481 ■ **Mexico** SiSoft 5 5439770
■ **New Zealand** Training , 4 5666014 ■ **Poland** Systherm 61 480273 ■ **Portugal** Futurmatica 1 4116269 ■ **South Africa** D Le Roux, 11 886 4704
■ **Spain** PC Hardware, 3 4493193 ■ **Switzerland** Opag 61 7169222 ■ **Taiwan** Teco 2 555 9676 ■ **Turkey** Mikrobeta 312 467 7504

Windows all the way

One of Microsoft's goals when defining At Work was that 'the architecture must be compatible with the Windows PC architecture.' The easiest way to achieve this goal was to take Windows and modify it to fit the more usual embedded system requirements. What it imposes on processor and memory needs is not clearly specified in the documents I had access to, though one of them quoted an Intel 286 processor as a minimum. The detailed specifications of At Work are available only to OEMs through a licensing deal.

Like Windows applications, At Work applications are message driven and can have executables and DLLs. Even though At Work is pretty similar to Windows 3.1, it differs on several important points. For a start the whole kernel can be put in ROM and run from there. More subtle, At Work is a preemptive OS which can execute both the standard processes of Windows 3.1 that should not be interrupted and also processes which can be preempted. In an embedded device one would expect only preemptive processes, but this is not possible to execute safely in Windows 3.1 applications. These programs do not expect to be interrupted at any time and so usually do not lock any shared resources. For multi-tasking to work, preemptive processes are restricted in their use of shared resources. The two types of processes can communicate together via IPC mechanisms such as mutexes, signals and pipes.

The BOSS is at the bottom

As can be seen in Figures 1 and 2, at the lowest level the hardware is abstracted by the *basic operating system services* (*BOSS*) and by the *device drivers*. The device drivers provide a standard interface to the appliance's peripherals while the BOSS abstracts the appliance's hardware. In the BOSS module the *BOSS Managers* are located above the *Hardware Abstraction Layer* and are hardware independent. The BOSS controls the RTC (real time clock), DMA, timers and interrupts in order to provide low level non-volatile memory management, error handling a timing functions services to the next level in the At Work architecture.

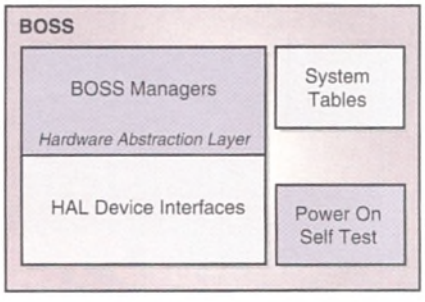


Figure 2 - Components of the Basic Operating System Software (Source Microsoft)

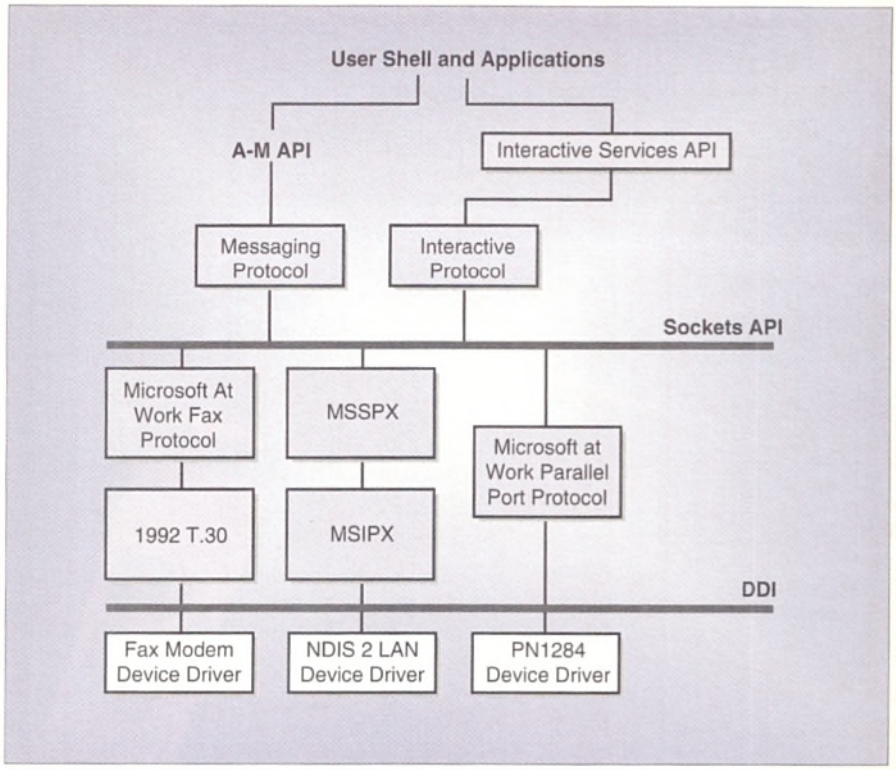


Figure 3 - Internals of the Communications module and associated interfaces (Source Microsoft)

On top of the BOSS sits the At Work Operating System which, like Windows 3.1, is composed of a *kernel*, *graphic services* and *file system*. As mentioned before, the At Work kernel extends Windows 3.1's to support preemptive processes. The graphic services provide a subset of the USER and GDI functions. The graphic services export about 100 API calls while USER and GDI export over 500 calls. True Type fonts, DDE, OLE, clipboard and multimedia extensions are amongst those discarded. Also, At Work cannot display multiple or overlapping Windows. The file system is a subset of the one which can be found in Win32. It can cope with multiple installable file systems but the default is FAT. A file system is not required for At Work to run. The file system interface supports some MS-DOS `int 21` calls even though the MS-DOS functionality is not present. The Win32 file system API is the one recommended for At Work application.

Applications and the user shell do not access the operating system directly. Instead they interface with the *communications* and *services* modules. The former deal with all the communication between At Work machines. The latter gives access to the functionality provided by the underlying hardware. To be more specific communication between At Work machines can be either message based or interactive depending on the applications' needs. For example, one messaging protocol currently available is the EFAX protocol which provides fax for At Work Fax (see Figure 3). The interactive protocol connects two At Work engines synchronously through remote procedure calls (RPC). It's main use is for control information. Also, both protocols are interfaced with network protocols be it fax T.30, a network stack or whatever.

At your service

Most of the functions of the At Work device get executed by the services module. It is made up of three elements (see Figure 4), covering messaging, imaging and remote access. Anytime one of these software modules needs to store information, it has access to an *object store* similar in principle to the MAPI message store. All the data contained in the object store is manipulated by the messaging services through a variant of MAPI 1.0 named *A-MAPI* for At Work MAPI. The imaging services modules are concerned with image processing such as rendering, printing, scanning. Remote access module, as its name indicates, processes locally commands issued remotely to the At Work device. It exports the A-MAPI and MAPIUTIL (a higher level API, see below) interfaces to the remote machine. In the other direction, an At Work application can connect to a remote Windows application with the sockets protocol.

The Hardlock software protection system.

Universal and individual.

MICROSOFT WINDOWS COMPATIBLE

yes It runs with NetWare

Available and Ready for OS/2

CE 93

Around the world, there are currently over 1.5 million Hardlocks in use. They are all virtually hackproof, daisy-chainable, individually interchangeable, compatible, and with an almost incredible variety of forms and features, they are fast and easy to implement.

Want to know more? Want to find out which Hardlock will suit your application? Want to order the FREE Hardlock evaluation pack? No problem! Just dial...

CIRCLE NO. 047

FAST Fast Electronic (UK) Ltd.

FAST-INFO-LINE: 0171-221-8024

FAST Electronic (UK) Ltd., Walmer Studio 4-6, 235-239 Walmer Road, London W11 4EY

Access to an At Work device is done through *applications* or a *user interface shell*. The connection is either direct from some sort of input mechanism located on the device or remotely from a computer connected to it directly or via a LAN The latter is just a specialised application. In fact, standard applications can access the user interface shell. One of its functions is to export MAPIUTIL, a high level interface to the object store. MAPIUTIL is to A-MAPI what Common Message Calls (CMC) is to MAPI 1. The user shell also provides some common controls such as a keyboard for touch screens. Applications have a choice for accessing At Work services. They can call the exported APIs but they can also generate an action by creating a message in the object store in a similar way to what happens when sending a message with MS-Mail.

An At Work application can be a program that resides completely in the device, a Windows application on the PC or hybrid software with components on both systems. On the PC side, a Windows application developer who wants to access some services provided by an At Work device does so with the A-MAPI and MAPIUTIL protocols. This is how it was done in Windows for Workgroups 3.11 which has the first implementation of the Microsoft At Work fax. Fax data is transferred from MS-Mail via MAPI.

Building a NEST

So what of Novell's foray into this lucrative market? NEST has a more layered, brick building, oriented hierarchy but its most original feature is hardware and operating system independence. The goal of NEST is to 'provide the benefits of NetWare, such as NetWare connectivity, NetWare services and NetWare interoperability to intelligent devices.' The NEST SDK 1.0 contains the source code of all modules. It's written in C and has been developed and tested using both 16- and 32-bit platforms with GNU C, Sierra C, Watcom C for DOS and Borland C.

Even though IPX/SPX is very widespread, the availability of a TCP/IP stack will be welcome

The whole NEST architecture comprises an embedded OS with three layers. These are connectivity, services and applications (see Figure 5). As can be seen, most of the modules are portable amongst different systems. The only modules that OEMs have to modify or rewrite are the *Portable Operating System Extension (POSE)* and the *Hardware Specific Module (HSM)*, which provides independent program access to network card and drivers. These are always device dependent. Only modules necessary for a specific implementation needs to be included. For the modules needed, NEST goes one step further by providing the code in library form. So, only the functions used need to be compiled in the final code. This is a design choice intended to reduce the overall memory capacity required.

At the bottom of the architecture is the *connectivity layer* which manages network independence (see Figure 6). Although the connectivity layer included in the NEST SDK 1.0 is based on an IPX/SPX protocol stack, it conforms to the ODI model. So other stacks can be added. Novell has already announced that it will provide a TCP/IP protocol stack in the next edition of the SDK. The foundation of this layer is the *Multiple Link Interface Drivers (MLID)* which is itself subdivided in three layers. The *Hardware Specific Module (HSM)* is specific to a particular network board. The *Topology Specific Module (TSM)* is specific to a particular topology such as Ethernet and the *Media Support Module (MSM)* contains functions common to all drivers. The brick architecture enforced in NEST means that a NEST device can have several concurrent MLIDs. The communication between MLIDs and protocol stacks is done by the *Link Support Layer (LSL)*. This module keeps track of all active MLIDs and stacks present in the device.

Layers of layers

On top of the connectivity layer is the *NetWare services layer* which provides C libraries to access services from the NetWare OS. This layer is again subdivided into a *NEST requester* and *client API libraries*. The requester assumes the role of an interface between the connectivity layer and the client API libraries. It provides basic services such as login and connection management. Client API libraries include NetWare functions such as a subset of the NetWare Core Protocol (NCP), client APIs (NWCALLS), directory services (NWNET), localisation services (NWLOCALE). Depending on the requirements of the device, client APIs can be added or dropped in a specific implementation. Novell announced that NEST will be enhanced to include most of the 400 functions of NCP.

The top of the architecture consists of the *application layer*. That's where the control of the device's operation and network access reside. The NEST SDK 1.0 includes an application for remote printers based on NetWare's NPRINTER and PSERVER but any type of application can be developed. The application layer is made up of modules to separate device dependent code from more generic code. For example in the remote printer example included in the SDK, the embedded PSERVER, the embedded

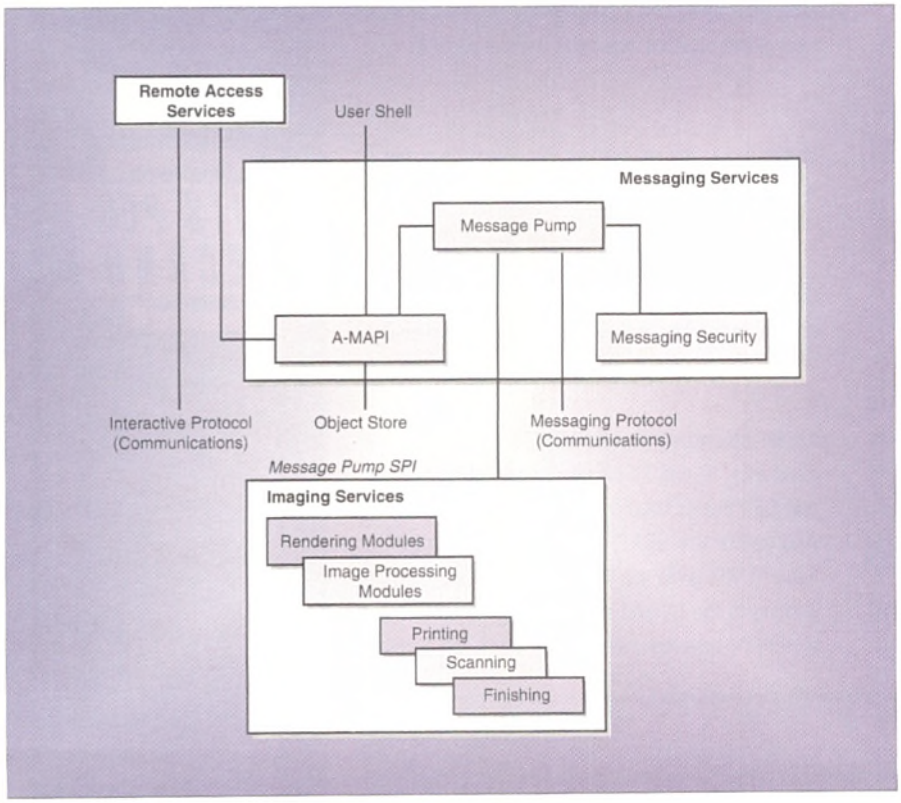
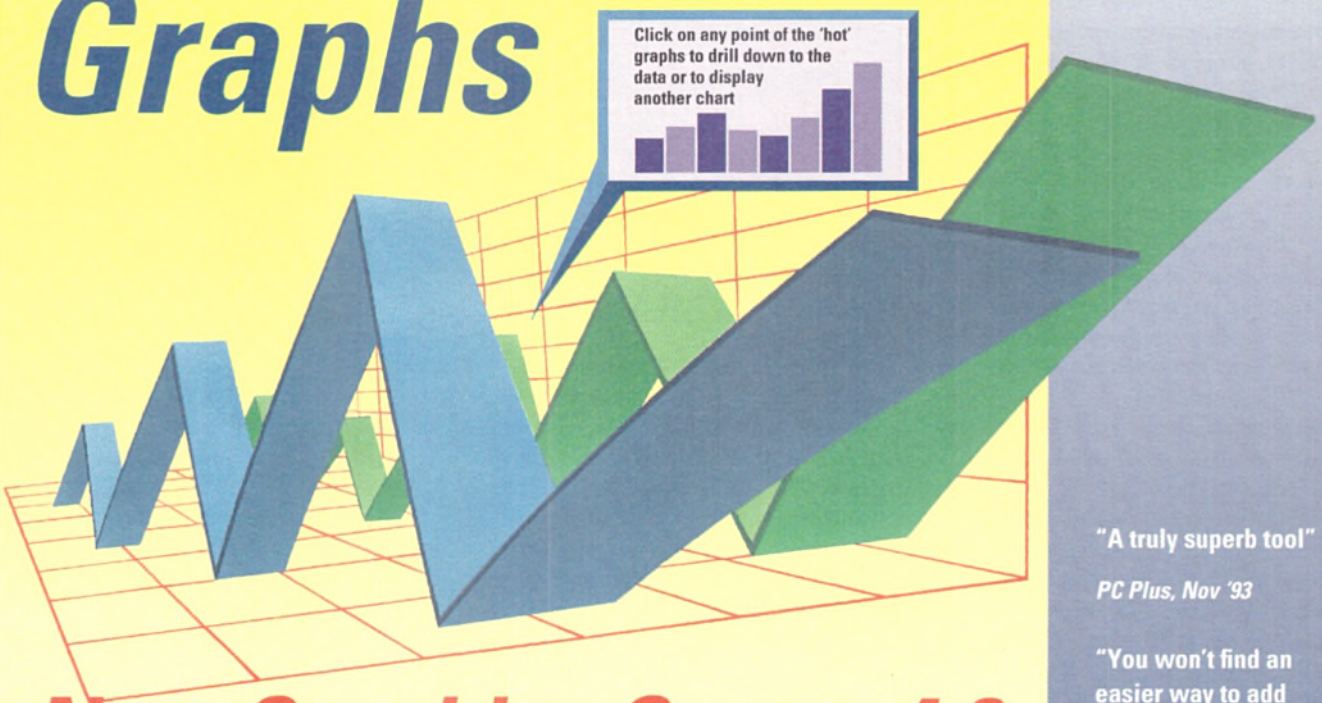


Figure 4 - Internals of the Services module (Source Microsoft)

Language

Visual C++
C/C++
SQLWindows
Visual Basic
Turbo Pascal
Powerbuilder
FoxPro
Windows4GL
Access

Programmable Graphs



New Graphics Server 4.0

Thousands of developers use the original GRAPH.VBX, licensed from us by Microsoft back in 1992.

Thousands more took advantage of the enhanced graphs and charts in the 1993 release of Graphics Server.

Now Graphics Server 4.0 is here.

NEW ***Interactive Toolbar***

By accessing the 'property pages', this new feature of the VBX can be used by you at design-time and by your users at run-time to change the properties of the graph or chart - and you can configure the toolbar.

You can also control every aspect of a graph or chart by accessing the 180+ low-level functions in the DLL.

NEW ***Rotatable True3D***

Now your users can project graphs in True3D with control over rotation, elevation and eye position.

NEW ***Real-time graphing***

One or more data sets can be displayed in a scrolling window and updated several times a second.

NEW ***More Graphs and Stats***

Contour plots with colour-grading, combination graphs, open-high-low-close, box-whisker and more.

AND STILL

All the interfaces you need (VBX, DLL, FLL, C++ graph class etc), 16 and 32 bit versions for Windows NT and '95 - there's even an OLE Custom Control (OCX). And unlike the GRAPH.VBX in VB Pro, Graphics Server's VBX gives you 'hot' graphs, full axis control, printing to the DC and null values for missing data, plus all the new features like True3D and the Toolbar.

STAY OUT IN FRONT !

Call us now, and find out how Graphics Server 4.0 can keep you and your users out in front.
(And remember, you can distribute the run-times royalty-free).

"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

BITS PER SECOND

14 Regent Hill
Brighton
England
BN1 3ED

Telephone
01273 727119

Fax
01273 731925

CIRCLE NO. 048

NPRINTER and the printer control module are generic for any type of remote printing device but the printer driver is specific to one printer hardware.

Which OS?

NEST doesn't specify the operating system needed to run all the modules but instead, to remain OS independent, it defines an interface between the embedded OS and all the other NEST modules. This POSIX-like API is called the *Portable Operating System Extension (POSE)*. It defines APIs for all the OS services that NEST software may call. POSE covers functions such as memory and process management, event, timer, interrupt and I/O control. The main requirements that NEST imposes on the underlying operating system is support for multiple thread priorities. Threads can also suspend themselves. All NEST modules run in a single memory address space. So if the underlying OS supports a process model then all NEST code has to run in a single process. A POSE interface is provided for Novell's own realtime operating system called FlexOS.

The SDK also includes a *NEST configuration tool* for clients to log on to a NEST device and access its configuration settings. This software is based on the NPLIB library which allows a remote client to log on a NEST device. In the other direction, a NEST device can log on to a NetWare server and access files on it by using the services of the client API libraries. NEST is compatible with NetWare 3.12 and NetWare 4.x

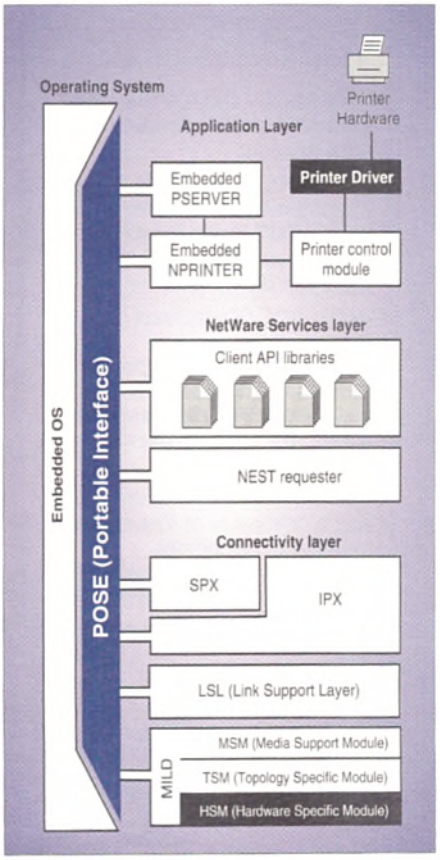


Figure 5 - NEST 1.0 architecture. OEM specific modules are in black (Source Novell)

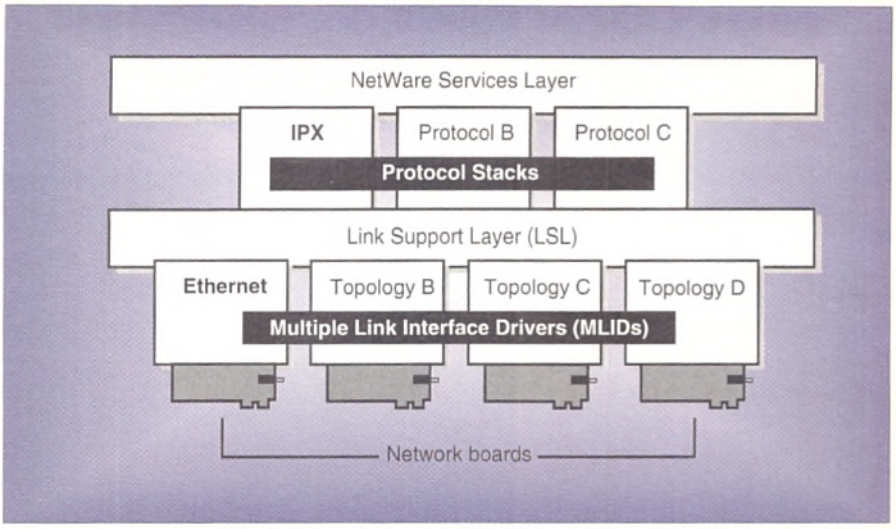


Figure 6 - NEST connectivity layer (Source Novell)

I have described the content of the NEST SDK 1.0 but Novell has already announced some of the elements that will make it in a future release. One of these will remove a limitation of the current SDK. In the actual implementation, when a client wants a service provided by a NEST device, the client has to know *which* device provides *which* service and request access to this specific device. What will be delivered in the next release is a CORBA-compliant service broker. When it will be available, clients will be able to request a service and the service broker will check if a device has advertised such a service. If so, it will connect the client to the correct device without the client having to know where the device is located.

Who controls the standard?

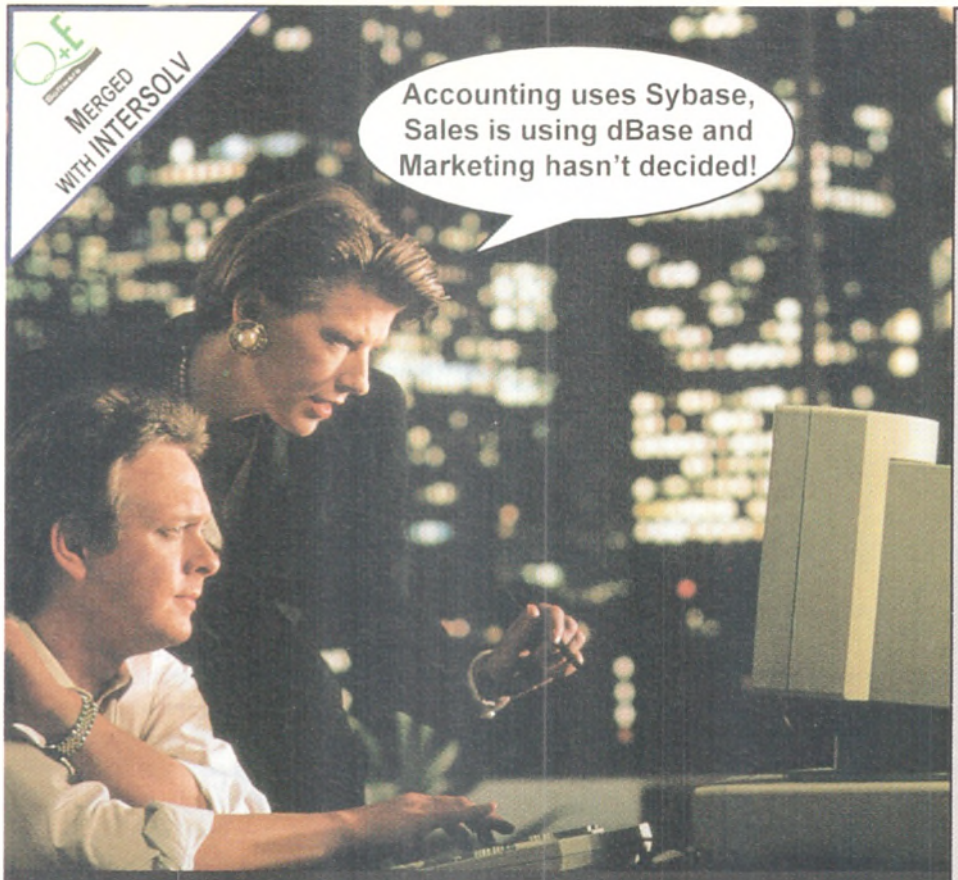
In the case of At Work, Microsoft imposes its will on both the client developers and the OEM. Even though Microsoft has the longest list of OEMs committed to its system and At Work was the first 'standard' available, there are not many compliant products available today. It looks like its a failure. The At Work group has now been incorporated in the Windows 95 team, even though Microsoft still says it is committed to this architecture. At Work seems well suited to high end office appliances which incorporate display and input facillites such as a top of the range fax machine or photocopier. But somehow I feel isn't it overkill for a small appliance such as a telephone? Since it's based on Windows, it must need a fast processor and lots of RAM hence increasing substantially the cost of the appliance.

Novell's NEST is more flexible but it also has a few limitations. The first implementation has only one stack. Even though IPX/SPX is very widespread, the availability of a TCP/IP stack will be welcome. Also Novell provides all the source of NEST but OEMs have to implement a solution based on it. It is a restriction in its own right.

If the standard currently developed by IBM *et al* is anything like what has been leaked so far, then it might be really attractive for OEMs. First they would not be tied to one company strategy, be it Microsoft or Novell. Amongst the 15 companies at least some will be users of such technology and so will hopefully influence a very workable design. What seems to be a winning decision is the fact that this standard, allegedly, does not define the software that has to be incorporated in the appliance but only the API to which it must conform. This simple fact means that OEMs will be free to write whatever software as long as it conforms to the API. In other words they will be more free in their design decisions.

As expected, there is some confusion in the industry. For instance, Ricoh is amongst 15 companies defining the new standard but it is also a licensee of both Microsoft's and Novell's embedded technology. Lexmark is also a licensee of the two existing products and will probably adopt any standard in which IBM is involved. So for an OEM the choice is not easy.

I have the impression that in the current situation, OEMs have licensed all the available technology to 'have a look'. In other words, the list of licensees probably do not reflect a real commitment from all the companies on the list. For this very specific reason I would not yet advise application developers to look in too much detail at how they can write applications which take advantage of communicating with all the standard electronic equipment lying around the office or at home. Stay tuned... ■



"Hardwiring our client/server applications to a single DBMS is crazy!"

INTERSOLV's ODBC Connects To:

- 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
- Paradox
- PROGRESS
- SQL/400
- SQL/DS
- Sybase
- Sybase SQL Server
- Teradata
- Text files
- XDB

Operating Systems Supported:

- Windows
- OS/2
- Windows NT
- Macintosh
- Solaris

One Development View

INTERSOLV's cross-platform data accessware is for developers building database-independent applications using ODBC. Our unique, thinly layered approach leverages productivity and insulates your application without blocking access to key DBMS functionality. With INTERSOLV, you code and maintain a single set of highly efficient source code regardless of the number of different data sources.

Many Deployment Options

Using INTERSOLV's drivers, you can link any ODBC-enabled software to your choice of over 30 different data stores. Even ODBC applications not built using INTERSOLV's developer kit are eligible. Most important, our ODBC drivers are fast and memory efficient—unlike other ODBC drivers that hamper performance. So, if you need to link Excel to dBase, Lotus 123 to SYBASE or Access to INFORMIX, check out INTERSOLV's ODBC drivers

INTERSOLV DataDirect - the standard in client/server data access:

INTERSOLV DataDirect Developers Toolkit allows you to build DBMS-independent, ODBC-compliant applications. **£500 per developer.**

INTERSOLV DataDirect ODBC Pack enables users to seamlessly access corporate data from any application that supports ODBC standards. **£350 for all drivers, £85 per driver.**

INTERSOLV DataDirect Multilink/VB increases the power and functionality of Visual Basic to boost development productivity and optimise system resources. **£300 per developer.**

INTERSOLV with ODBC is the Answer!

The Standard for Client/Server Data Access

Building or deploying an ODBC-enabled application?

Call today: 0727 812812, ext. 262

INTERSOLV

CIRCLE NO. 049

How would you like your steak Sir?

The choice ranges from 'blue' to 'well done'. Everything in between is cooked to some degree. Both the chef and the carnivorous diner use fuzzy logic to communicate. **Kevin Yeandel** prepares some food for thought.

There are many possibilities for using fuzzy logic, particularly in engineering. But please be clear from the start: 'fuzzy' *does not* imply 'imprecise'. Some companies genuinely fear a reduction in customer confidence if their products are associated with the word 'fuzzy'. If this term is uncomfortable then call it 'multivalent logic'. Later we use C++ to demonstrate how an electric vehicle may be run at optimum efficiency when faced with constantly changing conditions.

A fuzzy recap

For the sake of those who missed *The fuzzyman cometh* in EXE (September 94), fuzzy logic is a way or finding the best output from a given set of uncertain inputs. To put it another way, number crunching of real life information may require the use of various, often complex, equations. For example, say we want to maintain the efficiency, comfort and speed of an electric vehicle operating under certain environmental conditions. How would we code the following?

```
if ROAD is FLAT then
  POWER = LOW.
```

Perhaps not a big problem but what about other factors

```
If  LOAD is MED_HEAVY and
    ROAD is STEEP and
    BATTERY is QUITE_LOW then
  POWER = HIGH.
```

Easy to read, easy to understand, slightly harder to implement but very hard to define a fully optimised and efficient system which provides the *perfect* output with a *real world* data input. However, thanks to Lotfi Zadeh's fuzzy logic, engineers need only get aquatinted with elementary lingo such as the above. The complexities remain hidden behind the fuzzy engine, either as a program or dedicated chip.

To understand the program presented in this article we need a little more background on what it is we are trying to achieve. The statement `LOAD is HEAVY` may be true, partially true, or false. `HEAVY` is a set defined using human judgement. The definition being a range from a low to high, say 1.5 to 2.5. In this case, for `LOAD` to belong to `HEAVY`:

```
(1.5 < LOAD < 2.5).
```

Nevertheless `HEAVY` is a set with boundaries which, in fuzzy systems, should overlap the edges of adjacent sets, say, `AVERAGE` and `VERY_HEAVY`. Figure 1 demonstrates this in the form of a fuzzy graph. The input value

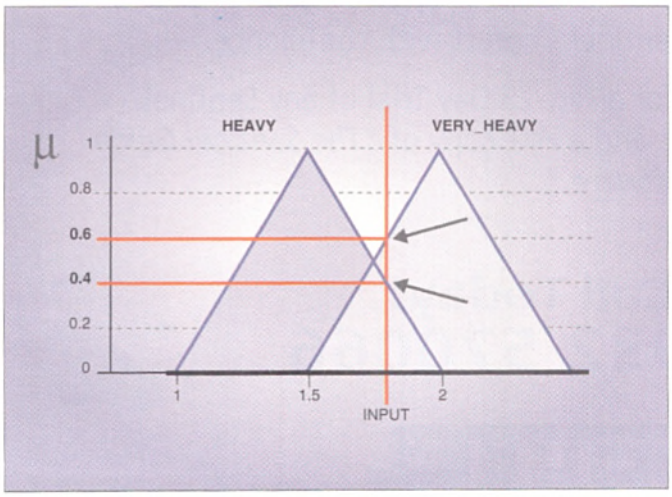


Figure 1 - Input membership determined by intersection of triangles

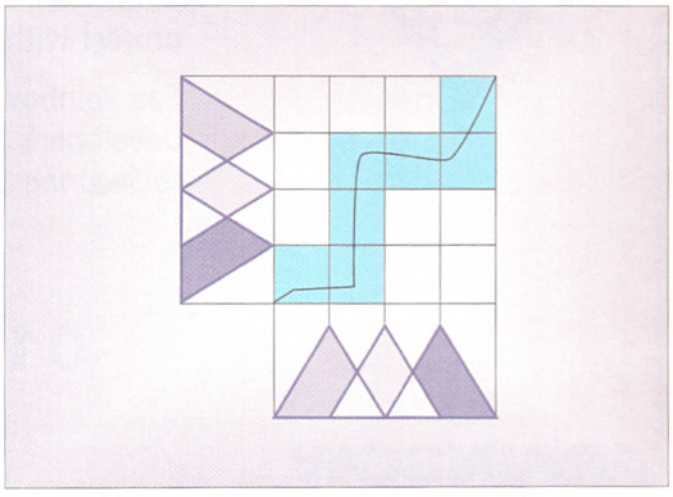


Figure 2 - Complex relationships lead to complex formulae

There are Billions of reasons to protect your software.

You only need one reason to protect with Sentinel:

It's the worldwide standard in software protection.

Piracy is the greatest threat to the world's software industry. Developers lose billions in sales to software piracy each year. Protect your software and get all the revenue you deserve.

More developers rely on Sentinel®, from Rainbow, than any other software protection in the world. And for good reason. Sentinel performs where it matters most: leading the industry in technology, quality, reliability and support.

So when it's time to protect your DOS, OS/2, Windows, NT, Macintosh, UNIX, XENIX, LAN or CD-ROM application – protect with Sentinel. Protect with confidence.

Call Rainbow for a FREE 28 Day Trial of any Sentinel Developer's Kit and a FREE copy of *"The Sentinel Guide to Securing Software."*

Call Today!
01932 570066

RAINBOW TECHNOLOGIES

SENTINEL
Securing the future of software

4 The Forum, Hanworth Lane • Chertsey, Surrey KT16 9JX • Fax: 01932 570743 • *International offices and distributors located worldwide.*
Rainbow USA 1-714-454-2100 • Rainbow France (33) 1 47 38 21 21 • Rainbow Germany (49) 89 32 17 98 0

CIRCLE NO. 050

presented to the fuzzy graph belongs to both HEAVY and VERY_HEAVY. The actual membership is determined by the intersection of the input against the edges of the triangles as shown by the two arrows. In this case, the input belongs mostly to VERY_HEAVY (0.6) and slightly less to HEAVY (0.4). This is called the *mu* value or membership and forms a value between 0 (false) and 1 (true). Therefore it is 'more' true to say that the vehicle is VERY_HEAVY than to declare it HEAVY. A simple expression such as

```
if LOAD is LIGHT then
  POWER = LOW
```

would produce an output directly corresponding to the input and life would be simple. More often, we need to accommodate situations with many uncertainties as previously described. This causes a problem: How do we derive an output based on multiple, uncertain input values?

Rule base

A graph, as in Figure 2, would require formulae of severe complexity. It could be difficult deriving a formula for all possible situations. So, instead, we use a rulebase comprising sequences of `if...or...and..then` to describe the problem easily. The example program listed in this article will demonstrate the technique..

The rulebase, as its name implies, can exist internally (as in our example code) or be a program in a file, like a C or Basic source file. It uses simple terminology describing the problem and shifting the output to the appropriate set.

Defuzzification

We are not done yet! We still don't have the perfect answer. The rulebase provides values belonging to the output set, not the actual output. The true result is derived from a process called *defuzzification*. This involves finding how much of the output each set possesses. We defuzzify the answer by using the *centre of gravity* method (Figure 3). But more about this later.

A fuzzy program

The roots of the example program in Figure 5 are based on common Prolog-style implementations which use defined tables of data effectively 'drawing' the membership sets. In our example, `ROAD[]` is an array of floats `{0, 5, 10, 15, 20}` defining the angles of climb which may be faced by our truck. Following this, `FLAT`, `INCLINE` and `STEEP` determine the shapes of the membership functions. The value 1 or 0 corresponds directly with the `ROAD[]` values. For example, at 0 and 5, `FLAT` has a value of . At 10, `FLAT` is at 0. Mentally plot and join the points you see the form of a left-inclusive. A study of the example data reveals the presence of the right inclusive and plain, triangular inclusive for the two inputs and single output.

Probably the most important function within the program is `membership`. `Membership` applies one of the input values to the appropriate range to determine which set the input belongs and returns a value corresponding to the degree of membership. This degree is a derivation of **dy/dx * d** as Figure 4 depicts. The membership function is capable of discriminating between both left/right inclusives and pyramidic inclusives hence the two independent formulae which can be seen within this function.

The next important function to consider is `TryIt`. Taking two inputs, `TryIt` passes this data to the rules which simply create priority data by comparing the current calculation with that of the previous and returns the maximum value through a call to the generic `max`. The multiplication of the two `mu` values, to put it simply, find the location within a graph such as in Figure 2.

A completion of the rules (as many as you like in any order) results in three float variables which contain data created from both inputs. This information is multiplied by 'vertical strips' of the particular input set. For example:

```
if fHEAVY = 0.6 then:
0.6 * (1,1,0,0,0) =
          (0.6, 0.6, 0, 0, 0)
```

The `SetSetData` function does this for us and fills out a temporary array `(fCOPY_2)` to pass to `GetTheMax` which subsequently compares each member of the temporary array with that of a previous. `GetTheMax` is initially called with an array of 0s `(f0)` and returns `fCOPY`, which is an array containing the maximum values created from the two input arrays.

At this point we have established member values for the three output sets, `LOW`, `MEDIUM` and `HIGH`. Three `mu` values? So which one determines the actual output? The trick of getting a single output is achieved in the final `Defuzz` function.

`Defuzz` takes one parameter, an array of floats, in this case `CURRENT {25, 50, 75, 100, 125}`. For each of these values the mid point between `u` and `u+1` is found. So for the first one we have

```
(25+50)  /  2 = 37.5
```

We then multiply by the height of the section below the `mu` value for that set. We are only interested in areas (or masses) beneath the `mu` value. As `Defuzz` applies its trigonometric calculations it generates a pair of running totals the eventual ratio of which provides the output.

Now the sequence starts again with the next inputs pair. The program contains the fewest lines possible. Similar systems are successfully used in the stock markets in Japan. Although they do have a few more rules - about 5000 more! ■

Code for this article may be found overleaf.

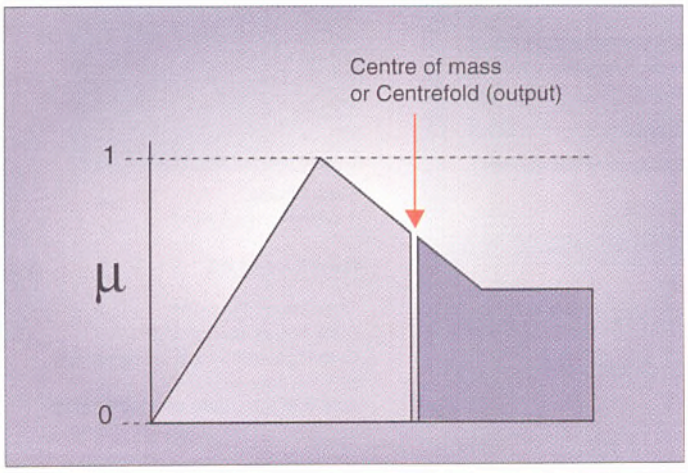


Figure 3 - Output is derived from the average centre of mass

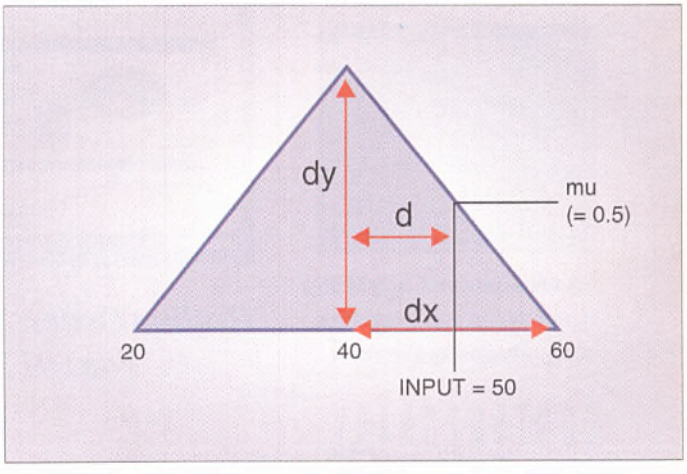


Figure 4 - dy/dx can be used to find the value for mu

Accelerate reporting with the #1 award-winning reporting tool.

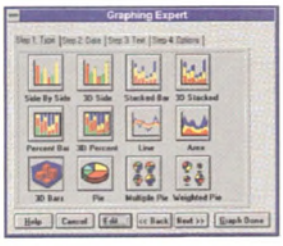
New Crystal Reports Version 4.0

There's no better way...to create reports!
Design reports without coding! Do two-pass calculations, sort and subtotal in just one step, sort on groups, export reports to many file formats, distribute reports via integrated e-mail and much more!

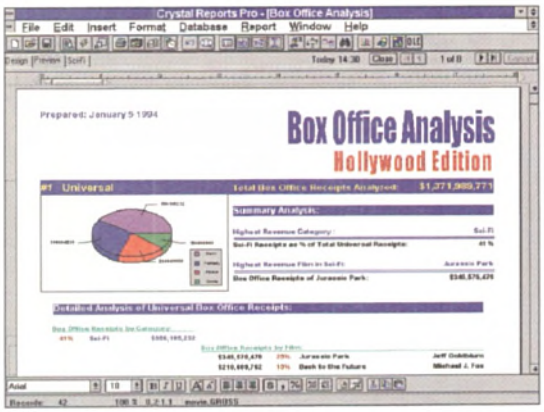
There's no better way...to integrate reports into applications
Crystal Reports' unique architecture consisting of a Report Engine DLL and open APIs make it the ultimate tool for fast, seamless report integration into database applications. That's why Microsoft, Borland, Sybase and over 50 other vendors bundle Crystal in their shrink-wrap programs.

There's no better time to buy the #1 reporting tool...now with hot NEW features in Crystal Reports 4.0

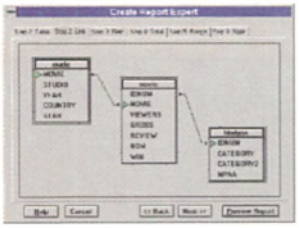
- Greater Report Engine Control
  >80 calls (20 new)
  80 VBX properties (40 new)
  All new OCX
- Blob bitmap support (Paradox, xBASE)
- Report Distribution Expert collects DLLs, EXEs, and configuration files for report distribution.
- Integrated Graphing: 12 styles



- New Select Records Query Interface ▪ Tab Interface ▪ Best Fit
- Automatic Styles ▪ Grid Control Options ▪ Ruler ▪ Undo
- Title and Summary Pages ▪ Bar code fonts



- 2-10 times faster report processing with Smart Engine Technology
- Stored Procedure support
- Ability to sort data on the SQL server
- Drilldown analysis capability
- Crystal Reports Experts:



- Free Runtime
- *And much, much more!*

Crystal Reports 4.0 *accelerates* report design and integration of reports into applications.

Hotline Order: Tel: 01727 811999
Fax: 01727 848991

Prices R.R.P. ex VAT

Crystal Reports Pro 4 £299
Crystal Reports Standard £149
Crystal Reports Pro 4 Special upgrade price for Visual Basic users £149
CALL FOR ALL OTHER UPGRADE PRICES

CONTEMPORARY *Software*

Contemporary Software
Abbey View, Everard Close, St Albans, Hertfordshire AL1 2PS

CONTEMPORARY SOFTWARE A DIVISION OF INTERSOLV PLC

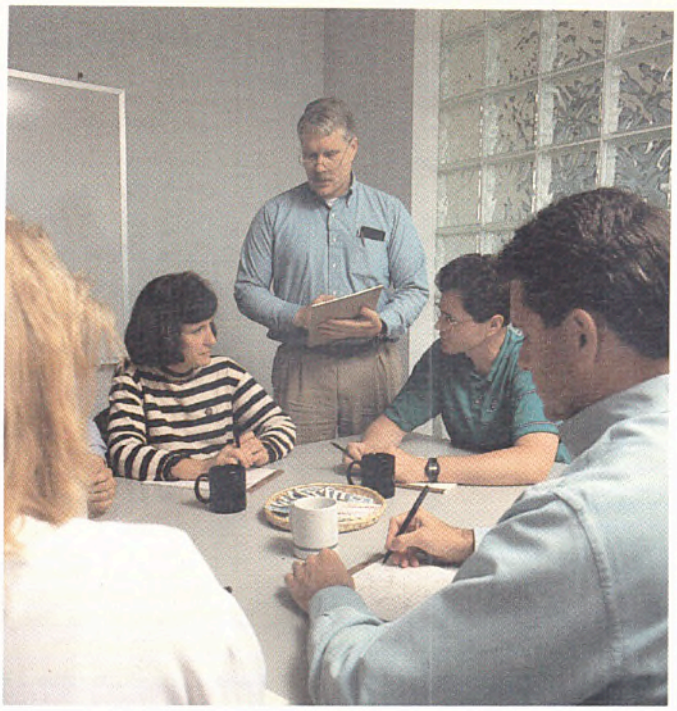
The #1 award-winning reporting tool is now even better!

Sept. 27, 1993 — May 25, 1994 — August 19, 1994

CIRCLE NO. 051

```
#include <stdlib.h>
#include<fstream.h>
//   FUZZY C++ Example for an electric truck
const CData = 5; // number of data items in the
arrays
class Fuzzyfier
{
public:
  Fuzzyfier();
  ~Fuzzyfier(){delete fCOPY_2, fCOPY, f0;}
  float TryIt(float, float);  // process the
options
protected:
  float Membership(float , float[], float[]);
  void  GetTheMax(float[], float[], float[]);
  void  SetSetData(float , float[]);
  float Defuzz(float[]); // returns the output
value
private:
  float *fCOPY, *fCOPY_2; // temporaries float
arrays
  float *f0; // an array of 0's
};

Fuzzyfier::Fuzzyfier()
{
  fCOPY = new float[CData + 1];
  fCOPY_2 = new float[CData + 1];
  f0 = new float[CData + 1];
  _fmemset(f0, 0, sizeof(float) * CData);
};

float // sample data
    ROAD[] = {0,5,10,15,20}, // range data (angle
of climb)
    FLAT[] = {1,1,0,0,0},    // Membership data.
This forms the
  INCLINE[] = {0,0,1,0,0}, // left and right
inclusives
    STEEP[] = {0,0,0,1,1},   // and pyramid centre
value

// range of weights of loads in an electric truck
    WEIGHT[] = {0,1,1.5,2,2.5},
    LIGHT[] = {1,1,0,0,0},    // left inclusive
  AVERAGE[] = {0,0,1,0,0},  // inclusive
    HEAVY[] = {0,0,0,1,1},    // right inclusive

  CURRENT[] = {25,50,75,100,125}, // range in amps
    LOW[] = {1,1,0,0,0},    // left inclusive
    MED[] = {0,0,1,0,0},    // inclusive
   HIGH[] = {0,0,0,1,1};    // right inclusive

float Fuzzyfier::Membership(float fI, float
InputSet[],
                        float fA[])
{
float ff;
  if (fI < InputSet[0])   // can't be in the low-
est set
    return InputSet[0];  // data out of range,
return min
  else
  if (fI > InputSet[CData - 1])  // is above the
highest set
    return InputSet[CData - 1]; // data out of
range, return max

  for (int u = 0; u < CData - 1; u++) {
    if ((fI >= InputSet[u]) && (fI <= InputSet[u +
1])){
    if (InputSet[u] == InputSet[u + 1])
      ff = max(fA[u], fA[u + 1]);
    else
      ff = (fI - InputSet[u]) * ((fA[u + 1] -
fA[u]) /
          (InputSet[u + 1] - InputSet[u])) + fA[u];
    }
  }
  return(ff);
}

void Fuzzyfier::GetTheMax(float f1[], float f2[],
float f3[])
{
  for (int u = 0; u < CData; u++)
    f3[u] = max(f1[u], f2[u]);
}

void Fuzzyfier::SetSetData(float fOset, float
fVal[])
{
  for (int u = 0; u < CData; u++)
    fCOPY_2[u] = fOset * fVal[u];
}

float Fuzzyfier::Defuzz(float fOut[])
{
float A_summation = 0, B_summation = 0;
  for (int u = 0; u < (CData - 1); u++){
    A_summation += ((fOut[u] + fOut[u + 1]) / 2) *
            ((fCOPY[u] + fCOPY[u + 1]) / 2);
    B_summation += (fCOPY[u] + fCOPY[u + 1]) / 2;
  }
  if (B_summation == 0)
    return 0;   //       /0 error
  else
    return A_summation / B_summation;
}

float Fuzzyfier::TryIt(float fInputA , float fIn-
putB)
{
float   fFLAT, fINCLINE, fSTEEP,
      fLIGHT, fAVERAGE, fHEAVY,
      fLOW,   fMED, fHIGH;

  cout <<"ROAD = "<<fInputA<<" WEIGHT = "<<fIn-
putB<<" Output ";
  fHEAVY   = Membership(fInputB, WEIGHT, HEAVY);
  fAVERAGE = Membership(fInputB, WEIGHT, AVERAGE);
  fLIGHT   = Membership(fInputB, WEIGHT, LIGHT);
  fFLAT    = Membership(fInputA, ROAD, FLAT);
  fINCLINE = Membership(fInputA, ROAD, INCLINE);
  fSTEEP   = Membership(fInputA, ROAD, STEEP);

  fLOW  = fMED = fHIGH = 0;

  // IF ROAD = FLAT AND WEIGHT = LIGHT THEN POWER
= LOW
  fLOW  = max(fFLAT * fLIGHT, fLOW);

  // IF ROAD = INCLINE AND WEIGHT = LIGHT THEN
POWER = LOW
  fLOW  = max(fINCLINE * fLIGHT,fLOW);

  // IF ROAD = STEEP AND WEIGHT = LIGHT THEN POWER
= MED
  fMED  = max(fSTEEP * fLIGHT,fMED);

  // IF ROAD = INCLINE AND WEIGHT = AVERAGE THEN
POWER = HIGH
  fHIGH = max(fINCLINE * fAVERAGE,fHIGH);

  // IF ROAD = FLAT AND WEIGHT = AVERAGE THEN
POWER = HIGH
  fMED  = max(fFLAT * fAVERAGE,fMED);

  // IF ROAD = STEEP AND WEIGHT = AVERAGE THEN
POWER = HIGH
  fHIGH = max(fSTEEP * fAVERAGE,fHIGH);

  // IF WEIGHT = HEAVY THEN POWER = HIGH
  fHIGH = max(fHEAVY,fHIGH);

  SetSetData(fLOW,LOW);
  // first time round compare with zero
  GetTheMax(fCOPY_2,f0,fCOPY);
  SetSetData(fMED,MED);
  GetTheMax(fCOPY_2,fCOPY,fCOPY);
  SetSetData(fHIGH, HIGH);
  GetTheMax(fCOPY_2,fCOPY,fCOPY);
  return Defuzz(CURRENT);
}

void main(void)
{
Fuzzyfier Fzzy;
float
  // sample data
  fRoadData[] =   {7  ,7.9 ,4  ,4  ,4  ,8.2,8  ,8
,8.1,15,20},
  fWeightData[] =
{0.4,1.62,1.2,1.2,2.2,2.4,0.2,1.7,1.9,1.3,2};

  // test it with some data
  int nCount = sizeof(fWeightData) /
sizeof(float);
  for(int i = 0; i < nCount; i++)
    cout <<
Fzzy.TryIt(fRoadData[i],fWeightData[i]) << endl;
}
```

Figure 5 - C++ for simple rules to control an electric vehicle



NEW RELEASE! Call for information

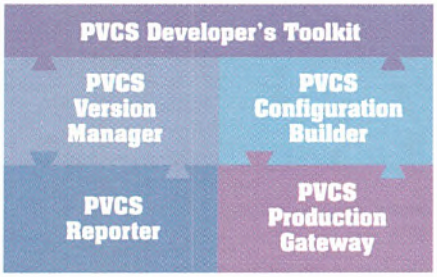
"We chose PVCS because the vendors of our new development tools recommended it. They told us Software Configuration Management was a necessity for productive development on the LAN. I'm glad we listened to them. PVCS has proven to be invaluable. We now have superb control over all development efforts."

FREE DEMO DISK

The Major Development Tool Vendors Agree: PVCS Is Essential for Productive Development on the LAN

Selected Vendors:

Borland International IDE
Digitalk Team/V
Gupta Technologies, Inc.
HP SoftBench
IBM SDE/6000
IBM OS/2
Interactive Unix
Micro Focus COBOL
Microsoft Visual C++
Microsoft PWB
Powersoft PowerBuilder
SCO
SunOS
SunSoft
Solaris
Symantec C++ IDDE
And many more...



Leverage Your Tool Investment with PVCS

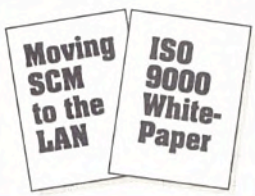
There's a reason why all major Client/Server and LAN development vendors have formed alliances with INTERSOLV. It's PVCS—the standard for Software Configuration Management on the LAN. It's a vital ingredient for productive LAN development.

Every Major Development Tool Vendor Partners with Intersolv

The list on the left is only partial. PVCS is available in all the environments developers work in—today and tomorrow. That means less training–you don't have to learn another SCM tool if there's a change. That means more efficient development and easily enforced standards. It means consistent audit trails for all development, and the ability to correct errors and anomalies at any time.

Serious Application Development Demands PVCS

The trend is clear. Important applications in your company are moving to the LAN. They need the control and reliability that PVCS provides. Your development tool vendors know that–that's why they recommend PVCS.



Call Today for More Information:
Tel: 071 625 5255
Fax: 071 624 9404

NEW VISUAL BASIC INTERFACE

The Standard for Software Configuration Management on the LAN

Readmar Systems Ltd, 239 Kilburn Park Road, London NW6 5LG

CIRCLE NO. 052

C++ Copy Constructors

Copy constructors might look complex but **Graham Kendall** thinks he's cracked them. Here is his guide to how and why they work as they do.

Since starting to use C++ I have read on a number of occasions that copy constructors are important, especially when pointers are used within a class. When developing classes I tried to avoid them by either telling myself that the class did not contain a pointer or there would only ever be one instance of an object. But as I get more into the spirit of C++ I decided this was a very naive approach and was not really fair on people who might use my classes in the future.

I decided to take some action so that I could understand the problem. Hopefully, I would discover it to be not as frightening as some publications would have us believe. The best starting point seemed to be to describe the problem in English. This is what I came up with.

Copy constructors for everyone

When you create a new instance of an object by copying another instance, it appears as if everything works okay with the exception of pointers. The problem is that the copied pointer will point at the memory location for the instance of the original object. This is a problem for a couple of reasons. The first is that the data will not be correct as you will be accessing the data of the original instance. The second and more dangerous reason, is that the data you are pointing at could be overwritten when the original object goes out of scope or is `delete'd`.

```
//: EXE001.CPP - Demonstrate problems
//  of copying objects
#include <conio.h>
#include <iostream.h>

class demo {
  int i;
  int *ip;
public:
  demo(int x=10,char *msg="");
  // Constructor
  void seti(int x) {i=x;};
  void dump(char *msg=
       "Dumping demo object : ");
};

demo::demo(int x,char *msg) {
  cout<<"In demo constructor "<<msg<<endl;
  i=x;
  ip=&i;
}

void demo::dump(char *msg) {
  cout<<msg;
  cout<<"\n i : "<<i<<" ("<< &i<<")"<<endl;
  cout<<" ip : " <<ip<<" ("
      << *ip<<")"<<endl;
}

void main() {
 cout<<"\n\nStart...."<<endl;
 demo one(10,"for object one");
 demo two(12,"for object two");
 demo three=two;
 one.dump("Object one : ");
 two.dump("Object two : ");
 three.dump("Object three : ");

 two.seti(14);

 one.dump("Object one : ");
 two.dump("Object two : ");
 three.dump("Object three : ");

 getch(); // To freeze display
}
```

Figure 1 - Copying objects

```
Start....
In demo constructor for object one
In demo constructor for object two
Object one :
 i : 10 (0x3c64fff2)
 ip : 0x3c64fff2 (10)
Object two :
 i : 12 (0x3c64ffee)
 ip : 0x3c64ffee (12)
Object three :
 i : 12 (0x3c64ffea)
 ip : 0x3c64ffee (12)
Object one :
 i : 10 (0x3c64fff2)
 ip : 0x3c64fff2 (10)
Object two :
 i : 14 (0x3c64ffee)
 ip : 0x3c64ffee (14)
Object three :
 i : 12 (0x3c64ffea)
 ip : 0x3c64ffee (14)
```

Figure 2 - Sample output from program in Figure 1

The program in Figure 1 will hopefully confirm this theory. It's quite simple. It defines a class with two data members. One data member is an `int` and the other is a pointer to an `int`. The idea is that the pointer points to the first data member. You can see this being set up in the constructor. The other function of interest is `dump`. This displays the two data members. It shows the value of `i` together with its address, the value of `ip` and the value that is stored at its location. By showing these two pieces of information we can see if they all reference the same data, which they should.

The program creates two instances of the object using the normal method. It then creates a third instance but copies another instance to do this. The three objects are now dumped so we can see if we have a problem. If you look at the sample output, in Figure 2, you can see that object `one` and object `two` look okay. By this I mean that the address of the variables `i` and `ip` match, ie the location

```
//: EXE002.CPP - Show introduction
//  of copy constructor
#include <conio.h>
#include <iostream.h>

class demo {
  int i;
  int *ip;
public:
  demo(int x=10,char *msg="");
  // Constructor
  demo(demo &dmo);
  // Copy Constructor
  void seti(int x) {i=x;};
  void dump(char *msg=
       "Dumping demo object : ");
};

demo::demo(int x,char *msg) {
  cout<<"In demo constructor "
      <<msg<<endl;
  i=x;
  ip=&i;
}

demo::demo(demo &dmo) {
  cout<<"In demo copy constructor"<<endl;
  i=dmo.i;
  ip=&i;
}

void demo::dump(char *msg) {
  cout<<msg;
  cout<<"\n i : "<<i<<" ("<< &i<<")"<<endl;
  cout<<" ip : " <<ip<<" ("
      << *ip<<")"<<endl;
}

void main() {
 cout<<"\n\nStart...."<<endl;
 demo two(12,"for object two");
 demo three=two;
 two.dump("Object two : ");
 three.dump("Object three : ");

 two.seti(14);

 two.dump("Object two : ");
 three.dump("Object three : ");

 getch(); // To freeze display
}
```

Figure 3 - A copy constructor

of `i` is what is held in `ip`. Object `three` gives cause for concern. The variable `three.i` has its own address but `three.ip` points to `two.i`. Unless we take specific action, this seems to confirm our theory that a pointer will point to the instance from which it was created.

The remainder of the program changes the value of `two.i` and dumps the three objects again. You can now see that `three.i` has a value of `12` whereas `three.ip` points to a value of `14`, ie `two.i`, further proving that things are not working as they should.

My first copy constructor

The program in Figure 3 introduces a copy constructor. It's also based on the previous program. The main function is slightly different. If you do not supply a copy constructor a default one is created for you. This default function copies the object in the last program. If you write your own copy constructor the default copy constructor does not get generated and you have to take over responsibility for its actions.

```
Start....
In demo constructor for object two
In demo copy constructor
Object two :
 i : 12 (0x3c66fff2)
 ip : 0x3c66fff2 (12)
Object three :
 i : 12 (0x3c66ffee)
 ip : 0x3c66ffee (12)
Object two :
 i : 14 (0x3c66fff2)
 ip : 0x3c66fff2 (14)
Object three :
 i : 12 (0x3c66ffee)
 ip : 0x3c66ffee (12)
```

Figure 4 - Sample output from program in Figure 3

In our copy constructor we need to do two things. We need to copy the value of `i` from the object we are copying from. We also need to make `ip` point to the new version of `i`. A copy constructor is passed a reference to the instance it is copying. It's this reference parameter which allows the compiler to recognise we are introducing a copy constructor. You can use this reference to access the member data of the original instance, hence the line `i=dmo.i`. We also set `ip` to point at the new `i`. After modification, the sample output of the program is listed in Figure 4. You can see that the copy constructor has been called. After `two.i` has been set to `14`, `three.i` and `three.ip` still have the correct values.

So that's it. Copy constructors cracked. Well not quite. Take a look at the program in Figure 5. This time we have overloaded the = operator. We need to do this because if we write something like `one=two`, as opposed to `demo one=two`, the copy constructor is not called but the `operator=` function is. Therefore we need to provide similar code within the overloaded = function as we did in the copy constructor. You will notice that the `operator=` function has a return type of the class it is defined in. This is why we have to `return *this` in the function, so that the object is passed back out to us.

If you look at the output from the program you might wonder why the copy constructor is called after the `operator=` function. If you remove the `operator=` function from the program the copy constructor will no longer be called but you will get pointer problems. This illustrates that the copy constructor is being called as part of the `operator=` function. In fact it is called as part of the `return *this` statement. This is due to a temporary copy of the object being created.

You could also try changing the `return *this` to something like `return 0`. When you run the program you will find that the

```
//: EXE003.CPP - Show introduction
//  of operator =
#include <conio.h>
#include <iostream.h>

class demo {
  int i;
  int *ip;
public:
  demo(int x=10,char *msg="");
  // Constructor
  demo(demo &dmo);
  // Copy Constructor
  demo operator=(demo &dmo);
  // Overloading = operator
  void seti(int x) {i=x;};
  void dump(char *msg=
       "Dumping demo object : ");
};

demo::demo(int x,char *msg) {
  cout<<"In demo constructor "<<msg<<endl;
  i=x;
  ip=&i;
}

demo::demo(demo &dmo) {
  cout<<"In demo copy constructor"<<endl;
  i=dmo.i;
  ip=&i;
}

demo demo::operator=(demo &dmo) {
  cout<<"In overloaded = function"<<endl;
  i=dmo.i;
  ip=&i;
  return *this;
}

void demo::dump(char *msg) {
  cout<<msg;
  cout<<"\n i : "<<i<<" ("<< &i<<")"<<endl;
  cout<<" ip : " <<ip<<" ("
      << *ip<<")"<<endl;
}

void main() {
 cout<<"\n\nStart...."<<endl;
 demo one(5,"for object one");
 demo two(12,"for object two");
 demo three=two;
 one.dump("Object one : ");
 two.dump("Object two : ");
 three.dump("Object three : ");

 one=two;

 one.dump("Object one : ");
 two.dump("Object two : ");
 three.dump("Object three : ");

 getch(); // To freeze display
}
```

Figure 5 - The operator=

operator= function is called but that the copy constructor is no longer called, proving what was said above. You will, however, notice that the normal constructor is called.

A better understanding

Once the copy constructor is included and the = operator overloaded, the class behaved as expected. We could have left it there and moved onto the next problem but I decided to do a little more investigation just to see how these functions were working. My main reason for doing this was to try and understand what was happening so I could get a better understanding of C++.

Up until this time I had been putting in copy constructors and operator= functions parrot fashion. I did some testing to ensure they worked. But once I was convinced they did, I just left them alone to do their job in the background. I reasoned that if I could see how they were working, how they handle objects which are passed to them and which they pass back out, I could gain a better understanding that would help my programming. The program shown in Figure 7 is the result.

The key is the data member struct time t. This is used to hold the time at which the instance was created. It is important that we do not simply copy the time from one instance to another. That is why both the copy constructor and the operator= function call gettime(&t), so that the current time is placed in the relevant instance. The alternative I tried was to use the *desc member function to *track* the instances but this proved to be a problem. If you simply copy the description from one instance to another they all end up looking the same.

If you run the program you should get output which looks similar to the sample shown in Figure 8, I have added the lines numbers for reference. Up to and including line 7 should be self explanatory. We are creating two instances of demo and dumping them. You can see that they have been created at different times, because of the delay we have built into the program, and that the *desc pointers are pointing to two different addresses.

The next instance we create comes from the line demo three=two. What we are saying is: create a new instance called three but copy the data members from instance two. To perform the copy, the program will call the copy constructor or a default one if we do not supply any.

```
//: EXE004.CPP - Copy Constructor/operator=
#include <conio.h>
#include <iostream.h>
#include <dos.h>

class demo {
  char *desc;
  struct time t;
public:
  demo(char *msg="Default Description");
  // Constructor
  demo(demo &dmo);  // Copy Constructor
  demo operator=(demo &dmo);
  // Overloading = operator
  void dump(char *msg="Dumping demo object :
");
};

demo::demo(char *msg) {
  cout<<"In demo constructor "<<msg<<endl;
  desc=msg;
  gettime(&t);
}

demo::demo(demo &dmo) {
  cout<<"In copy constructor"<<endl;
  dmo.dump("Copy Constructor : ");
  desc=dmo.desc;
  gettime(&t);
  this->dump("*this object
            (copy constructor) : ");
}

demo demo::operator=(demo &dmo) {
  cout<<"In overloaded = function"<<endl;
  dmo.dump("Operator = : ");
  desc=dmo.desc;
  gettime(&t);
  this->dump("*this object
            (operator=) : ");
  delay(500);
  return *this;
}

void demo::dump(char *msg) {
  cout<<msg;
  cout<<"Object Created at ";
  cout<<(int)t.ti_hour<<":"<<(int)t.ti_min
      <<":";
  cout<<(int)t.ti_sec<<":"<<(int)t.ti_hund
      <<endl;
  cout<<" "<<desc<<" (" << &desc
      << ")" <<endl;
}

void main() {
 cout<<"\n\nStarting program"<<endl;
 demo one("Object one");
 delay(500); // To make sure objects are
             // created at different times
 demo two("Object two");
 one.dump("Object one : ");
 two.dump("Object two : ");

 cout<<"create three from two"<<endl;
 delay(500);
 demo three=two;

 cout<<"copy two to one"<<endl;
 delay(500);
 one=two;

 one.dump("Object one : ");
 two.dump("Object two : ");
 three.dump("Object three : ");

 getch(); // To freeze display
}
```

Figure 7 - Following objects

```
Start....
In demo constructor for object one
In demo constructor for object two
In demo copy constructor
Object one :
i : 5 (0x3c6cfff2)
ip : 0x3c6cfff2 (5)
Object two :
i : 12 (0x3c6cffee)
ip : 0x3c6cffee (12)
Object three :
i : 12 (0x3c6cffea)
ip : 0x3c6cffea (12)
In overloaded = function
In demo copy constructor
Object one :
i : 12 (0x3c6cfff2)
ip : 0x3c6cfff2 (12)
Object two :
i : 12 (0x3c6cffee)
ip : 0x3c6cffee (12)
Object three :
i : 12 (0x3c6cffea)
ip : 0x3c6cffea (12)
```

Figure 6 - Sample output from program in Figure 4

What we are interested in is how the copy constructor manipulates the instances. The copy constructor is passed a reference to an instance. We suspect that this will be a reference to the instance on the right hand side of the equals sign, ie two. This is proved by the lines 10 and 11 which reference instance dmo. These show us the time the instance was created. The fact that we are referencing instance two is further ascertained by seeing that the *desc pointer is the same as the one for instance two, although we should not rely solely on pointers until we are confident in what we are doing.

Next we dump the instance pointed to by *this, in lines 12 and 13. We do this after putting the current time into this instance. This shows us that a new instance was created as part of the copy constructor and that it is the current instance as it is pointed to by *this.

So far so good

We have managed to follow an instance through a copy constructor and got the results we probably would have predicted. Now what about the operator= function? If you look at the program you can see the next instance manipulation we code is one=two. This time we are copying an existing instance of an object. You will see from line 15 that the operator= function is called to perform this operation. Like the copy constructor this function is passed a reference to an instance.

We could reason that this will be a reference to an instance on the right hand side of the equals sign. By looking at the time the instance was created we can confirm this

C/C++ without Object Master

Introducing a powerful new way of looking at code development.

Open your eyes to Object Master™, the most innovative programming tool available on the market today. With its powerful editors and intuitive windows, Object Master gives you the unsurpassed freedom to develop code quickly and accurately.

A Real Eye Opener

Use Object Master's unlimited number of browser windows to access code components and display their definitions, ready for editing. All changes you make in the browser windows are automatically displayed throughout the environment, consistently ensuring current and accurate code. With the browser windows, you can also view a class list displaying the hierarchical relationships between classes, and use premade templates to create classes and methods.

True browser windows let you see code like you never have before.

Picture Perfect Code

While the browser windows let you edit specific pieces of code, Object Master's File editor gives you a full-file view of your code, allowing you to quickly perform universal edits.

To help you identify important pieces of code easily, Object Master colour codes and formats language elements. With a single keystroke, Object Master will look up parameters for methods or functions contained in the project and paste them directly into your source code.

"Object Master is a tremendous tool. The different ways of browsing your source code make it easy to get a handle on how your program is supposed to work. It's almost impossible to say enough good things about it."
–Mike Rockbold, QA Engineer at Symantec.

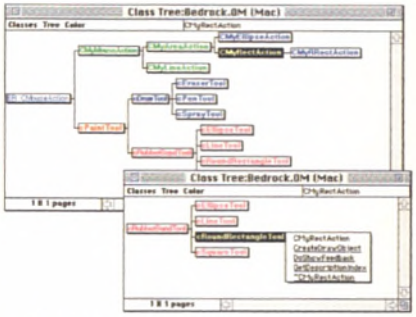
Improved Insight

Don't worry about the physical location of your code – Object Master parses all files and maintains a data dictionary of project components. The dynamic environment updates your entire project automatically as you edit without compilation!

And, Object Master works seamlessly with all major compilation systems to trigger compilation and receive compilation errors without switching to the compiler environment.

Object Master provides you with all the editing and navigational capabilities you require to write code productively. For example, the Class Tree window graphically displays your project's class hierarchy and allows you to expand and collapse "branches" and open multiple windows displaying specific code components.

"I use the browsing and navigational capabilities of Object Master and I couldn't live without them."
–Chris Knepper, Software Engineer at Taligent



This is where the similarities end.

Object Master for Windows works with all major C/C++ compilation systems, including VC++, Symantec C++, Borland C++, and others. The price is just £195 + shipping & VAT. Object Master for the Macintosh, which supports MPW, Symantec, and Metrowerks CodeWarrior for 68K and PowerPC systems, works with the C, C++, Pascal, and Modula-2 programming languages. The price (for C/C++) starts at £195 + shipping & VAT.

Published by ACI UK Ltd, Barons Court, Manchester Road, Wilmslow, Cheshire SK9 1BQ. Tel (01625) 536178. Fax (01625) 536497. AppleLink: ACI.UK

Dealer enquiries welcome. Distributed by
Full Moon Software Distribution Ltd,
PO Box 862, MAIDENHEAD, Berks.
SL6 0QJ. UK. Tel: (01628) 660242
Fax: (01628) 666084
Email: sales@ctalk2.exnet.com
AppleLink: FMSDL

A few words from our competitors:

Prtablty.

Prtbilty.

Portblty.

Portabit.

No matter how they misspell it, it's still not portability. At Zinc, we understand that porting the last 20% of your code takes 80% of your time.

Only Zinc offers complete portability.

With Zinc® Application Framework™ you can develop on the platform you prefer. And since Zinc is the only one that delivers 100% portability, you'll have your application on other platforms as fast as you can recompile. It's part of what makes Zinc the most productive—and affordable—tool a programmer can use.

Productivity that leads to opportunity.

Portability is just one road you'll find a little easier. Zinc zips through tedious tasks with C++ object orientation and a unique visual development tool. Globalizing your application is as easy as translating the text. And Zinc is the only product that supplies 100% of the source code.

It all adds up to productivity. Which means more profitability. Which is a concept we probably don't have to spell out for you. For free information and demonstration software, call :

+44 (0) 181 855 9918

Or contact us via fax at: +44 (0)181 316 7778.
In the U.S. call: +1 801 785 8900. Or fax: +1 801 785 8996.
In Asia call: +81 (052) 733 4301. Contact Zinc electronically at: info@zinc.com. Access our BBS at: +44 0 (181) 317 2310.

PCWEEK LABS ANALYST'S CHOICE JAN. 30, 1995

"Developers seeking easy delivery of GUI applications on DOS, Windows, OS/2, Macintosh and Unix platforms, or pursuing international markets will find Zinc their best option by far."

INFOWORLD February 6, 1995

"Zinc came closest of all the products we tested to our ideal of portability: Just copy the code to the target machine, then recompile and relink the application... In short, Zinc did a great job."

Z i n c

NO LIMITS

```
(1)   Starting program
(2)   In demo constructor Object one
(3)   In demo constructor Object two
(4)   Object one : Object Created at 20:30:56:22
(5)    Object one (0x4359fff0)
(6)   Object two : Object Created at 20:30:56:71
(7)    Object two (0x4359ffea)
(8)   create three from two

(9)   In copy constructor
(10)  Copy Constructor : Object Created at 20:30:56:71
(11)   Object two (0x4359ffea)
(12)  *this object (copy constructor) : Object Created at 20:30:57:21
(13)   Object two (0x4359ffe4)
(14)  copy two to one

(15)  In overloaded = function
(16)  Operator = : Object Created at 20:30:56:71
(17)   Object two (0x4359ffea)
(18)  *this object (operator=) : Object Created at 20:30:57:70
(19)   Object two (0x4359fff0)

(20)  In copy constructor
(21)  Copy Constructor : Object Created at 20:30:57:70
(22)   Object two (0x4359fff0)
(23)  *this object (copy constructor) : Object Created at 20:30:58:20
(24)   Object two (0x4359ffde)
(25)  Object one : Object Created at 20:30:57:70
(26)   Object two (0x4359fff0)
(27)  Object two : Object Created at 20:30:56:71
(28)   Object two (0x4359ffea)
(29)  Object three : Object Created at 20:30:57:21
(30)   Object two (0x4359ffe4)
```

Figure 8 - Sample output from program in Figure 7

theory (line 16). The fact the pointer also matches with what we would expect to see makes us doubly sure (line 17).

As part of the `operator=` function we also dump the `*this` instance in lines 18 and 19. By looking at the time this instance was created we can see that it's a new instance as it was created after all the other instances to date. So far the `operator=` function has acted in exactly the same way as the copy constructor function. Line 20 is interesting as it shows something a little unexpected. For some reason the copy constructor has also been called. By looking at line 21 you can see that it has been passed a reference to the new instance that the `operator=` function has just created. The copy constructor, as we know, creates a new instance and we can see it happening again here. The `*this` instance is obviously a new instance as it is created after all other instances to date as shown by lines 23 and 24.

But we still have the question as to why the copy constructor was called as part of the `operator=` function. The last three dumps provide us with a clue. You will see that the `*this` instance created in the last call to the copy constructor no longer exists. This must mean it was a temporary instance. The reason for this is because it was *copied out* via the `return *this` statement at the end of the `operator=` function. To do this a copy is made of the `*this` instance which needs to invoke the copy constructor to ensure that the instance is copied correctly. Therefore, once the copy has been done, the `*this` instance disappears.

Of course, in normal circumstances, we would not see this as the copy constructor would make the instances look the same and it would only be by studying the pointer locations that we would be able to follow what is going on. You might wonder why a temporary instance is created in the `operator=` function but not for the copy constructor function. The answer is that a copy constructor does not have a return type in its prototype so that there is no return type as part of the function. The `operator= function` does have a return type in its prototype and it is the `return * this` that invokes the copy constructor. In summary, when the `operator=` function does a `return *this` a copy is made of the instance and the `*this` instance disappears. ■

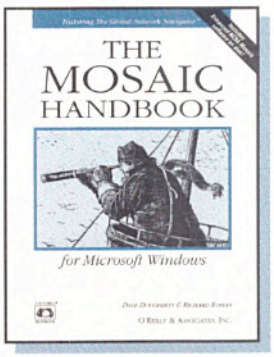
All you need to know about navigating the Internet with Mosaic

Mosaic is an important application that is becoming instrumental in the growth of the Internet. These books, created for Microsoft Windows, X, and the Macintosh, introduce you to Mosaic and its use in navigating and finding information on the World Wide Web (WWW).

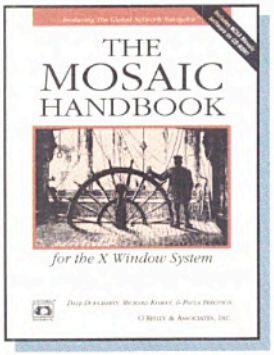
It shows you how to use Mosaic to replace some of the traditional Internet functions like FTP, Gopher, Archie, Veronica, and WAIS. For more advanced users, the books describe how to add external viewers to Mosaic (allowing it to display many additional file types) and how to customize the Mosaic interface, such as screen elements, color, and fonts.

The Microsoft and Macintosh versions come with Enhanced NCSA Mosaic on diskettes; the X Window version comes with NCSA Mosaic on CD-ROM. All three books come with a subscription to The Global Network Navigator(GNN®), the interactive guide that makes the Internet more enjoyable and easier to use.

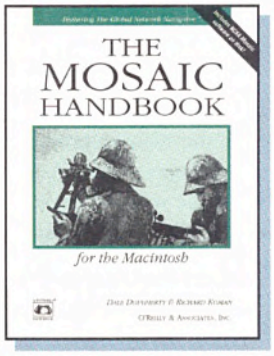
Another new O'Reilly book on a much-requested topic is *Managing Internet Information Services: World Wide Web, Gopher, FTP, and more*. It describes in detail how to set up information services to make them available over the Net. It begins by discussing why a company would want to provide Internet services and how to select which services to provide. Most of the book describes how to set up email services and FTP, Gopher, and World Wide Web servers. This book will be published in December.



Dale Dougherty & Richard Koman
1st Edition October 1994, 230 pages
ISBN 1-56592-094-5. With two diskettes containing Enhanced NCSA Mosaic for Windows, V1.0



Dale Dougherty, Richard Koman, & Paula Ferguson
1st Edition October 1994, 288 pages
ISBN 1-56292-095-3. With CD-ROM containing NCSA Mosaic for the X Window System, V2.4



By Dale Dougherty & Richard Koman
1st Edition October 1994 198 pages
ISBN 1-56592-096-1. With diskette containing Enhanced NCSA Mosaic for Macintosh, V1.0

INTERNATIONAL THOMSON PUBLISHING

Available from:

International Thomson Publishing Europe

Berkshire House, 168-173 High Holborn, London WC1V 7AA, United Kingdom

Tel: 44-71-497-1422 • Fax: 44-71-497-1426 • Email: ora.orders@itpuk.co.uk

CIRCLE NO. 056

All the time in the world...

'Time is important' says it all really. We're always trying to keep track of time. Once we've done that we wonder where it all went. **Michael Marshall** puts time into objective...

Programs, like people, often need to keep track of time, especially when tasks need to be scheduled for execution. Clocks need to know when a minute has elapsed; word processors need to know when to auto-save their documents; communications programs need to know when a time-out occurs. Microsoft Windows programs use *timers* to track intervals of time. With current versions programmers have a choice between two different interfaces for timers. This article will discuss these timer interfaces, their strengths and weaknesses and how they are used. I will also present a C++ class to encapsulate timer functionality and present a single interface for all versions of Windows.

Message timers

Windows 2.0 introduced the message timer APIs, `SetTimer()` and `KillTimer()`. I'll call them message timer APIs because of their use of the `WM_TIMER` message.

```
UINT SetTimer
(
 HWND hwnd, UINT id,
 UINT interval,
 TIMERPROC callback
);
BOOL KillTimer
(
 HWND hwnd, UINT id
);
```

To create a timer, you specify an ID (unique to your program), an interval (in milliseconds) and a window handle and/or a callback function to receive the timer events. The return code depends on the `hwnd` parameter. If `hwnd` was not NULL, `SetTimer()` returns non-zero on success; with a NULL `hwnd`, `SetTimer()` returns the non-zero handle of the new timer on success (the ID parameter is ignored).

When the interval has passed, a timer event is generated; how it is handled depends on how the timer was created: If callback was not NULL, it is called with the details of the timer event. If callback is NULL, and hwnd is not NULL, then a `WM_TIMER` message is posted to the window. If both callback and `hwnd` are NULL, then the `WM_TIMER` message is sent to the application message queue.

To destroy the timer, simply call `KillTimer()`. If you created the timer with a NULL `hwnd`, pass the handle `SetTimer()` gave you as the ID parameter. Sounds simple, doesn't it? Well, of course, it's not. First of all, there are only a limited number of timers available system-wide (32 at any one time), so you should always verify that `SetTimer()` actually succeeded.

You can request timer events with intervals as low as 1 millisecond, but in reality the lowest interval you can use is 55 milliseconds, which corresponds to about once every hardware clock tick (18.2Hz).

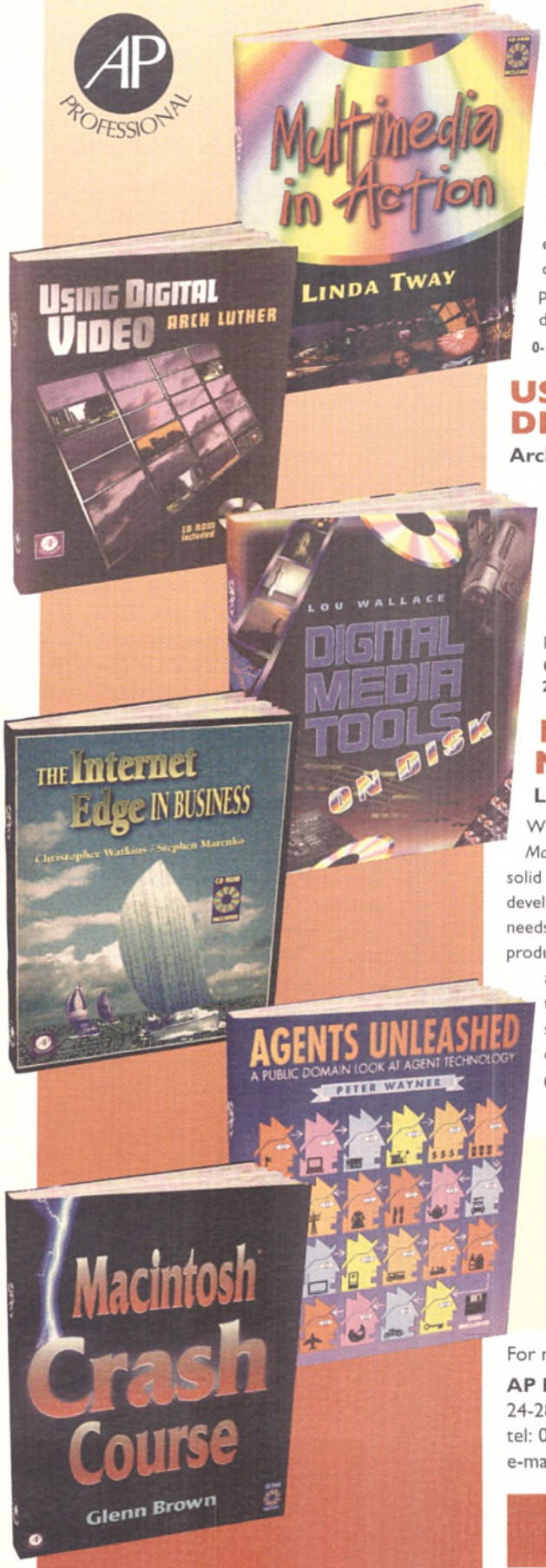
The resolution parameter is a 'hint' to the system as to how accurate the timing needs to be

```
// TIMER.H
// TTimer class

#ifndef _TIMER_H
#define _TIMER_H

class TimerManager;
class TTimer
{
 friend class TimerManager;
public:
 TTimer( const TTimer &t );
 void operator=( const TTimer &t );
 virtual ~TTimer();
 virtual void execute( DWORD now ) = 0;
 virtual void enable();
 virtual void disable();
 virtual void setInterval( UINT intvl );
 UINT interval() const;
 BOOL isEnabled() const;
 DWORD expires() const;
protected:
 BOOL m_Enabled;
 UINT m_Interval;
 DWORD m_ExpireTime;
protected:
 TTimer( UINT interval );
};

#endif
```

Figure 1 - Definition of TTimer class

Power users are saying...

AP PROFESSIONAL



MULTIMEDIA IN ACTION

Linda E. Tway

Discover how award-winning multimedia titles have been developed for use in sales, training, education, entertainment, and referencing, with demos on the accompanying CD-ROM plus multimedia development software discussed in the book.

0-12-705250-X, May 1995, 300 pp., £29.95

USING DIGITAL VIDEO

Arch C. Luther

Written for the average computer user, this book and CD-ROM teaches the principles of digital video and audio and shows how anyone can incorporate video and audio into existing applications, electronic presentations, and information.

0-12-460432-3, February 1995, 290 pp., £24.95

DIGITAL MEDIA TOOLS

Lou Wallace

Written by the founder of *Digital Video Magazine (IDG)*, this book provides a solid background for the professional developer or sophisticated hobbyist who needs to use digital media in the production of multimedia or other applications. It discusses current technologies and concepts in software, and provides information on all the major hardware products.

0-12-732855-6, May 1995, 520 pp., £29.95

THE INTERNET EDGE IN BUSINESS

C. D. Watkins

Featuring a CD-ROM with a toolkit for business solutions to help get any DOS or MAC machine on-line, this book outlines what the Internet is and how it can enhance your business's efficiency and accessibility.

0-12-737840-5, June 1995, 300 pp., £19.95

AGENTS UNLEASHED

A Public Domain Look at Agent Technology

Peter Wayner

An invaluable tool for anyone interested in implementing agent-based net protocols, this book describes a model agent system and promotes a public domain standard. It introduces digital cash and opening shop on the net, and comes with a software agent on the enclosed disk.

0-12-738765-X, May 1995, 350 pp., £29.95

MACINTOSH CRASH COURSE

Glenn Brown

This book will show Macintosh power users what to do when things go wrong with their system, including how to overcome systems crashes, lock-ups, and various frustrating and cryptic error messages. It includes a CD-ROM full of shareware and freeware to help diagnose and repair these system failures.

0-12-135910-7, June 1995, 450 pp., £22.95

WATCH OUT FOR...

Graphics, Fuzzy Logic, and **Mathematica** CD-ROM Libraries, July 1995, £44.95

Migrating to Windows 95, June 1995, £24.95

Wireless for the Newton, April 1995, £24.95

Graphics Gems V, May 1995, £34.95

For more information and a complete catalogue contact Andy Crowther at:

AP PROFESSIONAL
24-28 Oval Road, London NW1 7DX, UK.
tel: 0171-267-4466. fax: 0171-485-4752
e-mail: app@apuk.co.uk

RESERVE YOUR COPIES TODAY BY PHONE OR E-MAIL!

CIRCLE NO. 057

Like WM_PAINT, Windows will only hold a single WM_TIMER message in its queue at any one time. If the system is busy for a while, some timer events will be 'lost' and WM_TIMER messages will only be delivered when the message queue is otherwise empty.

DLLs present an interesting problem. You'd think that creating a timer with SetTimer(NULL,0,interval,callback) should work fine, but in general it will *not* work. The reason is that timers require a message queue, even if a call-back function is used. When the timer is created, the message queue used will be that of the task active when SetTimer() is called; when that task terminates, the timer will be destroyed with it. The work-around is for the DLL to launch a dedicated application, which passes the DLL an hwnd to a hidden window to be used in a SetTimer() call. The dedicated application will always be around until the DLL is finished with it.

Multimedia Timers

The Multimedia extensions (incorporated into Win32), added new high-resolution timers that were not message-oriented. Before any timers are created, timeBeginPeriod() must be called to set the minimum timer period required. For example, if no timer will have a period less than 0.5 sec, you would call timeBeginPeriod(500). The timeGetDevCaps() API retrieves

```
// TIMER.CPP

#include "stdafx.h"
#include "timer.h"

const UINT TickSize = 100;  // 0.1s

class TimerManager {
public:
 TimerManager();
 ~TimerManager();
 void addTimer( TTimer *t );
 void removeTimer( TTimer *t );
 void processTick();
protected:
 UINT m_TimerId;  // Windows timer ID
 // List of active TTimer objects
 CPtrList  m_Timers;
};

static TimerManager _TM;

void CALLBACK _export TimerMgrCall-
back
(
 HWND, UINT, UINT, DWORD time
)
{
 _TM.processTick();
}

TimerManager::TimerManager()
{
}

TimerManager::~TimerManager()
{
 // Ensure the Windows timer is
destroyed
 if ( m_TimerId )
  KillTimer( 0, m_TimerId );
}

void TimerManager::addTimer( TTimer
*t )
{
 m_Timers.AddTail( t );
 if ( m_Timers.GetCount() == 1 )
 {
  // Create Windows timer now
  m_TimerId = ::SetTimer(0, 0, Tick-
Size,
              TimerMgrCallback);
  if ( !m_TimerId )
   AfxMessageBox("Unable to allocate
system timer"
            , MB_ICONSTOP );
 }
}

void
TimerManager::removeTimer(TTimer *t)
{
 POSITION pos = m_Timers.Find( t );
 ASSERT( pos != 0 );
 m_Timers.RemoveAt( pos );
 if ( m_Timers.IsEmpty() )
 {
  // Destroy timer
  // Don't need it right now
  ::KillTimer( 0, m_TimerId );
  m_TimerId = 0;
 }
}

void TimerManager::processTick()
{
 DWORD tickCount = GetTickCount();
 POSITION pos =
  m_Timers.GetHeadPosition();
 while ( pos )
 {
  TTimer *t = (TTimer*)m_Timers.Get-
Next
                    (pos);
  // pos now indicates the NEXT timer
  ASSERT( t != 0 );
  if ( t->m_ExpireTime <= tickCount )
  {
   // Process this expired timer
   t->execute( tickCount );
   // Okay if the timer is disabled
   // Iteration not affected?
   if (t->isEnabled() )
   {
    while(t->m_ExpireTime <= tick-
Count)
     t->m_ExpireTime += t->m_Inter-
val
                * TickSize;
   }
  }
 }
}

TTimer::TTimer( const TTimer &t )
{
 m_Interval = t.m_Interval;
 m_Enabled = FALSE;
 m_ExpireTime = 0;
}

void TTimer::operator=( const TTimer &t
)
{
 m_Interval = t.m_Interval;
 m_Enabled = FALSE;
 m_ExpireTime = 0;
}

TTimer::~TTimer()
{
 if ( m_Enabled )
  disable();
}

void TTimer::enable()
{
 if ( !m_Enabled )
 {
  if ( m_Interval == 0 )
   return;
  // Calculate the expiration time
  m_ExpireTime = ::GetTickCount() +
           m_Interval * TickSize;
  // add to active list
  _TM.addTimer( this );
  m_Enabled = TRUE;
 }
}

void TTimer::disable()
{
 if ( m_Enabled )
 {
  // remove from active list
  _TM.removeTimer( this );
  m_ExpireTime = 0;
  m_Enabled = FALSE;
 }
}

void TTimer::setInterval( UINT intvl )
{
 if ( intvl == 0 )
  return;
 if ( m_Interval != intvl )
 {
  if ( m_Enabled )
  {
   // Adjust expire time
   if ( intvl > m_Interval )
    m_ExpireTime +=
     (intvl - m_Interval) * TickSize;
   else
    m_ExpireTime -=
     (m_Interval - intvl) * TickSize;
  }
  m_Interval = intvl;
 }
}

UINT TTimer::interval() const
{
 return m_Interval;
}

BOOL TTimer::isEnabled() const
{
 return m_Enabled;
}

DWORD TTimer::expires() const
```

Figure 2 - Implementation of TTimer class

get tooled up!

If you are a software developer looking for the right tools for your trade, if you need to test different products before making decisions, if you want live demonstrations of a range of development products, if you want to meet and talk to the world's leading software gurus and if you want to sit in on leading edge developer sessions... then make a date in your diary to get to the Royal Horticultural Halls in London's Westminster on June 8-9 to visit the **EXE Software Developers' Show** – *The most important developer event of the year!*

Brought to you by the number one software development title, **EXE Magazine**, this new event will have over 70 exhibitors from across the development business, demonstrating everything from client servers to relational databases, GUI tools to dongles and compilers to networking systems.

Running parallel to the event is an exciting seminar and workshop programme devised by the *EXE* team to meet your needs as a developer, bringing together the gurus in software development from both Europe and the United States. Also, the *Software Developers' Test Arena* – a chance for you to test a variety of products in their operating environments. Plus a continuous series of vendor presentations.

Whether UNIX or PC based and for whatever language, the **EXE Software Developers' Show** will be the focal point in the developer's diary for 1995... make sure you are there!

Organised by EXE Magazine

Admission £10 on the door

Complimentary tickets and further information including details of seminars and workshops available by faxing or posting the coupon.

Acucobol
Aladdin
Applied Conversion Technology
Bits Per Second
Blink Inc
Borland
Citadel Software
Computer Associates
Computer Manuals
Continuus Software ltd
Data Encryption Systems
Direct Technology
Fast Electronic
Gupta
Hypersoft Europe
ICS
Imperial Software Technology
IPL Information Processing
Logic Programme Associates
Lotus
Mercury Interactive
Oxford Computer Consultants
Parks Bookshop
PC Bookshop
Powersoft
Programme Analysers
Prolog Development Centre
Professional Software
QBS
Rainbow Technologies
Real Techniques & Methods
Revelation Technologies
Rhino Publishing
Scientific Software
Select Software
Sequiter Software
Shapewear
Software Practice
Software Security
Softwerk
Staria
Symantec
System FX
System Science
User Interface Technologies

at the

The main event for software developers in 1995

EXE
software
developers'
show

8-9 june '95
Royal Horticultural Halls
Westminster · London

FREE TICKET OFFER FREE TICKET OFFER FREE TICKET OFFER

Please send me details of the EXE Software Developers' Show

Name ..... Title .....

Company .....

Address .....

.....

Telephone ..... Number of tickets required .....

Send to: Tim Macpherson, The Exe Software Developers' Show, Centaur Exhibitions, 50 Poland Street, London W1V 4AX, or fax on 0171 287 0710.

CIRCLE NO. 058

information about the legal values for `timeBeginPeriod()`.

The `timeSetEvent()` API is used to create a timer. The parameters to `timeSetEvent()` are similar to `SetTimer()`, except that no window handles are involved, only call-backs are used. Timers can be created as one-shot (automatically destroyed when they expire), or periodic; `SetTimer()` only creates periodic timers. The resolution parameter is a 'hint' to the system about how accurate the timing needs to be. It should be as large as possible to reduce system load.

Rules of multimedia

Multimedia timer callbacks operate under much stricter rules than the message timer callbacks. Under Windows 3.1 the rules are even tighter. The callback must be implemented in a DLL. The timer events occur at interrupt times. Only a small number of API functions are available for use in callback functions. Under Win32, the DLL and API restrictions are removed and the timer events are handled in a separate thread of execution.

All periodic timers must explicitly be destroyed using the `timeKillEvent()` function When all timers have been destroyed, the `timeEndPeriod()` function should be called with the same parameter used in `timeBeginPeriod()`. An important note for multimedia timers is that

It is unfortunate that proper time functionality is only realistically available under Win 32

`timeBeginPeriod()/timeEndPeriod()` pairs can be nested, but each `timeBeginPeriod()` must be matched with the appropriate `timeEndPeriod()`.

The *TTimer* Class

Okay, now that we've reviewed the available Windows facilities, let's get to work and hide all of that complexity behind a C++ class, which we'll call `TTimer`. The great advantage of encapsulating the timer code this way is that the same interface will work under all version of Windows, from 3.1 up.

```
// MSGTIMER.H - TMsgTimer and
// TOneShotTimer definitions

#ifndef _MSGTIMER_H
#define _MSGTIMER_H

#include "stdafx.h"
#include "timer.h"

class TMsgTimer : public TTimer
{
public:
 TMsgTimer(UINT intvl, CWnd *win,
       UINT msg );
 void execute( DWORD now );
protected:
 CWnd *m_Win;
 UINT m_MsgId;
};

class TOneShotTimer: public TMsgTimer
{
public:
 TOneShotTimer(UINT intvl, CWnd *win,
        UINT msg );
 void execute( DWORD now );
protected:
};

#endif
```

Figure 3 - Definition of MSGTimer and TOneShotTimer classes

```
// MSGTIMER.CPP

#include "stdafx.h"
#include "msgtimer.h"

TMsgTimer::TMsgTimer(UINT intvl,
        CWnd *win,
        UINT msg)
 : TTimer( intvl )
{
 ASSERT( win != 0 );
 m_Win = win;
 m_MsgId = msg;
}

void TMsgTimer::execute( DWORD now )
{
 // Post the message to the window
 m_Win->PostMessage(m_MsgId, 0,
          LPARAM(now));
}

TOneShotTimer::TOneShotTimer(UINT
intvl,
                CWnd *win,
                UINT msg )
 : TMsgTimer( intvl, win, msg )
{
}

void TOneShotTimer::execute( DWORD
now )
{
 disable();  // disable self
```

Figure 4 - Implementation of MSGTimer and TOneShotTimer

The `TTimer` class implemented in this article has a resolution of 1/10th of a second. Windows is capable of much finer resolution, but for most uses (eg autosave timer, clock, process timer) this value balances the requirements of accuracy and low overhead. Too fine a resolution results in a lot of extra work with dubious benefits.

Virtual time piece

The `TTimer` class has a pure virtual function, `execute()`, which is called to process a timer event. This function should be implemented in derived classes to perform appropriate actions, such as setting a flag or posting a message. Examples will be given later.

Any number of `TTimer` objects can exist at any time. The only limits are available memory and common sense.
`TTimer` objects have enabled and disabled states. When a timer is disabled (the default state), no timer events are generated for it. No resources are allocated either.

Only a single Windows timer is located for all `TTimer` objects currently active. If no `TTimer` object is enabled, the Windows timer will be destroyed until a `TTimer` is enabled. This helps to ensure that only resources currently needed are allocated, with the caveat that the program might not be able to allocate a windows timer when needed.

Objects may be copied and assigned, but the copies will always be initially disabled.

The `TTimer` class is defined in the file TIMER.H (see Figure 1). The public interface is defined first, followed by the private implementation. The implementation of `TTimer`, TIMER.CPP, is in Figure 2.

Where the work is done

The `TTimer` class itself doesn't do much work. Its primary purpose is to maintain state information. The real work for the `TTimer` class is performed by the `TimerManager` class, which is fully defined and implemented in the TIMER.CPP file. This maintains a list of all enabled `TTimer` objects. We're not interested in disabled timers, for reasons which will become clear shortly.

When a `TTimer` is enabled, the first thing that happens is the expiration time is calculated, using the current Windows time and the timer interval. Next, the timer is added to the active timers list. If this timer is the only timer on that list, `TimerManager` allocates a Windows timer with a frequency of 0.1 seconds.

When a `TTimer` is disabled, it is removed from the active timers list. If the list is now empty, then the Windows timer is destroyed. When a windows timer event

occurs, each timer in the active list is checked against the current time. If a timer has expired, then the timer's `execute()` function is called with the current time as an argument. If the timer is still enabled (the `execute()` function might have disabled the timer), then a new expiration time is recalculated. We have to be careful while iterating through the timers list in case an expired timer has been removed. Because each tick of the Windows timer results in all `TTimers` in the `TimerManager's` list being checked, we want to keep that list as short as possible. To that end, only *enabled* `TTimer` objects are tracked.

If a large number of enabled `TTimer` objects are needed by an application, greater efficiency might be gained by keeping the enabled timers list sorted by expiration time. When checking for expired timers, the first unexpired timer terminates the search. Personally, I think the extra work needed to maintain the sorted list would counter the benefits. The reader is encouraged to experiment.

This implementation of `TimerManager` uses the message timer API's because they are supported unchanged in all current versions of Windows. If you're only targeting Win32 systems, then the multimedia timers will give you better performance and accuracy. The changes required are limited to the `TimerManager` class and do not affect the `TTimer` interface.

```
// time1.h : main header file for the
// TIME1 application

#ifndef __AFXWIN_H__
 #error include 'stdafx.h' before includ-
ing this file for PCH
#endif

#include "resource.h"       // main sym-
bols
#include "timer.h"
#include "msgtimer.h"

extern UINT  m_10SecMsg;
extern UINT  m_MinuteMsg;
extern UINT  m_OneShotMsg;
extern TMsgTimer *m_10SecTimer;
extern TMsgTimer *m_MinuteTimer;
extern TOneShotTimer *m_OneShot-
Timer;

/////////////////////////////////////

// CTime1App:
// See time1.cpp for the implementation
// of this class

class CTime1App : public CWinApp
{
public:
 CTime1App();


// Overrides
 virtual BOOL InitInstance();

// Implementation

 //{{AFX_MSG(CTime1App)
 afx_msg void OnAppAbout();
 //}}AFX_MSG
 DECLARE_MESSAGE_MAP()
```

Figure 5 - Main header file for Time class

```
// time1.cpp : Defines the class behav-
iors
// for the application.

#include "stdafx.h"
#include "time1.h"

#include "timefram.h"

#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE
           THIS_FILE[] = __FILE__;
#endif

UINT  m_10SecMsg;
UINT  m_MinuteMsg;
UINT  m_OneShotMsg;
TMsgTimer *m_10SecTimer;
TMsgTimer *m_MinuteTimer;
TOneShotTimer *m_OneShotTimer;

// CTime1App

BEGIN_MESSAGE_MAP(CTime1App,
CWinApp)
 //{{AFX_MSG_MAP(CTime1App)
 ON_COMMAND(ID_APP_ABOUT, OnAp-
pAbout)
 //}}AFX_MSG_MAP
END_MESSAGE_MAP()

// CTime1App construction

CTime1App::CTime1App()
{
}

// The one and only CTime1App object

CTime1App NEAR theApp;

// CTime1App initialization

BOOL CTime1App::InitInstance()
{
 // Set dialog background color to gray
 SetDialogBkColor();

 CTimeFrame *frame = new CTime-
Frame;
 frame->LoadFrame( IDR_MAINFRAME );
 m_pMainWnd = frame;
 frame->ActivateFrame();

 // Register timer messages
 m_10SecMsg = RegisterWindowMessage
        ( "Time1-10Sec" );
 m_MinuteMsg = RegisterWindowMes-
sage
         ( "Time1-Minute" );
 m_OneShotMsg = RegisterWindowMes-
sage
         ( "Time1-OneShot" );

 // Create the timers
 m_10SecTimer = new TMsgTimer
             (
              100, m_pMainWnd,
              m_10SecMsg
             );
 m_MinuteTimer = new TMsgTimer
              (
               600, m_pMainWnd,
               m_MinuteMsg
              );
 m_OneShotTimer = new TOneShotTimer
              (
               300, m_pMainWnd,
               m_OneShotMsg
              );

 return TRUE;
}

// App command to run the dialog
void CTime1App::OnAppAbout()
{
```

Figure 6 - Definition of Time class

Accurate timings

Regardless of what version of Windows you are using and which timer implementation is chosen, it is pretty important to minimise the amount to work done in a timer `execute()` function. Spending too long can result in inaccurate timings and loss of timer events. The most practical solution is to just send a message via `PostMessage()` (*not* `SendMessage()`!) requesting work to be done, which will be scheduled in the normal way.

There will be a practical limit to the number of enabled `TTimer` objects an application can have at any time before timer accuracy is impacted; this limit will be determined by the speed of the CPU and the workload of the system.

The source code was written using Visual C++ 1.5 and MFC 2.5 under Windows 3.1, but the same code will port easily to Borland C++ with appropriate changes to the TimerManager class (to use BIDS and OWL, for example).

Using TTimer

In order to use the `TTimer` fully, a derived class or two will be needed to implement specific timer functionality. Figures 3 and 4 list MSGTIMER.H and MSGTIMER.CPP respectively. `TMsgTimer` inherits from `TTimer`. When it expires, it sends a specific Windows message to a window. The message number to be used can be based on `WM_USER` or generated by `RegisterWindowMessage()`. The `LPARAM` argument to the message is the time passed

SYSTEM SCIENCE

SOFTWARE DEVELOPMENT TOOLS FOR WINDOWS, DOS, UNIX, & OS/2

Windows Development

Doc-To-Help - help authoring	£275
Robo-HELP - help authoring	£365
InstallSHIELD - installation utility	£319
WinMaker Pro 6.0 - Win & NT	£395
PC-Logo Windows	£60
Tools.h++, Linpack.h++, Math.h++	£call
Visual SmallTalk - ***new***	£945
Zinc new 4.0 for Win & NT	£call
Graphics Server SDK new version	£225
SmartHeap Win or DOS	£345
TCP/IP Development tools	£call
CodeBase or CodeBase++	£239
Greenleaf Commlib 5.0 Prof	£345
C-Tree Plus - with source	£595

VB Value

Disctinct TCP/IP Visual Ed	£225
Spyworks by Desaware	£110
Spread/VBx	£165
TrueGrid Pro	£110
VB Assist	£135
Designer Widgets or Data Widgets	£95
VB Tools 4.0	£110
QuickPak Professional for Windows	£139
PDQ Comm Windows	£95
Grid/VBx, Tab/VBx, and Tab/Pro - ***new***	£call
Win/Widgets VB	£139
Xref VB, HighEdit, etc	£call

Windows 32

SmartHeap Win/32	£525
Win Widgets/32	£275

Mathematics & Science

- Mathematica - DOS, Win, Mac, Workstations, Education & Student pricing.
- Derive - new ver 3.0 - call for upgrades.
- Origin windows scientific plotting.
- Lindo & What's Best Linear Programming

...many more for maths & stats.

UNIX

- SCO UNIX operating & dev. systems.
- SCO Open Desktop, TCP/IP and NFS.
- FoxPro for SCO UNIX (NEW)
- PC TCP, NFS and Xwindow servers

For FORTRAN

Lahey FORTRAN 90 - new	£625
MS FORTRAN-77 5.1 - DOS	£145
MS FORTRAN Powerstation - DOS	£229
MS FORTRAN Powerstation - NT	£399
IMSL Numerics for MS Fortran	£call
IMSL Exponent Graphics for NT	£595

Cross Development

Hi-Tech C Cross - many targets	£call
Introl C Cross - many targets	£call
2500AD Cross Asm & Simulators	£call

Phar Lap

FrontRunner - run VC++ 2.0 from Win	£95
TNTIDOS Extender SDK - 32 bit	£329
286IDOS Extender SDK - 16 bit	£329

Microsoft

Visual C++ 2.0 CD, & VC++ 1.5 16-bit	£299
Visual C++ 2.0 CD upgrades MS C/C++	£149
Visual C++ for MIPs or Alpha NT	£595
Visual C++ Cross for MAC target	£call
Visual Basic Windows PE	£229
TEST for Windows new 3.0	£465
Macro Assembler 6.11- NT or DOS	£110
Access/Access Dev Kit	£289 / £295
SQL Workstation System - NT	£415
Project for Windows	£375
Office and Office Professional	£call
SourceSafe Version Control	£call

(See FORTRAN Section)

Editors and Tools

CodeWright Prog Editor Win	£159
CodeWright Fusion for MS VC++	£99
Multi-EDIT Pro - DOS	£129
PC Lint C/C++ - Lint code checking	£149
Flowcharting & diagramming tools	£call
PC-Install DOS or Win	£call
DemoShield - Build Windows Demos	£call
Kedit for Windows - ***new***	£129

MKS

MKS Toolkit of Unix tools	£210
Source Integrity (ex RCS) ***New ver***	£315
MKS Lex & Yacc for C, C++ & TP	£215
Internet AnyWhere - off line news & Mail	£99

For Windows/Dos, OS/2 & NT, call for multi-user licences on all MKS products.

QUALITY IN CUSTOMER SERVICE AND SUPPORT

0171 833 1022

FAX 0171 837 6411

3-5 CYNTHIA STREET
LONDON N1 9JF

SYSTEM SCIENCE

Your first choice for better development tools

Watcom

Watcom C/C++ 10.0 CD-ROM ***(special)***	£155

Printed documentation is also available

Watcom Fortran-77[32] - DOS,OS/2, NT	£375
Watcom SQL - DOS,Win,OS/2	£229

SQL Servers for Win, Netware, NT or DOS

VX.REXX/VX.REXX Client Server-OS/2	£call

Borland

Borland C++ 4.5 - Win, NT & 95	£289
Borland C++4.5 & Dbase Frameworks	£415

Upgrades & Comp. Upgrades available

Power Pack for DOS	£69
Borland C++OS/2 new version 2.0	£325
Delphi - ***special pricing***	£199
Visual Solutions Pack of Vbxs	£59
Turbo Assembler	£59
Borland Pascal & Turbo Pascal	£call
dBASE Win or dBASE Dist Kit	£call

Nu-Mega

BOUNDS-CHECKER Win/32	£195
BOUNDS-CHECKER Win	£195
BOUNDS-CHECKER Win/NT	£195
Soft-ICE/W with 32-bit functions	£295

Call for savings on Nu-Mega bundles

PVCS Specialists

- PVCS Version Manager is available for Win, Win NT, OS/2 and many Unix platforms in single user and work-group configurations for all development teams.

OS/2

OS/2 Warp for Windows - 3.5" or CD	£Call
OS/2 Warp Win-OS/2 - CD/3.5"	£99/£115
IBM C Set++ - CD/3.5"	£345/£385
Visual Age 2.0 for OS/2 or Windows	£call

CLIENT/SERVER

Btrieve Development DOS or Windows	£395
Btrieve Client Engines DOS or Windows	£645
Scalable SQL for Netware	£call
PowerBuilder Desktop 4 ***(special)***	£call
Crystal Reports Std or Pro - 4.0	£139/£289

CASE

EasyCASE Professional Win 4.2	£695
EasyCASE DataBASE Eng for SQL	£call
InfoModeller Desktop	£call

NEWS

- Heap Agent - new from MicroQuill, for debugging those pesky Windows memory problems.
- Borland Delphi - new Windows applications development tool with Client/Server capability - now shipping.
- Competitive upgrades to PowerBuilder 4 Desktop for Clipper & dBASE users - call for details.
- PVCS Version Manager - new release 5.2 now shipping - call for upgrade details.
- Btrieve for Windows NT expected - Btrieve developers can add NT to supported platforms.
- MKS release a new version of Source Integrity - (formerly RCS), now with event triggers.

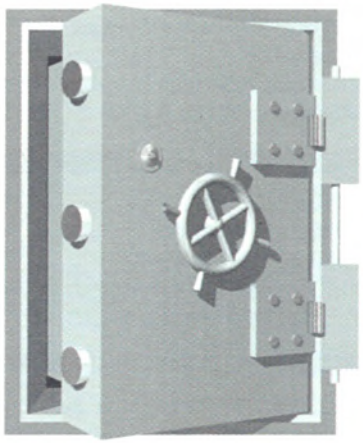
● PLEASE CALL IF THE ITEM YOU ARE LOOKING FOR IS NOT LISTED ● CALL FOR OUR NEW COMPREHENSIVE CATALOGUE ● PRICES ARE EXCLUSIVE OF VAT
● PRICES (EXCEPT UPGRADES & SPECIALS) INCLUDE DELIVERY TO MAILAND UK. SAMEDAY LONDON DELIVERY AT COST ● PRICES ARE SUBJECT TO CHANGE - PLEASE CALL TO CHECK ● VISA, ACCESS, & MASTERCARD ACCEPTED WITH PHONE ORDERS

NEW

Announcing CodeServer™

Discover Client/Server Power with xBASE file compatibility from C, C++ and Basic.

With FREE Client/Server Distribution



With our new CodeServer, your data is safe, your software is reliable and you save $$ with royalty-free distribution.

Completely Scalable Technology

Move from single-user, to multi-user, to Client/Server, all using the same CodeBase API–without code rewrites.

Multi-User xBASE Compatible

You can still access your data directly from dBASE, FoxPro and Clipper.

Improve Reliability and Safety

Using CodeServer only one computer physically changes your data. So even when a client crashes, the database is safe and others can continue working.

Improve Network Performance

Reduced network traffic means improved network performance.

xBASE File Formats

Choose between dBASE, FoxPro, or Clipper file formats for your data.

Logging

Now you'll know who changed what, and when.

Incremental Backups

Make daily backups of all the changes made that day.

Query Optimizer: 1000 times faster

Automatically uses available index information to return query results instantly.

FREE Report Writer Distribution

Now give your users the ability to easily create their own ad-hoc reports by distributing our report writer with your applications, ROYALTY-FREE!

FREE Visual Design Software

For visual, interactive application design, Windows programmers can use CodeControls for formatted data entry and data-aware controls.

CodeBase Multi-Platform

This special multi-user version supports DOS, Windows, NT, OS/2, Macintosh, AIX, HP, SCO, Solaris, SunOS, UnixWare, etc...

dBASE -> C Translator Available

CodeBase™

Database Management for Programmers

Special Offer:

Now for a limited time only, with every order of ***CodeServer***™ *you get:*

- a ***FREE*** CodeBase™ Technology product for the language of your choice
- a ***FREE*** copy of CodeControls™ visual design software,
- a ***FREE*** copy of CodeReporter™ visual report writer software,
- and a ***FREE*** CodeReporter™ end-user distribution license.

Call System Science on 0171 833 1022

Risk-Free, Unconditional 30-Day Money-Back Guarantee on all Sequiter product orders.

3-5 Cynthia Street
London N1 9JF
Fax: 0171 837 6411

SEQUITER SOFTWARE INC.
Fax: +44-81-317-4040

CIRCLE NO. 060

cute(), but a different implement could pass a pointer to a parent data object. Clocks and autosave notifications can be implemented using TMsgTimer.

TOneShotTimer inherits from TMsgTimer. As the name suggests, it disables itself on expiration in addition to sending the message. Time-outs of all sorts can be implemented using TOneShotTimer.

To test out the timer classes discussed, Figures 5-8 are the source files for a test harness called TIME1. From the timer menu, the user can activate any of the three types of timers available, or disable them all. The 10-second timer displays the current Windows time (milliseconds since start-up) in the upper left of the frame window. The other timers pop up a message box. The code was written using the Visual C++ AppWizard. I then removed most of the unwanted doc/view parts before adding the TimerFrame class. I use RegisterWindowMessage to create the timer message numbers, but they could just as easily have been based on WM_USER. The STDAFX.H file has not been listed, as it is the standard version.

A final idea for a TTimer-derived class is an alarm clock, that triggers once at a specific time in the future, or perhaps that triggers at specific times during the days, much like the useful Unix cron utility. Design and implementation are left to the reader.

```
// timefram.h : header file

// CTimeFrame frame

class CTimeFrame : public CFrameWnd
{
  DECLARE_DYNCREATE(CTimeFrame)
public:
  CTimeFrame();

// Attributes
public:

// Operations
public:

// Implementation
protected:
  virtual ~CTimeFrame();

  // Generated message map functions
  //{{AFX_MSG(CTimeFrame)
  afx_msg void OnTimers10second();
  afx_msg void OnTimersMinute();
  afx_msg void OnTimersOneshot();
  afx_msg void OnTimersCancel();
  //}}AFX_MSG
  LONG On10SecTimer( UINT, LONG param );
  LONG OnMinuteTimer( UINT, LONG param );
  LONG OnOneShotTimer( UINT, LONG param );
  DECLARE_MESSAGE_MAP()
};
```

Figure 7 - Header file declaing CTimeFrame class

```
// timefram.cpp : implementation file
//

#include "stdafx.h"
#include "time1.h"
#include "timefram.h"

#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE
                THIS_FILE[] = __FILE__;
#endif

// CTimeFrame

IMPLEMENT_DYNCREATE(CTimeFrame, CFrameWnd)

CTimeFrame::CTimeFrame()
{
}

CTimeFrame::~CTimeFrame()
{
}


BEGIN_MESSAGE_MAP(CTimeFrame, CFrameWnd)
  //{{AFX_MSG_MAP(CTimeFrame)
  ON_COMMAND(ID_TIMERS_10SECOND,
          OnTimers10second)
  ON_COMMAND(ID_TIMERS_MINUTE,
          OnTimersMinute)
  ON_COMMAND(ID_TIMERS_ONESHOT,
          OnTimersOneshot)
  ON_COMMAND(ID_TIMERS_CANCEL,
          OnTimersCancel)
  //}}AFX_MSG_MAP
  ON_REGISTERED_MESSAGE(m_10SecMsg,
                  On10SecTimer )
  ON_REGISTERED_MESSAGE(m_MinuteMsg,
                  OnMinuteTimer )
  ON_REGISTERED_MESSAGE(m_OneShotMsg,
                  OnOneShotTimer )
END_MESSAGE_MAP()

// CTimeFrame message handlers

void CTimeFrame::OnTimers10second()
{
  m_10SecTimer->enable();
}

void CTimeFrame::OnTimersMinute()
{
  m_MinuteTimer->enable();
}

void CTimeFrame::OnTimersOneshot()
{
  m_OneShotTimer->enable();
}

void CTimeFrame::OnTimersCancel()
{
  m_10SecTimer->disable();
  m_MinuteTimer->disable();
  m_OneShotTimer->disable();
}

LONG CTimeFrame::On10SecTimer(UINT,
                              LONG param)
{
  CDC *dc = GetDC();
  char buf[20];
  _ultoa( param, buf, 10 );
  dc->TextOut( 10, 10, buf, strlen(buf) );
  ReleaseDC( dc );
  return 0L;
}

LONG CTimeFrame::OnMinuteTimer(UINT, LONG)
{
  AfxMessageBox( "One-Minute Timer has
triggered", MB_ICONINFORMATION );
  return 0L;
}

LONG CTimeFrame::OnOneShotTimer(UINT,LONG)
{
  AfxMessageBox("One-Shot Timer has trig-
gered",
                MB_ICONINFORMATION );
  return 0L;
}
```

Figure 8 - Implementation of CTimeFrame class

Conclusion

Given the great importance of time to Windows applications, it is unfortunate that proper timer functionality is only (realistically) available under Win32. The TTimer class presented here provides an easy way to manage timers that are portable from Win16 through to Win32 while at the same time hiding most limitations of the underlying implementation. ■

Michael Marshall is a software engineer for Connection Software in Islington, North London, which specialises in communications and financial software. Michael programs in C, C++, and Visual Basic under Unix and Windows. His other interests include sci-fi, mythology and writing. The author can be reached via e-mail as michael@xon.co.uk.

The code discussed in this article is available on diskette. Please send a SAE and diskette to EXE Magazine, 50 Poland Street London W1V 4AX. Mark envelopes 'TTIMER'.

References

Win16/Win32 API Reference
Charles Petzold -
Programming Windows 3.1 - The Book.

MFC Reference
Microsoft KnowledgeBase PSS ID Q74412 This document has much more detailed information on using message-based timers from DLL's, including implementation notes.

Play with fire

Neither should children play with matches, nor inexperienced developers with multiple inheritance. But as **Francis Glassborow** explains, in the hands of those that 'know', both have their rightful place.

When I was discussing this column with your esteemed editor I intended writing two articles on programming methodologies. The first of the two came out substantially as planned and was published in the last issue. When I turned my mind to using C++ for object-oriented programming I realised that much depends on what is meant by OOP. I should warn you that this article is very much based on my own personal views. I think that at the centre of OOP is the concept that objects have a degree of knowledge about themselves.

It's noop, not oop

Inheritance is a key element for providing this 'knowledge' though other things have to be added if the potential is to be realised. It is possible (and perhaps desirable in the right contexts) to use inheritance to tailor a class to meet a specific objective for which it was not originally designed, or to tune it for a specific use. As tools provided by class libraries get better this kind of derivation becomes less often needed. So the following example becomes less likely.

Suppose that you have a container class - a linked-list will do - that includes a sort based on an algorithm that has poor performance characteristics for the data set I want to use. I certainly do not want to rewrite such a class. Even if I have the original code I do not want to breach encapsulation by altering it. Remember that code maintenance is the responsibility of the originator. Derivation from such a class allows me to replace the ineffective sort routine with a better one, so long as efficient access functions have been provided.

THE DREADFUL STORY OF
HARRIET AND THE MATCHES

It almost makes me cry to tell
What foolish Harriet befell.
Mamma and Nurse went out one day
And left her all alone to play:
Now, on the table close at hand,
A box of matches chanc'd to stand;
And kind Mamma and Nurse had told her,
That, if she touch'd them, they should scold her.

But Harriet would not take advice,
She lit a match, it was so nice!
It crackled so, it burn'd so clear, -
Exactly like the pictur here.
She jumped for joy and ran about
And was too pleas'd to put it out.

And see! Oh! what a dreadful thing!
The fire has caught her apron string;
Her apron burns, her arms, her hair;
She burns all over, everywhere.

Before I go any further, I should make clear that persuading the originator to make the data `protected` is not the right way. Data always belongs in the `private` interface. The `protected` interface is for member functions that are safe in the hands of another class designer, but ill-suited to safe use by client code. Now, useful as the technique of tuning a class by derivation may be, it has nothing to do with OOP.

What is OOP?

To me there are two things that make up OOP though they often seem to get confused. The first is the ability to support generic methods which can be specialised in derived classes via virtual functions and dynamic binding. This means that I do not need to know the precise type of an object at compile time. Indeed when writing library code I do not even need to know all, or any, of the subtypes that an object may have. In other words I can handle objects through more primitive views of them, relying on polymorphism to supply correct behaviour.

I think this aspect of OOP programming in C++ is well covered by at least the better books on the subject. The approach is characterised by a (base) class having several derived versions. The classic example is that of an abstract base class `Shape` supporting a polymorphic `draw` function that will dynamically bind to the correct version for a specific shape such as circle or square.

The second ingredient in my variety of OOP is that of being able to take alternative views of an object; being able to take an appropriate view in context. This aspect is often largely ignored, probably because it needs multiple inheritance for it to be useful in any but the simplest of cases.

Changing views

I can remember spending most of a coach journey from Oxford to London discussing what should be the correct design approach to a text-window. There seemed to be four possibilities. The first was designing an entirely new class by aggregation from a window class and a text class such as:

```
class WindowText
{
 Window wndw;
 Text txt;
 // the rest of the data
 // and functionality
};
```

This approach at least has the virtue that it makes no decision as to which view, Window or Text, is primary by taking neither view and making both secondary. Distinct views would then have to be taken via access functions or explicit conversion functions. This approach makes me unhappy because programmers need a very clear understanding of what they are doing if they are to avoid errors from using copies when they intended to use the original. It also breaches the fundamental precept of OO in that you derive for an *is a* relationship and aggregate for a *has a* one. This last objection also applies to the two options via single inheritance:

```
class WindowText :
        public Window
{
Text txt;
 // the rest of the data
 // and functionality
};
```

and

```
class WindowText :
         public Text
{
 Window wndw;
 // the rest of the data
 // and functionality
};
```

In addition both these approaches suffer from taking one or other view as the primary one and relegating the other to secondary status. The reality is that such objects are simultaneously Window objects and Text objects. Only the context can determine which is appropriate. In languages that do not support multiple inheritance you are stuck. You are obliged to choose one of the above. In C++ you have a fourth choice:

```
class WindowText :
        public Text,
        public Window
{
 // other data
 // and functionality
};
```

Surely this approach is conceptually correct? As in all aspects of programming, you have to learn how to use your tools and not simply do something because you can. I think that this is one of the main problems with multiple inheritance. It is used by naive programmers who simply glue two classes together and expect the result to work correctly.

With multiple inheritance it is possible to use the derived object as any one of its bases. This is often the view I want but I must understand this or else forgo the benefits. In the example above, I would want to manage scroll bars, events and so on through a view of the object as a window. Although I may want to provide some special extra tuning to handle text display in the window. On the other hand, I would wish to handle all the text as a text object. This would share such features as position of cursor, methods for writing to file etc with other text objects.

Pretty poly

The full power of C++ to deliver objects with self-knowledge is only available when you learn how to couple multiple inheritance to polymorphism. There are at least two programming methods (**addin** and **mixin**) that capitalise on this combination. As either one of those methods could be the subject of one or more columns I will leave detailed study of them to a later date.

None-the-less I would like to give you a glimpse of what is possible by providing a simple **addin** class that encapsulates a single capability - that of being 'printable'. Consider the effect of the following:

```
class Printable {
public:
 virtual void printon
       (ostream &) =0;
// ensure proper destruction
 virtual ~Printable() {};
};
inline ostream & operator <<
(
 ostream & out,
 const Printable & data
)
{
 data.printon(out);
 return out;
}
```

The class 'Printable' can now be used as a base for any class for which you wish to provide an inserter. Of course, I have to write an actual printon function for the derived class concerned. So I get no great gain except that I have encapsulated a single property. This property can now be used (its inserter will be automatically supported), but more importantly the new **dynamic_cast<>** facility allows me to determine if an object 'is printable'. For example the code in Figure 1

```
class Basic_data
{
  // abstract base class specs
};
class Data : public Basic_data, vir-
tual public Printable
{
// make concrete
};
void main()
{
  Data dt;
  if (dynamic_cast<Printable>(&dt))
       cout << "It's printable";
  else
       cout << "It's not printable";
   cout << endl;
return 0;
}
```

Figure 1 - Is the object printable?

would output 'It's printable'. Please note that virtual inheritance is an essential tool for this method because a heavily derived class might inherit the 'printable' property from more than one ancestor. All the textbook virtual base examples about things such as the number of engines an amphibious vehicle might have are really based on simplistic flawed designs.

Conclusion

Multiple inheritance enables the programmer to encapsulate single properties that can be reused in many classes. Polymorphism allows the implementation of those properties to be tailored to specific needs. The new **dynamic_cast<>** supports objects knowing about their properties. Of course use of such methods by an inexperienced or under trained programmer is like giving a box of matches to a child. Nevertheless, matches can be used constructively by adults and MI is a powerful tool in the hands of fully trained programmers.

If you consider the inheritance lattices that you get for multiple inheritance coupled with polymorphism, you will note a kind of symmetry. Multiple inheritance brings classes into a node while polymorphism and single inheritance branches out from a node. I believe that both these are necessary for full OOP. I am aware that you can manage without, but then you can, if you choose, write all your code in assembler, or if you really insist, you can write it in hex and use a hex keypad (I have actually done both those things in the past but not from choice). That is all for now. Next time I will look at abusing friendship! ■

Mayhem!

Why is a program like a garden? No, it's not a trick question. Jules has an answer.

Every living thing, in my garden and yours, is made of the same basic stuff. They all look different though. And they all work in different ways. What controls these shapes is the genetic information, DNA, contained in every cell. In effect, the DNA is a program to assemble the organism.

Lots of research has been done into what genes are available to each organism and how they relate to each other. But you can learn a lot about genes just by looking around. For one thing, genes have a value. Living things go to extraordinary lengths to preserve their own genes, often at the expense of almost identical genes from individuals of the same species. Since these genes have a value, they must represent something which has a cost. What could that be?

Again, look around. The simplest things, moulds and lichens, look just like lumps of goo. Each cell is like every other; there's no real organisational structure to the organism. We look at such assemblies of cells (which all share

identical genes, by the way) and call them colonies rather than creatures.

But, when the cells start to differentiate and the organism achieves some recognisable structure, we have reached what clearly is a more complex type of organism. We recognise that complexity by calling it a plant. The simplest plants are merely branching stems, but at a higher level of organisation there's a distinction between roots, stems and leaves, as happens in ferns.

You can drill a hole in the leaf or stem of most plants, put rooting compound into it and a root will magically appear. There's not much differentiation between the parts. What's more, even complex plants like trees show a very similar structure. The branches branch; the leaves appear on the very ends. It's only when we get to very complex creatures like fish that we see real differentiation, even though there's still a lot of repetition going on.

Now we know what the cost is. It's differentiation. Most organisms can reasonably assume that they can get sufficient of what they need to survive and grow. A small, simple, DNA program can represent repetition with no difficulty whatsoever. There's no real cost because, as the organism grows it can reach more of the raw materials it needs to build more repetitions. But differentiation carries a big cost. The organism needs more nutrients, and different ones. It needs chemical signals around itself in order to make sure that each cell develops so as to do the job that is required of it and no other. We can, for example, see that practically all animals have a head. Vertebrate animals all share roughly the same bone structure. They all have wrists, elbows, ankles and pelvises, even though many don't use these bones at all. This is because the cost required to create differentiation is very high. The cost of differentiating differently is higher than the cost of adapting what has been inherited. But vertebrate animals have different numbers of vertebrae and different numbers of ribs, showing the cost of repetition is still comparatively low. When you look even closer, the details of nerve and capillary structure are very different from one individual to another, almost random in fact, resembling the structure of trees in their distinctiveness, their branching structure, and the nature of their repetition.

Why am I telling you this? I want to make the case that, in nature, repetition is cheap, differentiation is expensive. I want to suggest that, if it's true in nature, it's probably a natural law of some kind.

We can see the same law at work. Look at modern engineering. Mass-production means that we can make as many of something as we want, so long as all the somethings are the same. Henry Ford cut his costs by making all his cars black; differentiating even the colour would put his costs up. We're developing on-demand ordering systems, but these deal with very superficial levels of differentiation. The core product doesn't change.

Even when making one of something, the same rules apply. A modern suspension bridge, though it has some differentiation, has many parts doing very similar jobs. The suspension cables are all working the same way. The individual wires in the span cables are all working together, and in the same way. No matter what you look at, you can see the same principle at work. Even in programming.

In programming, we get repetition for free. We have well-understood mathematical principles telling us how to make repetition. A large part of programming analysis is finding those repetitions so the languages and the hardware can perform that repetition for us. But, in a real sense, the science and discipline of programming is one of differentiation. It's not the fact of repetition which we seek when designing a program; it's the different repetitions, and their specific interactions. The repetitions themselves are so cheap they're uninteresting.

That's why computers are so powerful. We have, in effect, a factory in a box. We can make as much as we want of any one thing. It costs us no more than making one. The problem is that the differentiation does cost. When we're talking about the difficulty of writing a program, we're talking about the level of differentiation within it. And computers allow an enormous amount of differentiation too.

And here, we're hoist by our own, collective petard. We have no consistent theory of differentiation. As programs get more complex, we're striking out into virgin territory without any analytical tools at all. More and more big projects are being cancelled after they've blown their budgets. Programming is getting a really bad name. To counter this, the engineers are asking that programming become less of an art or a cottage industry, and more like an engineering discipline. That won't help, because engineering doesn't have a sufficient concept of differentiation to help us. If it did, we wouldn't have needed to invent computers in the first place.

It's a mistake, I think, to look at the history of engineering and suggest that computer science should proceed in the same way. Instead, we should turn to natural history and ask what lessons we can learn from there. Engineering used to be one-off, not clearly differentiated constructions. Then we invented mass-production, which corresponded to repetition. Then we invented on-demand, which corresponded to differentiation. Now we've progressed beyond that and we're complaining because differentiation is expensive and fraught with difficulty.

If we're going to solve this problem, we can't do it by regurgitating the old answers. We'll certainly need lots of ideas and lots of theory. We'll probably need lots of computers too. ■

Despite appearances, Jules actually hates gardening, and thinks a wild meadow looks healthier than the best-maintained flowerbed. None of his neighbours agree, but on the other hand, they're not prepared to dig his garden for him. He can be contacted on 01707 644185, or on cix as jules.

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

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

Right Sed Fred

SED is the utility **Peter Collinson** has used for the past three years to put his articles in a format suitable for EXE magazine. Ever wondered how he does it?

Unix is stuffed with tools which help us to transform data. The stream editor, `sed` is one such tool that contributes to the power of the system. We can use it to scan logs selectively, to make repetitive changes to files, or as part of a shell script transforming text between other tools. It's good as a programming aid as well. After all, programming is a text handling function. Personally, I will use `sed` if I am changing a routine name in a large program composed of several modules. I can create a little shell script to get things right and then apply it to every file that comprises the program.

Enough SED

I also use `sed` for scanning text files like my address database. I spent a happy 15 minutes the other day making 'Phone Day' changes. Some numbers are changing radically so this is slightly non-trivial, as you probably know. It's this kind of external event that argues strongly for storing all data in text files so that we always have the ability to edit the file using normal means when the unexpected happens.

The `sed` stream editor springs from the Unix filter concept. Rather than reading data into an editor and using commands to change it, we read *commands* into the editor and apply them to a passing data stream. We can load commands from a file or enter them from the command line.

Y'know Ed?

Commands are based on `ed` commands. and have the general format of:

```
<address><function>
```

or

```
<address>,<address><function>
```

or simply,

```
<function>
```

Some functions are followed by optional arguments. The default operation of `sed` is to read data into an internal buffer called the *pattern* space. It then applies all the commands whose addresses match this pattern space. These are applied in order until the end of the script is reached. Commands with no addresses are always applied. Finally, the pattern space is written out to standard output and the process starts again.

In normal use, the pattern space is always output even if no transformations have been made on it. We can use this to make a `head` command:

```
% sed 10q infile
```

The command here is the address `10` and the function `q`, This will exit from `sed` after line 10 has been processed. Lines 1 to 10 will be printed.

Normal output is suppressed if we give `sed` the `-n` argument, so:

```
% sed -n 10p infile
```

uses `p`, the print command, to output line 10. And

```
% sed -n 1,10p infile
```

will print lines 1 to 10. This will be slower than the script that uses `q` because the remainder of the file is processed, looking fruitlessly for lines that match the address.

Strictly, a `sed` command on the shell input line should be preceded by a `-e` flag, but this is waived if there is only one command. It's useful if we are doing something a little more complex:

```
% sed -e 1,10d -e 20q infile
```

prints lines 11 to 20 of the file. The other alternative is to place the commands one to a line in a script file and say:

```
% sed -f script infile
```

where `script` contains the commands that we want to execute. I don't like to do this if my `sed` command is part of shell script because I like shell scripts to be self contained. We can supply a multi-line input to the `-e` flag:

```
#!/bin/sh
sed -e '1,10d
    20q' infile
```

The above code works fine in Bourne and Korn shells because they deal correctly with

Regular expressions are really WORN: Write Once, Read Never

Nu-Mega
TECHNOLOGIES INC

Give Your Bugs The Bird!

Get The Combined Power of A Heap Checker, Spy Utility, Debug Kernel, API Debugger And GPF Tool, All In A Single Package.

Need the perfect bug finding tool? Nu-Mega has it! — in Windows, DOS, WindowsNT, WIN32s even Windows 95. It's called BOUNDS-CHECKER™. Many say it's the most versatile and powerful automatic bug finding tool money can buy. Our customers love it because it saves them huge amounts of time and does "clean-up" better than anything out there.

Where Do Bugs Come from?

Today's bugs are just as likely to come from improperly interfacing with the operating system or third party components as with your programming language. That's why we've expanded BOUNDS-CHECKER into a single comprehensive tool that points it's long bug hunting finger at the types of problems you are most likely to encounter.

Stop Problems like:

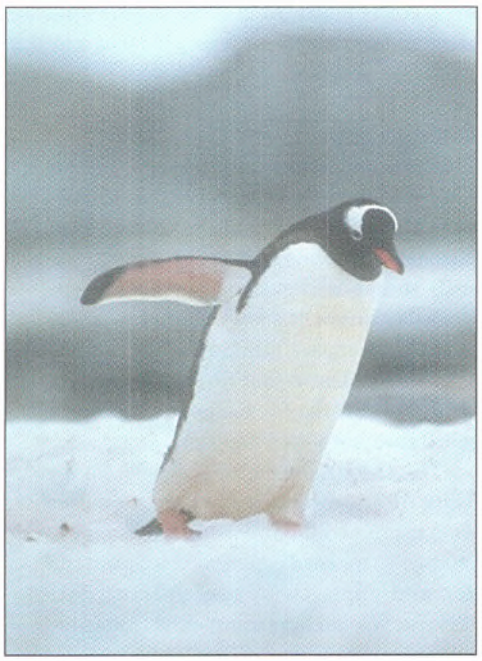
■ API Parameter Errors ■ Memory And Resource Leakage ■ Data And Heap Corruption ■ API Compliance Errors ■ Processor Faults ■ C/C++ Pointer Errors ■ API Return Value Errors ■ VBX Interface errors and more!

Find Out Why They Happen

Your work is not always done when the bug is found. In fact, 50% of the job is finding out what caused the bug. There's great news though — BOUNDS-CHECKER works as your partner in helping you diagnose the problem. Along with giving you the description of the bug, BOUNDS-CHECKER reveals the source line that most likely caused the bug! For more difficult problems, you can access a variety of windows and inspectors that show you the state-of-the-program at the instant the bug was found! And for *"killer bugs"*, BOUNDS-CHECKER shows you the history of events that led up to the problem! What's more, with BOUNDS-CHECKER, you can go back in time and observe the series of Windows messages, APIs and other events that preceded the bug. This is the power you've been waiting for — now turn week-long debug sessions into a few minutes work!

New In BOUNDS-CHECKER!

VBX and Windows 95 support— BOUNDS-CHECKER now validates the VBX custom control interface and logs VBX events as part of its trace history. Whether your program is written in VB, C/C++ or any other language, you can get a handle on some of the weirdest bugs you are likely to hit.



TView™ - Cool Event Viewer!

As your program runs, BOUNDS-CHECKER transparently logs messages, API calls, hooks, callbacks, VBX calls and many other events in a trace file. TView, BOUNDS-CHECKER's event viewer graphically organizes and filters events so that you can see a high-level view of your program's run-time activity. When you want to see more detail, just click on events of interest to expand, level-by-level! You can also view API parameters and even bring up the source line associated with an event. Besides helping you see programs from the inside,(even on executables with no debug information), TView is an excellent tool for tracking down some of the toughest Windows problems.

The Perfect Q/A Tool

If your job includes making sure your company's programs are as solid as possible, you need the power of BOUNDS-CHECKER. Many of the bugs that BOUNDS-CHECKER finds will be invisible during test cases, but this tool has eyes! You cannot possibly test every permutation on every conceivable system configuration —but, by using BOUNDS-CHECKER, you can flush-out the hidden bugs and greatly improve your quality.

Don't Doubt It, Do It!

Next time, show your bugs who's boss...**Give'em the bird!**

Order A Trio And Save A Bundle!

Choose any two BOUNDS-CHECKER products and get the third FREE!

BOUNDS-CHECKER **£195+ VAT each!**

❑ For Windows ❑ For 32/NT
❑ For WIN32s ❑ For DOS
❑ Windows 95 Pre-Release Program
(Call for ordering information)

Special Trio Price: £415 + VAT
Call System Science Now! (0171) 833 1022
Or Fax This Form! (0171) 837 6411

❑ VISA ❑ MasterCard

Card # ___ ___ ___ ___ Exp. ___

Card Holder Name (Print) _____

Signature _____

Phone: _____

Call (0171) 833 1022
Fax (0171) 837 6411

SYSTEM SCIENCE

RISK = NULL
30 DAY
MONEY BACK GUARANTEE

3-5 Cynthia Street
London N1 9JF

24 HOUR BBS
001-603-595-0386

newlines inside quoted strings. Actually I often write something like:

```
#!/bin/sh
sedprog='1,10d
    20q'
sed -e "$sedprog" infile
```

because this divorces the specification of the program from its use and makes the shell script more readable.

In **csh**, we need to add an extra backslash to get the newline into the command as shown below.

```
#!/bin/csh
sed -e '1,10d\
    20q' infile
```

which is one of the reasons that I always caution people against using **csh** for writing scripts. Also, as we shall see, **sed** uses backslash internally. These tend to multiply as we force them through successive layers of interpretation.

A grep by any other name

Of course, line numbers are of limited use in addresses. What we need is the ability to apply commands to a file depending on its contents. We use the regular expression syntax here.

```
$ sed -n '/look/p' infile
```

This is a replacement for **grep**, printing out the lines in the file that contain the string **look**. I tend to quote arguments to **sed** out of habit, as it stops the shell messing with any regular expression meta-characters that I may use. We can also do **grep -v**, printing the lines that *don't* contain **look**:

```
% sed '/look/d' infile
```

Notice that **sed** deals with single lines from the original file. An address range is a little like a **for** loop allowing you to extend the command to be actioned on several lines.

```
% sed '1,/^$/d' mailfile
```

The above example deletes all the lines from line one to a blank line. We are removing the header from a mail message. The regular expression here is a common idiom for an empty line. It's a caret meaning the start of the line, followed immediately by dollar meaning the end of the line. So this matches an empty line. A blank line terminates the header in the mail message.

Search and replace

Perhaps the most used **sed** function is **s** replacing some old text by some new.

```
s/old/new/
```

The first occurrence of **old** in the pattern space is replaced by **new**. To ensure that all the occurrences are replaced, we must add the **g** flag to the end of the command.

```
s/old/new/g
```

We can also tack a **p** at the end of the command to make **sed** print the line that it has constructed.

An empty right-hand side is used to delete the string, so

```
s/old//g
```

will remove all occurrences of **old** on the input line.

Of course, the left-hand side of the pair of strings is a regular expression that is used to match the input line. We have to be a little circumspect using this because sometimes it will match strings unexpectedly.

We often need to match characteristics of the text rather than the actual characters. We have come across this already. Caret matches the start of a line and dollar matches the end. Other useful meta-characters are '**.**' (dot) matching any character; '*****' (star) matching zero or any number of the preceding character; '**\<**' and '**\>**' matching the start and end of words and '**|**' providing alternations. So:

```
s/\<unix\>|\<Unix\>/UNIX/
```

matches complete words that are **unix** or **Unix** replacing them by the capitalised form. Backslashes are usually used to force a meta-character itself to be matched in the text. However, the start and end of word meta-characters were invented after **sed** had been in operational use for some years and the reverse of the usual notation was used to permit backwards compatibility.

We can match single characters from a range by placing the options in square brackets. So

```
s/[Uu][Nn][Ii][Xx]/UNIX/
```

will match all upper and lower case variants of UNIX in the source text. It's sometimes useful to be able to match characters that are not in a particular range. This is done by starting the square bracketed list with a caret, so

```
s/[a-zA-Z][^0-9]*//
```

will match text starting with alphabetic characters and followed by any number of characters that are not numeric.

There are a couple of special actions that involve the right hand side of the string. First, we can insert the text that was matched by using the ampersand character:

```
s/[Uu][Nn][Ii][Xx]/& system/
```

This will pick up all the variants of 'UNIX' in the source and replace it by the original text followed by the word **system**. Second, regular expressions can be grouped and identified by using **\(** and **\)**. The replacement text can then refer to the matched text. For example, let's say we want to swap the columns output by something that generates two columns, say the **du** command:

```
s/^\([0-9][0-9]*\) *\(.*\)$/\2 \1/
```

The left hand side first matches the start of the line **^**, then we have a group which is a numeric character, followed by zero or more numeric characters **[0-9][0-9]***. This is placed in a group **\([0-9][0-9]*\)**. This group is addressed on the right-hand side by **\1**. There is then the gap between the columns which I am assuming is a bunch of spaces, matched by a space followed by star. We might need to worry about tab characters too.

Finally the remainder of the line is matched. The **.*** construction is another common idiom that matches anything, this is grouped too **\(.*\)**. The group is referred to by **\2**. The last character, **$**, makes the *anything* match the whole of the rest of the line. The right-hand side inserts the matched text from column 2, a space and the text from column 1.

Pattern spaces

There are often occasions where we want to find some text and then use several commands on it. Command grouping is done using curly braces. All the commands inside the braces are applied in order to the pattern space. This becomes useful when we want to get **sed** to deal with data that spans more than one line, which turns out not to be hard.

To add the next line to the pattern space, we can use the **N** command.

```
/delete/{
  N
  d
}
```

The script deletes any line containing the word **delete** and the line that follows it. It looks for a line containing the word **delete** and applies the group of commands to it. The **N** command reads the next line from the input and appends it to the pattern space. The entire pattern space is deleted by the **d** command.

Joining lines can also be done with N:

```
/join/{
  N
  s/\n/ /


}
```

finds a line that contains the string `join` and pulls in the line after it. The substitute command then changes the newline character shown by `\n` into a space.

Sadly the `\n` construction only works on the left hand side of the substitute command. If we want to replace a character by a newline, then we must really use one and add an extra backslash to tell `sed` that it's not a mistake:

```
/-$/{
  N
  s/-\n\([^ ]*\) /\1\
/
}
```

This code joins words that have been hyphenated over a line break. It looks for lines ending in a hyphen (`-$`) and pulls in the next line using the `N` command. The substitute command searches in the pattern space for a hyphen followed by a newline - `\n`, followed by any number of characters that are not spaces `[^ ]*`. This is placed in a group `\([^ ]*\)`. Finally, we add a space to the regular expression, to deal with the space after the first word on the next line. The expression matches the italics (and the following space) in this example:

```
a <f>hyphen-
ated <t>line
```

We replace the hyphen and the newline by the text at the start of the next line picked by the group `\1`. This will be ated. The result is:

```
a hyphenated
line
```

Twisted hypens

The script appears to work. However, what happens if the second line also ends in a hyphen? This line will be pulled into the pattern space using the `N` command and will be output at the end of the command sequence. It will never be treated as a 'first line' and the hyphen will not be removed.

There's a couple of commands that help to deal with multiple lines in the pattern space that will solve this problem. The `P` function prints the contents of the pattern space *up to and including* the first newline. The `D` command will delete the data in the pattern space up to and including the first newline. `D` will start the next cycle with the pattern space already loaded. The full script that we need is:

```
/-$/{
  N
  s/-\n\([^ ]*\) /\1\
/
  P
  D
}
```

Let's step slowly through this slowly, using:

```
A hyphen-
ated line long-
er than before
```

The first line of the script will read line 1 into the pattern space and the `N` command will pull in the next line. The pattern space contains:

```
A hyphen-
ated line long-
```

We now do the substitute command that moves bits of the two lines around:

```
A hyphenated
line long-
```

The `P` command prints the first of these lines. The `D` command deletes it. The pattern space is now:

```
line long-
```

Since the pattern space is not empty, the script will be restarted without reading new lines from the file. This line is matched by the first regular expression and the `N` command will pull in the next line:

```
line long-
er than before
```

The substitute command does its bit:

```
line longer
than before
```

the `P` command prints the first of the two lines. The `D` command deletes it leaving the pattern space holding the last line. This does not match the first regular expression and will be printed. The pattern space is emptied and we will finish.

Collapsing lines

Here's another useful example. It collapses repeated empty lines into a single one:

```
/^$/{
  N
  /^\n$/D
}
```

The script looks for an empty line `^$`. It pulls in the next line using the `N` command and then deletes the first line of the pattern space if the pattern space contains a single character that is a newline.

In addition to the pattern space, `sed` allows us to use a scratchpad area called the *hold* space. This retains data until it is reloaded. It's a little like the memory on simple calculators. The `h` command loads the hold space from the pattern space and the `g` command reverses that, loading the pattern space from the hold space. We can exchange the two with the `x` command. There are also capitalised `H` and `G` commands that allow us to deal with multiline input. The `H` command appends the pattern space to the hold space and the `G` command appends the hold space to the pattern space.

SED it was useful

Let's use this to make a little script that scans our C program source for function definitions and makes the prototypes that we need to place in a header file. I declare functions on two lines:

```
char *
strcpy(char *to, char *from)
```

but I like to have function prototypes on a single line:

```
char *strcpy(char *, char *);
```

I also remove the names of variables in the prototype.

How do we do this? The first thing to notice about the problem is that function declaration lines are unique in a program. They are the only objects in the file where a C identifier appears at the start of the line *and* is followed by a round bracket. It's possible, then, to write a regular expression that identifies the declaration. Having found the line, we want to glue this to the previous one to ensure that the declaration is on a single line. The easiest way to do this is to retain all the 'last' lines in the hold space.

The script is designed to be run with `sed's -n` flag so we only output text with a `p` command.

```
/^[a-zA-Z][a-zA-Z0-9_]*(.*)/{
  H
  g
  s/\n/ /
```

```
    s/$/;/
    s/[a-zA-Z][a-zA-Z0-9_]*\([,)]\)/\1/g
    p
}
h
```

The first line is the regular expression that picks out the function definition. At the start of the line (^) we want to find a C identifier. A C identifier is an alphabetic character [a-zA-Z], followed by zero or more alphanumerics and possibly an underscore [a-zA-Z0-9_]*. The C identifier is followed by a pair of round brackets that enclose *anything* (.*) so that we can pick out the function declaration. Before I proceed, notice the last line. The h command loads all lines into the hold space, so when we find a function, the hold space will contain the previous line.

Back on the second line, we append the pattern space to the hold space using H. This creates a multiple line entity that is the function type declaration from the previous line and the function definition.

Now we use g to move the completed line to the pattern space. The string has an internal newline, so we delete it. The pattern space now holds:

```
char *strcpy(char *to, char *from)
```

We can then tidy up the line. We add a semicolon to the end of the line because we need one. We now want to remove any embedded identifiers. These are always a C identifier followed by a comma or a closing bracket. So, we look in the line for this, and replace the combination by the last character that is picked out by the group. Finally, we print the line.

Last words

I know that regular expressions are really WORN: Write Once, Read Never. They are much easier to write than read. I hope that by breaking them down carefully, you have followed them in the examples above. Of course, you can throw up our hands in horror and say: 'that's not for me, I'll keep on using my editor'.

However, there are two big benefits of using sed for these repetitive editing jobs. When we have a script, then that's the job done once and for all time. Next time, it's a matter of a few seconds to apply the edits. I have a sed script that I wrote in 1992 that transforms these articles from my document preparation system into the format expected by EXE.

Second, we know that a script is exhaustive, we know that all the objects which we wanted to change have been edited. We can be confident that we have not missed one. Of course, we need to be confident that the sed script has not been over-enthusiastic and changed things that it shouldn't. The diff command can help here.

As to multiplatform support, MKS provide an excellent version of sed that runs on DOS and OS/2 in its standard set of POSIX tools. There is also a GNU sed that is undoubtedly ported to several platforms.

Reading

Don't think that I have covered sed in great detail in this article, I haven't. There is a lot more. Many basic Unix books have sections on sed. The best book that I have come across is a joint book on sed and awk in the O'Reilly & Associates Nutshell series. The book is provocatively entitled *sed & awk*. It's by Dale Dougherty and sports an ISBN of 0-937175-59-5. Worth the money.

I have also dredged some of this from an excellent book in the Nutshell series: *Unix Power Tools*, by Jerry Peek, Tim O'Reilly and Mike Loukides (and more, in reality). It's published by O'Reilly Associates/Bantam and is ISBN 0-553-34502-7. It comes with a CD-ROM of all the programs and tricks that it mentions. ■

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as pc@hillside.co.uk, by phone on 0227 761824 or on WWW as http://www.hillside.co.uk/

Easy Peasy, when it's VXDeasy

To harness the true power of Windows you need to dive into the underworld of virtual device drivers. But the route is fraught with difficulties unimaginable in ordinary Windows programming. **Dave Jewell** thinks he has found the answer...

Although many Windows developers need only ever use the API calls, it's undeniably true that from an architectural view, the 'real' Windows API is the one that exists at the VxD level. The original design concept behind VxDs (or virtual device drivers) is that they would act as an arbitration layer around a real, physical device.

You'll recall from my comments a month or so ago that Windows really *is* a fully pre-emptive, multi-tasking operating system. You simply have to remember that all the different VMs (virtual machines) correspond to the 'programs' in this operating system. Thus, all your DOS boxes will be preemptively scheduled with respect to one another and with respect to the system VM. The VM is a special virtual machine that contains all the currently running Windows tasks.

Because of this structure, it was necessary to come up with some system component that would control access to a shared, physical device. Here you have the original raison d'être behind VxDs. Not only do they virtualise device access for your printer, keyboard mouse and other 'obvious' devices, but they also do the same for your PC's DMA chip, interrupt controller and other low-level services.

That's fine as far as it goes, but the original design concept is no longer adequate. The VxD mechanism has been heavily used within Microsoft to replace virtually (no pun intended) the core of DOS, thus providing 32-bit file access, an installable file system manager and more. Under Windows 95, many of the new Plug and Play facilities are accessible only at the VxD level. Microsoft has itself used VxDs as part of the new long file name implementation.

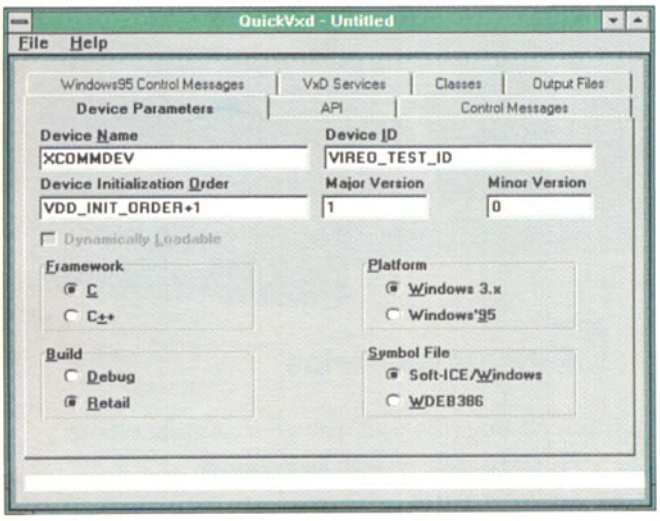


Figure 1 - The Device Parameter page of QuickVxD

Decent third-party books on the subject are about as common as the proverbial rocking-horse manure

The end result is an increasing awareness amongst Windows developers that they need to understand and develop their own VxDs. Only through VxDs can you get Windows to do everything you want. In his excellent article, (*InfoWorld, February 27th 1995*) Andrew Schulman has rightly referred to VxDs as 'TSR's of the 1990s'. Sooner or later, you'll be faced with the need to implement a VxD.

Breaking the VxD barrier

Traditionally, VxD development has been something of an obstacle course. The Microsoft DDK documentation is nothing if not obscure. All development work had to be done using unfamiliar 32-bit assembler code. Decent third-party books on the subject are about as common as the proverbial rocking-horse manure. Vireo has rightly guessed that there's going to be a big surge of interest in VxD development. As a consequence it has released VToolsD, a library of C and C++ routines that enable you to create a VxD with little or no assembler programming required.

A Windows 3.1 version of the library is shipping. You can also get hold of a pre-release Windows 95 version which will be finalised once the long awaited operating system finally appears. Figure 4 is a screen-shot from the Windows 95 version of the

The VToolsD library imposes a structure onto your VxDs that's rather like the 'application frameworks' (MFC, OWL, etc) used at a higher level. In fact, Vireo provides its own 'wizard', a program called QuickVxD, which can be used to fill in details on the operation of your new VxD. It then generates the necessary skeletal code for you (in either C++ or C as required) and the necessary include file and make file. Figure 1 shows the Device Parameter page in QuickVxD. Notice that VToolsD supports the new dynamically loadable VxD mechanism that's part of Windows 95. This option is grayed-out if Windows 3.1 is selected as the target platform.

Old friends

Vireo claims that the VToolsD libraries provide C/C++ interfaces for all the standard Microsoft VxDs and for the VMM services themselves which comprise the core VxD 'API'. A particularly welcome feature of the libraries is the inclusion of a large subset of the ANSI C runtime library routines. All your old friends such as `malloc`, `free`, `realloc`, `sprintf`, etc are available, making VxD programming very much more convenient and familiar. Some parts of the library are written in assembler code for maximum performance. And of course, you can easily add your own assembler fragments where necessary using the Microsoft C compiler's integrated in-line assembler.

The library has been written in a very granular manner, meaning that each routine occupies its own object module. As a result, the amount of linked in, extraneous code is kept to an absolute minimum, making VxDs built with VToolsD as small as possible. Full source code to the library (C, C++ and assembler) is provided which can be easily modified to suit your own requirements. You can also build both debug and retail versions of the C++ and C libraries using the make scripts supplied. Online help documentation is provided for the class library itself and for the QuickVxD program. Source code for numerous sample VxDs is included to get you started. In addition, the introductory chapters of the VToolsD *User's Guide* contain one of the clearest descriptions of VxDs that I've ever seen.

As simple as...

By far the simplest way of putting together a VxD is to use the QuickVxD program. This is organised as a tabbed dialog in which you can fill in the various fields of information that describe your VxD. The most important of these are the `Device Name` and `Device ID` fields. Vireo recommends that in order to avoid any future conflict, you should use a `Device Name` field which does *not* follow the Microsoft convention of beginning with a 'V' and ending with a 'D' character. Additionally, the `Device ID` fields is intended to be unique across all virtual devices. You're supposed to contact Microsoft in order to obtain a unique ID of your own. For the purposes of development, Vireo supplies a special ID, `VIREO_TEST_ID`, which you can use to get things up and running. The QuickVxD program also lets you specify whether you want to generate code for the C++ framework, or whether you want to stick with plain vanilla C. Experience suggests that a driver built using the C++ framework will be somewhat larger than its straight C equivalent. But using the C++ classes have the advantages of simplifying and abstracting the VMM/VxD API routines, just as MFC (for example) provides abstraction at the Windows API level.

The concept should be very familiar to any user of the message-cracking macros in WINDOWSX.H.

The QuickVxD utility will also allow you to set up version information, V86 mode and Protected mode entry points, Device Initialisation Order (useful if you want to ensure that some other device is initialised before your own when Windows starts up), decide between a debug or retail build and more.

Vireo particularly recommends using Nu-Mega Technologies' Soft-Ice/W debugger for debugging VxDs. Aside from Microsoft's arcane WDEB386 utility, it's the only Windows debugger which gets sufficiently close to the hardware to allow debugging at this level. There's another QuickVxD option which specifies whether you will be debugging with WDEB386 or Soft-Ice/W. If the latter is selected, the generated make file

```
// SIMPLE.C - main module for VxD SIMPLE

#define   DEVICE_MAIN
#include  "simple.h"
#undef    DEVICE_MAIN

Declare_Virtual_Device(SIMPLE)

DefineControlHandler(DEVICE_INIT,
                     OnDeviceInit);

BOOL ControlDispatcher(
  DWORD dwControlMessage,
  DWORD EBX,
  DWORD EDX,
  DWORD ESI,
  DWORD EDI)
{
  START_CONTROL_DISPATCH

    ON_DEVICE_INIT(OnDeviceInit);

  END_CONTROL_DISPATCH

  return TRUE;
}

BOOL OnDeviceInit(VMHANDLE hVM,
                  PCHAR CommandTail)
{
  return TRUE;
}

////////////////////////////////////////
///
// SIMPLE.H - include file for VxD SIM-
PLE

#include <vtoolsc.h>

#define SIMPLE_Major          1
#define SIMPLE_Minor          0
#define SIMPLE_DeviceID \
                  UNDEFINED_DEVICE_ID
#define SIMPLE_Init_Order \
                  UNDEFINED_INIT_ORDER
```

Figure 2 - Source code for the simplest possible 'no-op' VxD, generated using the QuickVxD utility

```
// SIMPLE.cpp - main module for VxD SIMPLE

#define DEVICE_MAIN
#include "simple.h"
Declare_Virtual_Device(SIMPLE)
#undef DEVICE_MAIN

SimpleVM::SimpleVM(VMHANDLE hVM) : VVirtualMachine(hVM) {}

SimpleThread::SimpleThread(THREADHANDLE hThread) :
                                    VThread(hThread) {}

BOOL SimpleDevice::OnDeviceInit(VMHANDLE hSysVM,
                                PCHAR pszCmdTail)
{
    return TRUE;
}
```

Figure 3 - The equivalent C++ class library for 'no-op' which improves the abstraction of the VMM / VxD API and reduces the amount of code that you need to write

get connected...

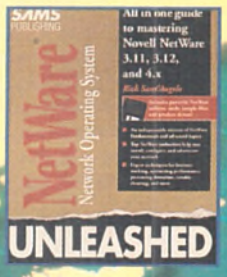
SAMS PUBLISHING

World Wide Web Unleashed
John December & Neil Randall
1058 pp 0-672-30617-4
pbk £26.95 inc VAT

Teach Yourself Internet: Around the World in 21 days
Neil Randall
676 pp 0-672-30519-4
pbk £19.50 inc VAT

GUARANTEED TIMELY & ACCURATE

Internet Unleashed
Foreword by Kevin Kelly
Exec Ed, WIRED Magazine
1387 pp 0-672-30466-X
pbk/disk £31.50 inc VAT



NetWare Unleashed
Rick Sant'Angelo
1258 pp 0-672-30209-8
pbk/disk £41.95 inc VAT

UNIX Unleashed
Written by UNIX Wizards
1620 pp 0-672-30402-3
pbk/CD Rom £39.51 inc VAT

Teach Yourself More Visual Basic 3 in 21 Days
Paul J. Perry
591 pp 0-672-30495-3
pbk/disk £27.50 inc VAT

Teach Yourself Visual Basic for Applications in 21 Days
Matthew Harris
909 pp 0-672-30447-3
pbk £23.00 inc VAT

Teach Yourself Access 2 in 14 Days
Paul Cassel
668 pp 0-672-30488-0
pbk £19.50 inc VAT

Develop Your First Paradox 5 for Windows Application in 14 Days
Lee Atkinson
484 pp 0-672-30597-6
pbk/CD Rom £27.50 inc VAT

Sam Lane
Prentice Hall,
Campus 400,
Maylands Avenue,
Hemel Hempstead,
Herts HP2 7EZ.
Tel: 0442 882163
Fax: 0442 257115

Handling & Carriage Charges
For delivery in the UK please add £3.00 per order (up to 5 books), plus £1.02 for each additional book

Prices are correct at the time of going to press but are subject to change. Orders are supplied at prices on receipt and are subject to our Terms and Conditions of Sale – a copy of which is available upon request.

Please charge my **Access/American Express/Visa/ Diners Club/Mastercard/Switch** card with the sum of: £

Account No: ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Valid From: ☐☐☐ Expiry Date: ☐☐☐

Credit Card customers: please ensure that the address stated below corresponds with the address at which the card is registered.

I enclose a cheque for: £

Payment to be made to: INTERNATIONAL BOOK DISTRIBUTORS LTD

Signature: _____

Name: _____

Organisation: _____

Address: _____

_____

Postcode: _____ Country: _____

automatically calls Nu-Mega's MSYM program to create a symbol file for input to Soft-Ice/W.

Once you've finished entering the specification for your VxD, you can save it as a .QVD file and reload it under QuickVxD at a later date. When you press the `Generate Now` button, QuickVxD verifies the input parameters and generates a skeletal C/C++ source file along with make and include file. You can then add your own code to the source file that's been generated and use Microsoft's NMAKE utility to build the actual VxD itself. The Vireo documentation includes information on how to tweak the generated make file and how to alter the USER.MAK file after the event. USER.MAK is a configuration file created during setup which, amongst other things, stores the location of the different development tools that are used during the build process.

All for NOP

The VxD shown in Figure 2 is essentially the VxD equivalent of a NOP instruction! Nevertheless, it's useful as a means of illustrating the structure of a VxD from a C programmer's point of view. The DEVICE_MAIN symbol must be defined in the main module of your VxD. This is followed by the inclusion of the SIMPLE.H header file which is generated by the QuickVxD utility according to the information that you specify. As you can see, no device ID has been defined and the device initialisation order is considered unimportant.

The `Declare_Virtual_Device` macro is responsible for building the device descriptor block (DDB), a structure that every VxD must have. This is very loosely analogous to the header used by MS-DOS device drivers. It contains important information about the device. The name of the device (as specified in QuickVxD) is passed as a parameter to this macro since one of the fields of the header contains the device name.

The next macro, `DefineControlHandler`, declares a handler, `OnDeviceInit`, for the control message `DEVICE_INIT`. A call to the `DefineControlHandler` macro is made for each control call specified in the QuickVxD utility. In addition, QuickVxD generates a skeletal control message handler with appropriate parameters for each call. You can see the skeletal `OnDeviceInit` call at the end of Figure 2.

Many of the new Plug and Play facilities are accessible only at the VxD level

The ControlDispatcher routine is called by the C framework library whenever a control message is received by the VxD. As you can see from the function declaration, the message itself is available for inspection, along with the EBX, EDX, ESI and EDI machine registers. This is the VxD equivalent of a 'window procedure'! The START_CONTROL_DISPATCH and ENC_CONTROL_DISPATCH simply start and terminate a switch statement. The intervening macro calls are all of the form ON_XXXX. These ensure that the appropriate number and type of arguments are passed to each control message handler. The concept should be very familiar to any user of the message-cracking macros in WINDOWSX.H.

Finally, the listing finishes with the skeletal OnDeviceInit code. By default, this returns TRUE, indicating to Windows that the driver was successfully initialised.

Figure 3 is the equivalent code, written to use the C++ class framework. As you can see, three different classes are involved here. The specific class SimpleDevice is derived from a base class called VDevice. A single instance of the SimpleDevice class is created when the VxD is loaded by Windows. The SimpleVM class is derived from a base class called VVirtualMachine. An instance of this class corresponds to a specific virtual machine in the VMM/VxD environment. Some control messages contain information that relates to a specific virtual machine. These messages are typically handled by member functions of classes derived from the VVirtualMachine class. Finally, the SimpleThread class is derived from VThread, a base class that's there to support the processing of thread-oriented control messages. To summarise then, thread-oriented messages are processed by control message handlers derived from VThread VM-oriented messages are processed by handlers derived from VVirtualMachine and common or garden control messages are processed by handlers derived from the VDevice class, an example of which is the SimpleDevice::OnDeviceInit member function in Figure 3. Easy...

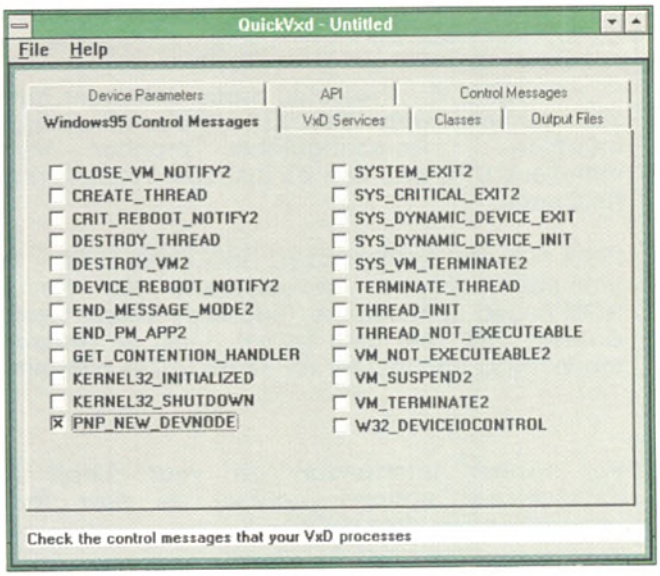


Figure 4 - The final Win95 version of VToolsD will be finalised once the operating system itself is available

System requirements

Installation of VToolsD is relatively straightforward, using the ubiquitous Microsoft installer program. A full installation requires around 9 MB of hard disk space. VToolsD is currently compatible with both Visual C++ 2.0 and with the 32-bit version of Visual C++ 1.0 (also known as C8). For compatibility with other 32-bit C/C++ compilers, you should call Vireo.

For obvious reasons, you do need a 32-bit compiler to create a VxD. I installed VToolsD under Windows/NT 3.5 with no problems although the install program did complain that the Visual C++ 2.0 linker was an unknown version. I suspect that this was due to my having previously 'bound' the linker (using Phar Lap's TNT DOS Extender), so that I could use it under MS-DOS.

How about it?

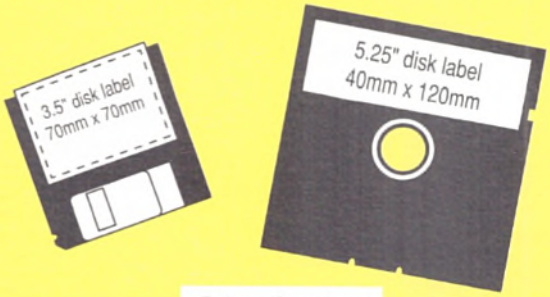
VToolsD is an elegant and very easy to use C/C++ library for VxD development. The documentation is excellent, full source code is included, there's on-line help and you get numerous examples to get you going. It really couldn't be made much easier than this, could it ? ■

System Science has just been appointed Vireo distributor in the UK. No details of UK pricing and availability are available at time of writing (it's Sunday night !) but you can get the low-down by phoning System Science on (0171) 833 1022, or fax them on (0171) 837 6411.

Dave Jewell is a freelance writer who's VxDs got in a twist so he wrote 'Instant Delphi' by Rox Press. Dave can be contacted on email as djewell@cix.compulink.co.uk.

DISK LABEL PRINTING

Customise your labels by adding your company logo or colour



Other Services

MANUALS FULL ARTWORK/ORIGINATION
WIRE-O-BINDING TABBING SHRINK WRAPPING
DATA CONVERSION OVERPRINTED DISK SLEEVES
RING BINDERS/SLIPCASES PLAIN OR PRINTED

Qty	Price one colour	Per extra colour
1000	£85	£36
per 1000 run-on	£22	£22

Supplied on A4 sheets - Permanent adhesive
Inclusive of delivery - VAT excluded

Ridgeway Press, 13 Horseshoe Park, Pangbourne, Berks RG8 7JW
Tel: 0734 845331 Fax: 0734 845186

CIRCLE NO. 065

LicenseServ™

Price Buster

The Most Powerful License Manager

Looking to secure your software?
Feel fed up with inflexible dongles?
Take a look at this!

- Floating Network Licenses
- Usage-Based, Shared Licenses
- Easy to Integrate
- Time-Bombed Demos
- Available on Unix, Windows, Novell and Macintosh *and a lot more!*

• Reliable • Free support; upgrades for 1 year
• No royalties • Steep discounts on multiple platforms
• Compliance to standards

HERITAGE SYSTEMS

Tel: 0181 959 3611

Fax: 0181 906 1700

Email: hsystems@cix.compulink.co.uk

***FREE* Evaluation Copy – Call 0181 959 3611**

CIRCLE NO. 066

Embedded 386/486 Development Demystified

If you are contemplating using the new embedded 386 and 486 CPUs and are grappling with the complexities of 32-bit protected mode programming then help is at hand! We are the UK's specialists in ROM-based 386 development and as such can offer a complete range of dedicated software tools for all non-PC 386/486 applications. Our tool support extends from powerful protected-mode debuggers and locators for 32-bit compilers up to complete embedded development suites and our own real time in-circuit emulators.

32-Bit 386 Tools Brief Overview

SOFTSCOPE III - **Protected mode debugger for all 386 variants with TX386EX in-circuit emulator interface. Re-configurable monitor kernel initialises CPU and can be included royalty-free in final code.**

Phar Lap TNT Embedded 386 Tools Suite - **The ultimate software development kit for all ROM-based applications. Includes Visual System Builder, TNT real time kernel, LinkLoc protected mode linker and locator for 32-bit C/C++ compilers.**

For further information on your 32-bit 386 development options, contact us now for a comprehensive data pack.

Hitex (UK) Ltd.
Tel: 01203 692066 FAX: 01203 692131

hitex

CIRCLE NO. 068

Too much, too late?

It would seem that COSE represents the big collaboration that Unix needs. But it appears the major players are trying to balance the years of neglect with one all-encompassing goal. **Niall Mansfield** investigates.

COSE, the Common Open Systems Environment, is supposed to unify all the different flavours of Unix from all the various vendors. Spec 1170 is the 'Single Unix Specification' and defines the base Unix. CDE is the Common Desktop Environment, a standard look and feel, with standard tools for managing files, launching applications, printing, and so on.

All this sounds wonderful. It's what we've been waiting for since Unix started to take off in the commercial world. But by the time I come to the end of writing this article I *know* I'll be in a foul temper, kicking the cat, whining about the Unix computer industry and vowing to take up basket weaving or blacksmithing or something sensible. Because yet another great idea looks like it will never take off. At heart the big Unix companies still think they can go it alone. The great thing about standards is there are so many to choose from. Well, COSE proves that and also the other old adage: Unix is the system of the future, and always will be. But CDE looks like it will actually deliver something quite useful. So it's not all doom and gloom. Well, maybe not.

In the beginning

COSE is the Common Open Systems Environment. It was founded in March 1993 by IBM, SCO, Sun, Hewlett-Packard, Novell and Unix Systems Laboratories. The stated aim was 'to accelerate the process by which open system software specifications are defined and submitted to recognized industry standard forums.'

'More effective cooperation was the key to meeting the demands of users for consistency and interoperability among software suppliers.' Or at least that's what they said. In fact what happened was Microsoft NT. It terrified the Unix boys. They considered it a serious threat. So much so that if they didn't do something quick, they believed Gates would sneak in behind while they squabbling amongst themselves, and steal all the market. They agreed to cooperate to fend off the Microsoft threat. But instead of setting up yet another standards body or consortium, COSE was defined as a *process* for getting standards formalized more quickly.

In fact what happened was Microsoft NT. It terrified the Unix boys

The formation of an open systems standard follows the steps in Figure 1 to a greater or lesser degree. In the early days of a new technology, there are almost by definition no standards. Instead, different vendors produce their own technologies for particular purposes. The customers then say which ones they want by 'free market selection'. We buy the ones which we think are best. Later on when the market leader technologies are established, we, the users, demand that different vendors' systems work together. For example, I want my HP workstation to share files with my Sun server and be able to run X applications from my IBM box too. For this to happen, the technologies must be standardized, which in turn means a standard must be defined. Of course, each vendor wants their own variant to win the standards race, so they battle it out for a few years. But eventually we end up with a new networking standard or a new API for something. And we all work happily every after. Or at least that is how it works in theory.

COSE is to act as a catalyst in this, to speed things up. Instead of beating their own separate drums at the draft standard stage, the COSE folk get together, come to a consensus view of the new technology, then bang a common COSE drum together. That sounds fairly trite. More significantly, they may work together to have a sample implementation developed fairly early on, which helps drive the standards process and helps avoid the pitfalls which a paper-only standard can easily fall into.

A means to an end

The trouble with this nebulous 'COSE is a process' definition is that it's about as long as a piece of string. It may turn out in the medium term to have very little effect, because the vendors still battle it out to have their own technology made the standard. Or it could be a major change in the computer industry. Here vendors would really work together by forming ad-hoc working groups very early in the life-cycle of a technology. They would then produce sample implementations, distribute them widely to get as much feedback and testing as possible and so promote an independent and good standard. The X window system was almost a miracle of cooperative working as it developed and grew.

To be optimistic (but only briefly) it looks like some areas will see a lot of vendor cooperation. The Open Software Foundation

(OSF) has undergone a transformation. Originally it was little more than an attempt by IBM, DEC and HP to stop Sun and AT&T dominating the windows-on-Unix market and the Unix operating system itself. The OSF developed the Motif GUI which has beaten Sun's OpenLook and is now the industry standard. It also created the OSF/1 operating system. Although it is really only available on DEC machines now. And as DEC is rapidly becoming little more than a fast chip maker, that doesn't say much. As a result of the COSE process however, Unix International has been disbanded and all the major vendors (including Sun) have joined the OSF, which has rejuvenated itself into a new form. Since COSE is only a process, a body is needed to coordinate it all. The New OSF fills this role. The OSF is moving from being a technology developer to being a project coordinator and a technology licensing body.

Fast track standardisation

The new OSF uses the concept of Pre-Structured Technology (PST) for acquiring new technologies. PSTs are formed by OSF members who will sponsor a project's development. Projects are coordinated by the OSF's Architectural Planning Council. The sponsors use OSF to manage the project, but development work is carried out by a Prime Contractor selected by the sponsors. Sponsors may acts as subcontractors and do some of the development work themselves. Once the prime contractor has refined the technology and prepared a formal draft specification and a reference implementation it will then be submitted to X/Open (or another standards body) for adoption as a standard under a 'fast-track' process.

The stated scope of COSE is enormous. Allegedly it will cover virtually every aspect of computing, from object technology to multimedia. So far there are two areas where COSE is starting to deliver: these are Spec 1170, the common Unix APIs and CDE, the Common Desktop Environment. We'll cover each of these in some detail below. There are a number of additional areas of activity and standardization where COSE working groups are starting to function

The standards for 2D and 3D graphics under X are being addressed by the X Consortium, with Xlib for 2D, PEXlib for 3D, and XIE for images. COSE's contribution is to help speed things up by supporting the existing standardization efforts.

The X window system was almost a miracle of cooperative working as it developed and grew

In terms of multimedia, COSE is working with an existing group, the Interactive Multimedia Association, on standards for integrated video, audio, and text-based objects.

All the COSE folk have committed to the Object Management Group's (OMG) CORBA (Common Object Request Broker Architecture). They will endeavour to produce software which complies with this.

With distributed computing, the agreement is to support Sun's Open Network Computing (ONC/ONC+), OSF's Distributed Computing Environment (DCE) and Novell Netware clients on each vendor's system. TCP/IP is used as the low level transport. My own view is that to call this a standard is like saying anyone who speaks Japanese can automatically converse with Greek or Portuguese speakers. In other words it's nonsense. The only sensible bit is that there is a commitment that each of the three 'standards' will be available on any COSE-participating vendor's system.

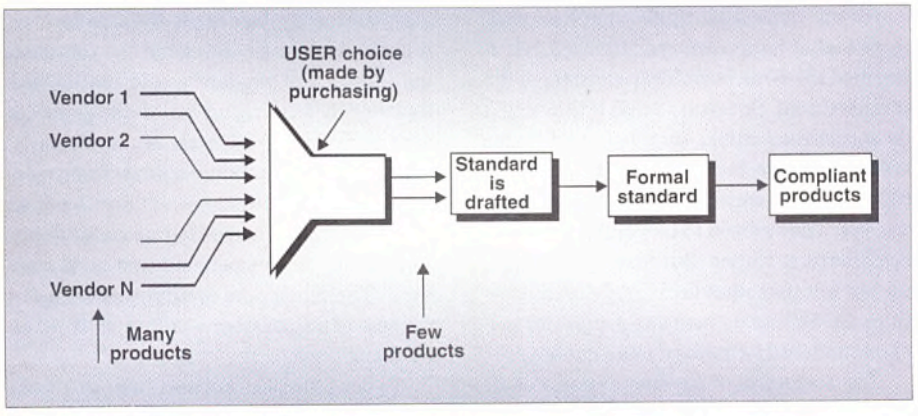


Figure 1 - Creating a new standard

Rather sensibly, system management has been recognized as one the disaster areas of Unix. The COSE aim is to develop standard APIs and basic technology, on which third parties can build good tools. COSE may implement some basic tools. Areas of particular interest are software distribution and installation, printing, user administration, licence management and backup/restore.

The public Windows interface or WABI was Sun's rather successful attempt to annoy Microsoft in public. The idea is to force the Windows application binary interface (WABI) into the public domain so that other people can develop their own implementations, on Intel or other machines, and perhaps on top of the X window system. Sun implemented WABI on its own systems by taking a Windows program and translating it before it is run, mapping the Intel instructions onto SPARC ones, and mapping the windows system calls onto Solaris equivalents.

The common Unix APIs

A little over a year ago, lots of vendors agreed to support a common set of APIs to the kernel, C libraries, networking and commands and utilities. The specification of this set has been called Spec 1170, because it contains 1170 API components.

Spec 1170 started from the base of X/Open's XPG4 which in turn is based on IEEE's POSIX. Then the COSE folk rummaged around in the top 50 applications, found any system calls and interfaces in these programs which weren't already in XPG4 and added them into Spec 1170. Even though this looks like an extraordinary way to go about developing a standard (rather like bolting the asylum door after the inmates have bolted) most of the bits that CODE added were quite sensible, such as BSD sockets and SVID curses support. Of course, the problem with this is that any new APIs necessary, for example to handle threads for multiprocessor systems, were not to be found in existing old programs, so the new 'standard' will have to be *enhanced* soon.

X/Open has published Spec 1170, calling it XPG/4.2. It is developing test suites to validate systems against this specification. If a system passes the validation suite it can be called Unix, otherwise it can't. Probably as a result of COSE, Novell, who acquired the rights to the Unix name when it purchased Unix Systems Laboratories, passed the Unix trademark over to X/Open, so at last Unix really is not controlled by a for-profit company.

So on balance Spec 1170 looks like a 'good thing'. POSIX went a long way to simplifying life for developers, because they no longer

had to spent most of their time coding their way around the differences in the various flavours of Unix. Spec 1170 may turn out to be an even better step along this road.

CDE

One of the great joys of open systems from a technical perspective, is that everything is so modular and customizable and replaceable that our own window layout and way of doing things can be made quite unique. Unfortunately this is what drives commercial users insane. Everywhere they look, they find something different. So for perfectly sound reasons a standard environment is required. Instead of having a thousand different desktops, there should be just one for Unix, And that's what CDE is about.

The Desktop Environment uses the Motif GUI. The desktop itself contains a Front Panel which allows us to control our workspace and launch commonly used applications. There is a File Manager, an Application Manager and a Session Manager, which lets us preserve the state of our desktop and applications across a logout/login again sequence. There is also a print server and a help system. Standard utility programs include a multi-user calendar, a mail program, a calculator, a text editor, an icon editor etc. In others words, it's the Unix equivalent of what you get with Microsoft Windows. This is really happening. IBM has shipped an early release. You can also get early implementations for Sun, HP and Unixware too. There are also companies porting CDE to SGI and MIPs machines. The logical conclusion is that you get as much desktop standardization across the various vendor's Unixes as you do on an MS-Windows PC.

It's rather like bolting the asylum door after the inmates have bolted

Conclusion

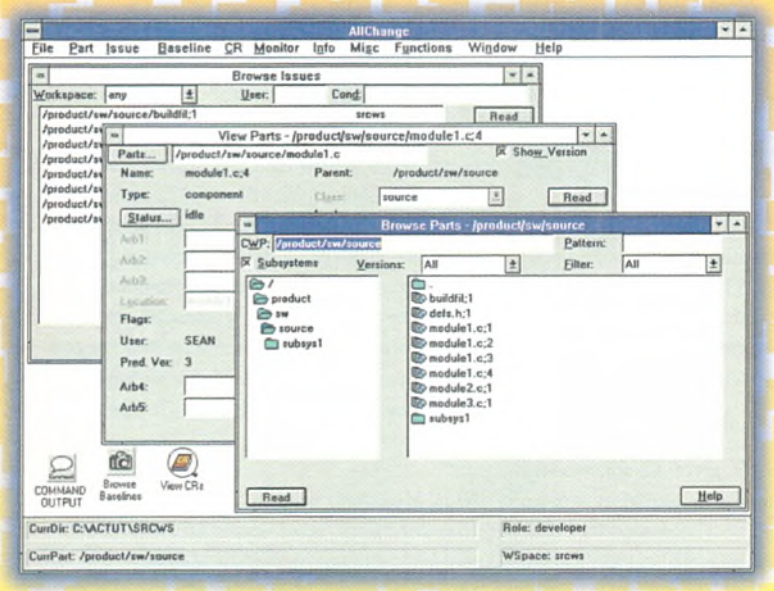
So why, I hear you ask, given all this wonderful cooperation and vendor harmony, am I so negative about COSE? The reason is I don't think it's going to happen, or at least in nowhere near as big a way as it's been hyped. With the possible exception of CDE, most of the tangible achievements were already complete or under way. Yes, it's true that CODE is only a catalyst, so maybe I'm expecting too much. But it all sounds like the emperors new clothes. And it's *astonishingly* difficult to get any information about COSE from anywhere that isn't about 18 months old. Given the huge range of technologies that CODE is supposed to be addressing, you'd expect to hear news all the time of stuff coming to fruition. Either there's a conspiracy of silence or else a lot of back-pedaling.

My own view of the future is that we will never have a unified Unix world. This will open the door to Windows-NT, which will dominate the market for bigger machines. Then we won't have open systems any more, because the software will be completely controlled by one monopoly company. By being dependent on a single vendor, a new generation of programmers will have to learn the hard way, why open systems are necessary and a 'good thing'. ■

Niall Mansfield is the managing director of User Interface Technologies Ltd. he can be reached on email as niall@uit.co.uk .

Bring it all together

with AllChange for Configuration Management



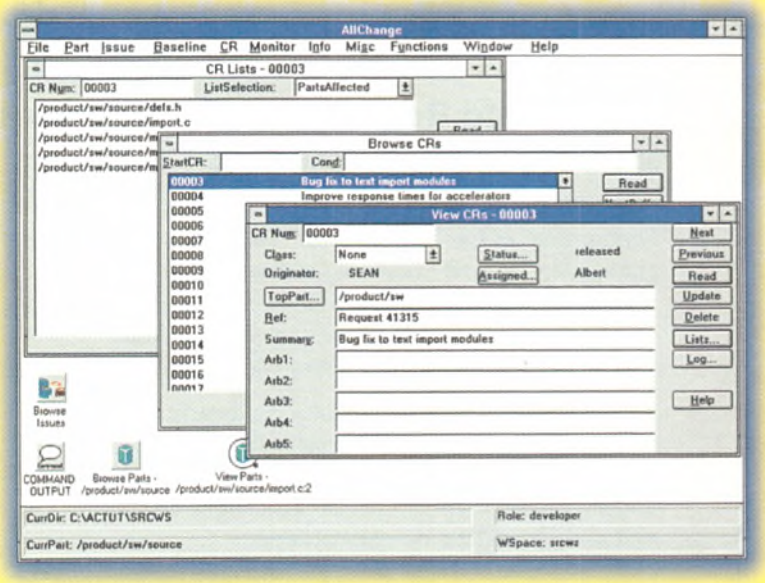
- Configuration item identification
- Version control
- Workspace management
- Baselines
- Bug tracking
- Life-cycle management
- User roles for access control
- Change requests
- Configuration build
- Release management
- Open interface to other tools

What is AllChange . . .

AllChange is a complete change control and configuration management system that may be tailored to site and project requirements. It is a database based system enabling it to know about the relationships between items and to control off-line items such as hardware or paper documents, as well as on-line files. Its unique action triggers enable **AllChange** to actively participate in enforcing procedures. It is a truly unique configurable system enabling it to match your specific requirements.

*"We chose **AllChange** because of its full C.M. functionality and its flexibility which enabled us to emulate and improve upon manual procedures already in place"*
– David Gilmore, Charterhouse Bank

*"We chose **AllChange** because we were impressed with the flexibility of the product, reassured by the manuals and pleased with the brainy support"*
– Simon Dalley, Solartron Transducers



Our products will help you with standards (e.g. ISO9000, BS5750) and are backed by our outstanding support.

Platforms: **AllChange** is available for Windows and UNIX.

CIRCLE NO. 069

NEW VERSION 3

Call for more information

01392 217670

INTASOFT

Quality tools for professional software developers

INTASOFT LIMITED, Tresco House, 153 Sweetbrier Lane, Exeter, Devon EX1 3DG, England · Tel: 01392 217670 · Fax: 01392 437877

Master of the house

Mailing lists are an alternative to USENET news as a topic based information system because they are easier to set up. However the maintenance of a large number of subscribers can be a headache. **Paul Richardson** describes a package that will simplify the task.

The principle behind using a mailing list is that any message sent to the list's mail address will be sent by email to all of the list's subscribers. A mailing list manager (MLM) is a suite of programs that help to automate one or more mailing lists, usually providing facilities such as user subscription management, moderated lists and interaction with the Message Transfer Agent (MTA).

There are a variety of MLMs available, the most common being ListProc, LISTSERV and Majordomo. As in most cases, they vary widely in their characteristics. Some are veritable behemoths, stuffed with every feature that any user could think of, while others are newer and slimline, keeping frills to a minimum. For a more detailed introduction to MLMs and a comparison between the various available implementations, read the MLM FAQ (see bibliography).

For the purpose of this article I am going to introduce you to the Majordomo mailing list manager, an implementation for Unix. I have chosen Majordomo for a variety of reasons, among which are its popularity, its manageable size and easy extensibility (since it is written in Perl) and I have recently installed it.

User Commands	Description
subscribe *list [address]*	Subscribes the sender (or *address*) to *list*
unsubscribe *list [address]*	Remove the sender (or *address*) from *list*
which *[address]*	Returns which lists the sender (or address) is on
who *list*	Returns the members of *list*
info *list*	Returns introductory text for *list*
index *list*	Returns a list of files in *list's* archive
get *list filename*	Retrieves *filename* from *list's* archive
lists	Returns the lists handled by this server
help	Returns help on each of the commands available
end	Indicates where majordomo should stop processing
List Owner Commands	
approve *passwd [subscribe|unsubscribe]*	
list address	Approves a subscription or unsubscription request from *address* to *list*
passwd *list old_passwd new_passwd*	Changes the list owner's password for *list*
newinfo *list passwd*	Changes *list's* introductory text
config *list passwd*	Returns the configuration file for *list*
newconfig *list passwd* (followed by config)	Installs a new configuration file for *list*
writeconfig *list passwd*	
mkdigest *digest list name passwd*	

Figure 1 - Table of Majordomo commands

Who needs a manager?

If we are thinking of setting up a mailing list, before we proceed, it is worth considering whether we need the assistance of an MLM. If subscribers to the list do not change frequently, then it is perfectly feasible to run either a *moderated* or *open* list without any support from an MLM. To run a moderated list, set up the mailing list address as an alias that points to the moderator. Set up another alias for all members of the list, such that when the moderator wishes to send a message to the subscribers, he sends it to the alias. The latter alias can be one that is only known to the moderator's mail reader/composer and so is easy to alter.

An open list is even more straightforward, requiring merely to have an alias for the mailing list address in the Message Transfer Agent (MTA), which for Unix is usually `sendmail`. The alias should contain all the members of the list. Thus if a message bound for the list is received the MTA will simply reflect it to all the list's members. The difficulty with this approach is that usually MTA aliases can only be altered by the system administrator. However, in the case of `sendmail` the alias can be defined to be the addresses contained in a file. This file can be made available for the list maintainer to modify.

The house-keeper

As an etymological aside, the word 'majordomo' has its' roots in the Latin for *Master of the House*. It is used these days to mean 'a person who speaks, makes arrangements, or takes charge for another', which seems quite an apt description of its task in managing the mailing lists.

So, given that we have chosen to use the Majordomo MLM, what facilities does it provide? The primary capability is that of automating the process of subscribing and unsubscribing to and from the mailing list. For an *open* or *auto* list, ie one for which no approval is required for subscriptions, absolutely no intervention is needed by the list owner to manage subscriptions. Users control their own subscriptions. Majordomo also supports the idea of a *closed* list in which the list owner has to approve all requests to subscribe and unsubscribe.

In moderation

Both *moderated* and *unmoderated* lists are supported by Majordomo. A moderated list is one in which the list owner sanctions and/or edits a submission before distributing it to the subscribers. Support is provided for automatically archiving retrieve messages that have been sent to the mailing list. Users are able to retrieve archive files. As an alternative, scripts are provided which create digests of messages. A digest is just a mail message that is comprised of several others, see RFC 1153. Users can subscribe to a separate *list-digests* list.

Users can retrieve files placed in the mailing list archive. One point worthy of note (especially to SCO users), is that Majordomo will work with MTAs other than `sendmail`, such as `MMDF`.

Mail me

For users and mailing list owners alike, most interaction with Majordomo is via email. To perform administration on a list, such as subscribing to it or changing the information associated with a list, email is sent to the `majordomo` alias and a command is specified in the body of the message. Figure 1 contains the full list of messages that can be sent. For an example of what Majordomo sends back in response to a subscription request, see Figure 2.

A guiding principle in the design of Majordomo was to separate the functions of administering the MLM and maintaining a list. It was to facilitate this dichotomy that so many commands were made available by sending email to `majordomo`. All list owner commands require passwords to be supplied.

To post a message to a mailing list, simply send the message to the mailing list alias. It will be forwarded, either straight to the members, in the case of an unmoderated list, or to the list owner.

Under the Hood

Majordomo is primarily written as a Perl 4 program, the majority of which is in the file `majordomo` which handles the subscription/unsubscription process The program `resend` is also important as it handles the routing of messages sent to a list. However there is a small amount of C code in the file WRAPPER.C, which when compiled creates a program called `wrapper`.

The need behind `wrapper` is rather interesting and is of special interest to anyone contemplating writing a Perl program. Majordomo's author, Brent Chapman, decided that he would need majordomo to run `setuid`, so that configuration files and mailing lists that were owned by a special `majordom` account could be accessed. However when he tried this out he found that Perl refused to cooperate. The problem revolves around a feature of Perl known as `taintperl` that kicks-in when a program is run with `setuid` set, and applies some rigorous security related checks. The unyielding nature of these checks proved too much for Brent. He was unable to coerce `majordomo` to perform the necessary tasks. So he gave up and wrote a simple wrapper in C that sets the user ID and group ID appropriately before calling `majordomo`.

It must be said ... that Majordomo was not the easiest package that I have ever installed

Stick together

The glue that holds the Majordomo system together is the MTA's *aliases*. An aliasing, in this context, is the replacement of one recipient address with one or more different recipient addresses. Thus, an alias could simply cause mail addressed to *x* to be delivered to *y* instead. Equally it could cause mail to be sent to *y* and *z*. Aliases also offer us the ability to do some more clever things such as appending mail to a nominated file or passing mail to a specified program for processing. In addition, the alias' expansion can be contained in a file, a technique that is crucial to the operation of Majordomo. The aliases in Figure 3 illustrate the use of several of these forms.

This is a stripped-down sample set of aliases Picking out the salient points we see that any mail addressed to the `majordomo` alias itself will be passed to the `majordomo` Perl program for processing. `majordomo` parses the contents of the mail message, acting accordingly. Its principle task is to handle adding or removing mail addresses from the file that contains the list's subscribers. In our example this is `/usr/local/majordomo/lists/testlist`. But it understands all the commands listed in Figure 1.

Mail sent to the list itself (in our example named `testlist`) will be passed to the `resend` Perl program. This has a crucial role to play in rearranging the message's header lines before routing it on to the appropriate place. The 'appropriate place' depends on whether the list is moderated or not. If it is not moderated then the message can be sent straight out to the list's subscribers. `resend` does this by routing the mail to the `-outgoing` alias (the alias `testlist-outgoing` in this example). The syntax used for the `testlist-outgoing` alias indicates that the contents of the file `/usr/local/majordomo/lists/testlist` contains the list of aliases that the mail should be sent to, ie the subscription list.

If the list is moderated (achieved by passing `resend` a `-A` parameter), `resend` will route the mail to the list owner (the `-owner` alias) who then has the opportunity to edit the message before sending it back to the list with an additional 'Approved:' header line. This indicates to `resend` that the message has been passed for sending out to all the subscribers.

Getting hold of it...

The current version of `majordomo` is 1.93, but I have only been able to find it at the primary distribution site (as specified in the bibliography). However, I found connecting to this site was a bit flaky and even when I could connect, it was slow. Version 1.92 is a good stable version and is available from many FTP sites, including `ftp.demon.co.uk`. Do not use versions before 1.92 as there is a security hole that allows intruders to gain access to the account that runs the Majordomo software, even if the site has firewalls and TCP wrappers.

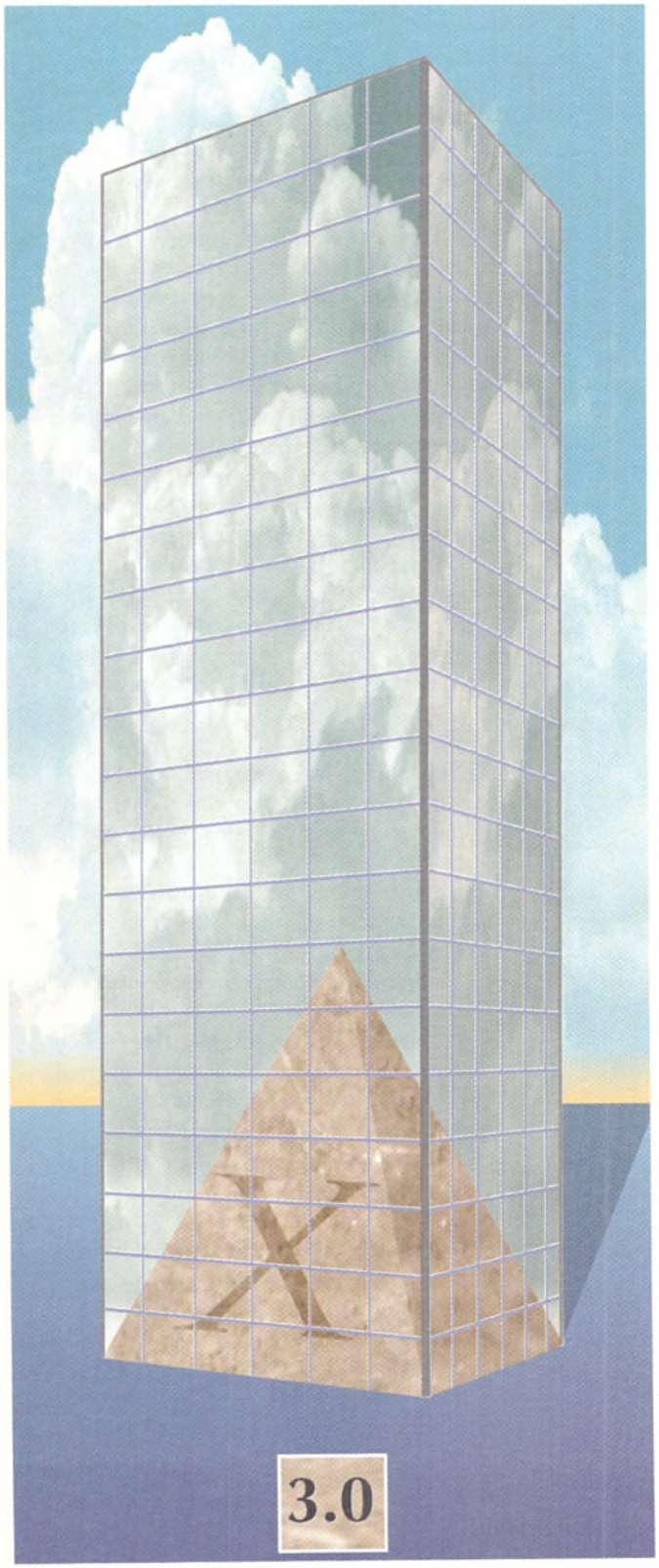
Once downloaded, you will need to obtain a full installation of Perl if you don't already have one. The recommended version for

```
Welcome to the majordomo-announce mailing
list!

If you ever want to remove yourself from this
mailing list, you can send mail to "Major-
domo@Chadwyck.co.uk" with the following com-
mand in the body of your email message:

    unsubscribe majordomo-announce
PaulR@Motiv.co.uk

Here's the general information for the list
you've
subscribed to, in case you don't already have
it:

This list is for announcements of new releases
of the Majordomo mailing list manager.

For further information, contact:

Postmaster and list manager, Chadwyck Healey
Ltd
postmaster@Chadwyck.co.uk
```

Figure 2 - Automated response from Majordomo to a subscription request

ADVANTAGE™

Xbase Server



Brings the Power of Client/Server to Your Existing Applications Without Rewriting a Single Line of Code.

- *Eliminates Index Corruption*
- *Increases Multi-User Performance*
- *Improves Concurrency Management*
- *100% Compatible with Existing Applications*

The Advantage Xbase Server provides networked CA-Clipper applications with the benefits of client/server technology, often without rewriting a single line of code. Multi-user applications benefit from significantly improved stability, integrity and performance. Rather than move huge data files across the network, Advantage Xbase Server applications transmit database access requests from the client application to the NLM running on the server. By processing those requests on the server, the Advantage Xbase Server can eliminate index corruption, increase application performance and reduce network traffic.

"The Advantage Xbase Server makes good on its promise of increased speed and better reliability. It's easy to install and couldn't be easier to use."

Craig Yellick
Data Based Advisor, May 1994

The Advantage Xbase Server is specifically designed to work with existing Xbase applications. It consists of a CA-Clipper Replaceable Database Driver (RDD) which integrates seamlessly with existing CA-Clipper applications, and a NetWare Loadable Module (NLM) which runs on a NetWare file server. The Advantage Xbase Server works directly with DBF files, and supports industry-standard NTX, CDX and IDX indexes.

QBS Software Limited
10 Barley Mow Passage,
London W4 4PH
Ph: +44 181 994 4842
Fax: +44 181 994 3441
BBS: +44 181 747 1979

QBS SOFTWARE

yes NetWare Tested and Approved

Extended Systems Ltd
In the UK phone: 0800 614780
Ph: +44 1705 875075
Fax: +44 1705 875095

CIRCLE NO. 070

Majordomo is 4.036, though an older or newer version (including version 5.0) may work. I retrieved Perl from `ftp.demon.co.uk`, but it is a very popular package that you should have no problems finding on your fave FTP site. If you want to know more about Perl then you need look no further than the December and January back issues of EXE magazine, in which Niall Mansfield introduces the language.

Major probs

It must be said at this point that Majordomo was not the easiest package that I have ever installed. The configuration is messy, being split across configuration files and the sendmail `/etc/aliases` file. In addition, there are several components to the executables; the C wrapper, the main `majordomo` program and several supporting scripts. And finally, the documentation inspires less than confidence in regard to the correct ownership and permissions of the various files, stating in one point that 'By far the biggest problem in setting up Majordomo is getting all the permissions and ownerships right.'

The installation instructions in the README file are very skimpy, though there is a fairly good step-by-step installation guide in a file containing a chapter of a forthcoming O'Reilly book (see bibliography). Even these instructions are far from satisfactory as they relate to a much older version. I've picked out a few points to watch out for.

The wrapper program, compiled with no modification under K&R C, is supplied with SunOS 4.1.2, though if you are using a POSIX compliant Unix such as SunOS 5.x, be sure to uncomment the appropriate flags in the Makefile. Be careful to ensure that the path to Perl specified at the top of each Perl program is correct. The path can be found at the top of the file and will look like:

```
#!/path/to/perl
```

Take care over setting the configurable macros in the Makefile. These not only define the group and user that the system will run under, but also the values of the environment variables that are passed to the wrapper. Getting these wrong can result in a lot of undoing or uncertainty if you end up having to set ownerships by hand.

Remember to run `newaliases` after you have added the extra aliases to `/etc/aliases`.

`majordomo.cf` is the principle configuration file and hence will be the focus of a lot of attention if things don't go according to plan. Spend time on it at this stage to get it right and it will pay dividends. Figure 4 is an example `majordomo.cf` with comments that explain the meaning of the various settings. For the purposes of brevity I have not listed the more obscure parameters. For a basic installation you will probably only want to modify these parameters. But rest assured that all the options are well commented in the `sample.cf` supplied with the package. Take note that this file is Perl code, so make sure that you preserve the syntax. For example, ensure that all lines end with a semi-colon.

```
# The following three aliases are needed for Majordomo itself -
# they handle the email commands sent to majordomo@... passing all such
# mail in to the majordomo Perl program.
majordomo:                      "|/usr/local/majordomo/wrapper majordomo"
owner-majordomo:                postmaster
majordomo-owner:                postmaster

# The following four aliases have been set up to support the unmoderated list named testlist.
# All mail sent to testlist@... will be passed in to the wrapper Perl program.
testlist:                       "|/usr/local/majordomo/wrapper resend -l testlist
                                -h Chadwyck.co.uk testlist-outgoing"
# In this next alias, the expansion (ie the list's subscribers) is contained in a file.
testlist-outgoing:              :include:/usr/local/majordomo/lists/testlist
owner-testlist:                 PaulR
testlist-approval:              PaulR
```

Figure 3 - Sample set of aliases

```
# $whereami - The mail domain to be used.
$whereami = "Motiv.co.uk";
# $whoami - The email address that is to be used for Majordomo.
$whoami = "Majordomo@$whereami";
# $whoami_owner - The email address of the Majordomo administrator.
$whoami_owner = "Majordomo-Owner@$whereami";
# $homedir - The path to the supporting Perl scripts.
$homedir = "/usr/local/majordomo";
# $listdir - The path to the mailing lists.
$listdir = "/usr/local/majordomo/lists";
# $log - The pathname of the log file.
$log = "$homedir/Log";
# $mailer - The program and accompanying arguments to the program
# that is used to send mail.
$mailer = "/usr/lib/sendmail -f\$sender -t";
# Majordomo will look for support files related to $list in
# directory "$filedir/$list$filedir_suffix.
$filedir = "$listdir";
$filedir_suffix = ".archive";
```

Figure 4 - Example majordomo.cf

Is everythink okay?

I applied a stepwise approach to testing the installation, initially ignoring the mail side and just checking the operation of the wrapper and the `majordomo` program. Running `wrapper` with no arguments should result in a usage message. If this fails to appear, it could be a pathing problem, in which case try `./wrapper`. Alternatively `wrapper` may not be an executable.

Having sorted out `wrapper`, I turned my attention to the `majordomo` Perl program. You can check whether `majordomo` is executed or not with the command line:

```
./wrapper majordomo
```

This test checks that the wrapper's built-in path to `majordomo` is correct; that `majordomo` is executable and that Perl is being found and is executable.

Next I attempted to verify that `majordomo` would respond correctly to a mail message with 'help' in the body. To do this I used the following command:

```
./wrapper majordomo <
To: majordomo
```

```
From: PaulR@Motiv.demon.co.uk
help
^D
```

The appropriate response to this is for `majordomo` to fire-up `sendmail` and send a help message back to `Paul R@Motiv.demon.co.uk`. Failure to do this is most likely to be caused by an incorrect entry against the `$mailer` option in the `majordomo.cf` file. The `$mailer` entry specifies the call that must be made by `majordomo` in order to start `sendmail`.

The next technique is useful for debugging other mail-related problems. Here you call `sendmail` directly specifying the `-v` ('verbose') flag that shows not only what `sendmail` is getting up to, but also pipes the response from the `majordomo` program to standard output. Thus this an excellent technique for checking the correct operation of the aliases file. It can also be used to determine whether `majordomo` has failed to assert locks. If the test seems to indicate that the aliases are not set up correctly, you should run `newaliases` to create a `dbm` database of aliases from the contents of `/etc/aliases`.

I must tell you about a problem I encountered which I hadn't been prepared for by any of the documentation. After several hours of anguished experimentation I realised that I had encountered a version of `sendmail` that had not been configured to allow any aliases to executables! This was actually rather sensible as the host was in a sensitive position on the network. But it completely screwed-up my chances of getting Majordomo to work. The work-around was to create 'mirror' aliases on the host that merely passed on the mail to a real Majordomo installation on a less sensitive machine that could handle running executables from aliases.

Majordomo creates a log file that records actions such as subscription. So you should check the log file for messages indicating correct operation. Of course it is also important to check that mail sent to a list is routed correctly according to whether it is a moderated list or not. Failure in this department is probably down to the incorrect use of the `resend` Perl program.

Majordomo software
ftp://ftp.greatcircle.com/pub/majordomo/majordomo-1.93.tar.Z

Perl software
ftp://ftp.demon.co.uk/perl/perl-4.036.tar.gz

Mailing list FAQ
send mail to majordomo@pop.psu.edu, with body get file mlm-software-faq

Majordomo FAQ
http://www.math.psu.edu/barr/majordomo-faq.html

Mailing Lists
Several including Majordomo-Users, Majordomo-Announce and list-managers, all at Majordomo@GreatCircle.com.

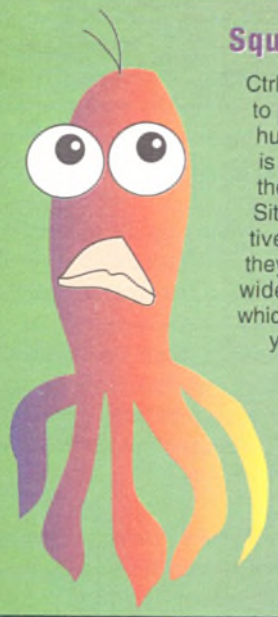
Book
Managing Internet Information Services, Cricket Liu et al., O'Reilly & Associates, Inc.

RFC
RFC1121 Problems with the maintenance of large mailing lists.

To Close

As ever, I recommend following some of the pointers for more information specified in the bibliography. Majordomo does have its problems, but seems to cover the middle ground for requirements. A fairly static subscription base is better handled manually, while a very dynamic list of subscribers would bring the host computer to its knees, as instances of the majordomo program are spawned for every request. If the latter is your problem then I would suggest investigating some of the MLMs that scale better, such as ListServ. ■

Paul Richardson is a Director of Motiv Systems Ltd, a consultancy specialising in the Internet. He can be contacted on 01223 576318 or by Email at PaulR@Motiv.demon.co.uk.



Squids R Us

Ctrl Break's Number 1 Internet site at time of going to press simply has to be the one entitled 'Have you hugged your squid today?' Don't laugh. Ctrl Break is sure that there are many cephalopod owners out there seeking contact with like-minded individuals. Site visitors can therefore read-up on all the definitive areas of Squid academics. And then, should they feel the need, they can also browse through a wide and varied selection of Squid related topics which the helpful Site-owner has also calculated for your delectation and delight. 'What if Madonna was a Squid?' for example. Or for the more avaricious of you tentacled mollusc fans 'The $14 billion dollar Squid'. Yet another worthy contribution toward the Information Superhighway, fuelling high debate on global issues. Good fun though.

Join the Squid Squad at...
http://www.algorithm.com/squid/squid.html

Phe, Phi, Pho...um.

Just how does one pronounce the moniker of that hot new RAD tool from Borland? We Brits seem convinced that Delphi should be pronounced with an 'ee' ending. However, the Yanks are equally adamant that it's with an 'aye'. Ctrl Break, ever keen to clear up international misunderstandings such as this, had a quick word with those in the know. Well errrm, having searched far and wide for the answer to this most philosophical question we came up with nowt. If you think you know the answer send us a postcard to the address opposite explaining why. Best of the bunch gets a free copy of Delph-ee (or should that read Delph-aye?).

Big Blue is Top of the Patent Pops

Readers of Ctrl Break may be interested to know that IBM applied for 1,298 patents in 1994. IBM thinks you're going to be interested anyway, as this apparently puts the Granddaddy of all computer companies at the top of the US Patent rankings. With '199 patents, or 18 percent more than second place Canon KK' as the press release trumpets. Ctrl Break hesitates to suggest that this 'news leak' is an attempt to prove that there's life in the old dog yet. So 'Congratulations Big Blue', on having all the patents. What a shame that Bill Gates has got all the dollars.

But on a lighter note...

If you hurry you may be able to snap up the last few tickets for the latter of two laugh-a-minute seminars. Unfortunately, you've already missed the two day stint on Digital Battlefields which ran from 27-28 March and incorporated such enlightened features as The Need to Digitise the Battlefield and Fratricide. But you may still be in with a chance of catching Tactical Electronic Warfare which runs from 3-5 April. This kicks off with Electronic Warfare: An Immutable Criticality and gets better from there onwards. Ctrl Break suspects that there may be one or two Luddites out there who protest at the overtly violent nature of such talks. These reactionaries would no doubt have us believe that the use of such state-of-the-art computer technology to calculate and effect the demise of the human population is little short of insanity. Stuff and nonsense good chums. If it's anything like some of the more commercial software applications it could quite easily crash before it kills anyone. And anyway, what's a population between friends?

WIN ACCESS DEVELOPERS BUNDLE

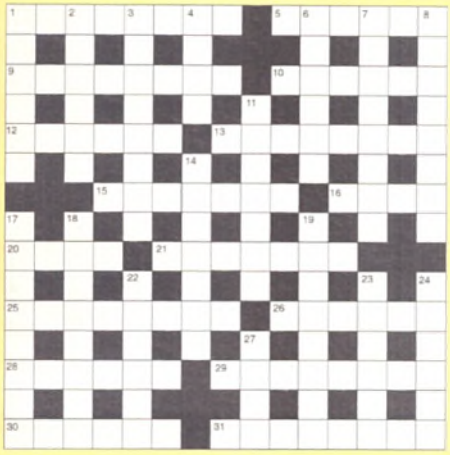
A bumper-bundle of Microsoft Access, Access Solutions, Access Toolkit and Access Training Videos, worth £700, is up for grabs this month. To enter, simply answer the question below:

How many copies of Access have been sold in worldwide since its launch?

(a) £250,000
(b) 1,350,000
(c) 3,000,000

Send your entries to the address on the next page. The first entry picked out of our hat will win.

PRIZE CROSSWORD



ACROSS

1. Buses for thieves in the night? (8)
5. Used a 15 to step through the software (6)
9. Hit rate of sporting types (8)
10. Crudely thrash in scalar ruptures (6)
12. Maybe deal with multiples of 2 in a fair way (6)
13. Working eight bits at a time (8)
15. Useful little routine! (7)
16. Spill gore when you meet him in the dungeon (4)
20. Meat to software always... (4)
21. ...and label every chunk of it? (7)
25. & 26 Maybe it has a Centronics port (8,6)
27. Young doctor may arrive when he should (6)
29. Precise splatterer when painting (8)
30. Common chunks of hardware in Soho (6)
31. Goes through it all again conservatively (8)

DOWN

1. Unusual soccershot on file (6)
2. Collect 20 or rosebuds (6)
3. Small analog signals ptoduced by veal stew (8)
4. Mysterious 16 of the Himalayas (4)
6. 1 dn loses head for king in the drive (6)
7. Bearing of printer being returned (8)
8. Poor layer in a diode (8)
11. Hardware, software, liveware - the lot (7)
14. One in a conspiracy may provide the picture (7)
17. Smartly flexible in packet routing (8)
18. 31 the routine (8)
19. Its object is to put the code together (8)
22. Once 10p to its friends (6)
23. Such a display makes sense (6)
24. Networks? That fits snugly! (6)
27. Such pointers are poor when in church

Solution to March's Crossword

ACROSS: 1. PALMTOP 5. ZEROING 9. OLE 10. OUTPUT 11. ROBOTIC 12. TWO 13. TAHITI 14. QUANGO 15. ALB 17. ROSTERS 19. DARES 21. DEBUG 24. THREADS 28. SOS 29. STAIRS 30. OUTRUN 31. NET 33. TEAWARE 34. EPITAPH 35. SPA 36. PROTECT 37. MALSTRO

DOWN: 1. PLOTTER 2. LATCHES 3. TRUSTEE 4. POST 5. ZERO 6. REBOUND 7. INTONER 8. GECKOES 15. ASSES 16. BOOTS 18. OPE 20. END 21. DESKTOP 22. BRAVADO 23. GARBAGE 25. ROUTINE 26. ABREAST 27. SYNCHRO 31. NEST 32. TEAM.

The prize for this month's crossword we have a an April Fool's delight, a veritable feast of goodies. Please send your entries to the address on top of the next page,

Break

Please send your rants, raves and competition entries to:

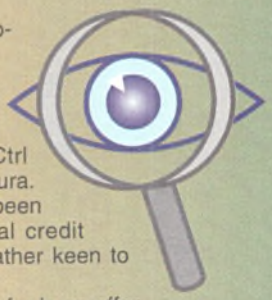
Ctrl/Break
EXE Magazine
50 Poland Street
London W1V 4AX

Techno-Oscar

One thing that the computer industry rarely seems to be accused of is a tendency towards 'glamour'. Far and few between are the tabloid stories of jet-setting software developers, aided and abetted by nasty cocaine habits, being hounded by screaming groupies and paparazzi. But this could all change soon if the Academy of Motion Picture Arts and Sciences has its way. For Dr Mike Boudry of the Computer Film Company has been awarded for his 'pioneering work' in the field of digital film input scanning. The ceremony was of course held in Beverley Hills (where else).

Ctrl Break very kindly offered to pack its bags quickly and scoop this story at source, but alas the funds were not forthcoming. Such short-sightedness. But we can still picture the scene; the limos, the press snapping furiously away. And of course the desperate scramble for that invite to the 'hottest' apres-awards party. No more 'geek' labelling for computer programmers from now on. What a joy to think that sycophancy has at last reached our quiet little pool of life.

Sticks and stones....



The curious tale of Kevin Mitnick and Tsutomu Shimomura was just breaking as Ctrl Break went to press last month. For those of you who have been asleep for the past two months, Kevin Mitnick was tracked down by 'Cyber-detective' (not Ctrl Break choice of phrase) Tsutomu Shimomura. Mitnick has, amongst many other things, been accused of accumulating 20,000 personal credit card numbers so the FBI were naturally rather keen to track him down.

Favourite amongst the press clippings so far is an offering from *The Guardian* which headlines Mitnick as the 'Nerd Dick Turpin of superhighway'. Mitnick has also become famed for memorising police badge numbers and wreaking his computer revenge later. So Ctrl Break can't help but ponder on the wisdom of labelling him with this very handy but, quite frankly, tired description.

Top marks for ingenuity, however, have to go to Mitnick's lawyers. These inspired lyricists claimed at Mitnick's last court appearance that his unlawful behaviour was simply the result of an addiction to computers. If only the Kray twins had thought of explaining to their jury that they were just exceptionally fond of hatchets.

SUBSCRIBERS CLUB

Have you taken advantage of the EXE Book Page yet? Turn over to check out this month's selection of good software-related reads. And they're all good value for money too, thanks to EXE's specially negotiated prices for readers.

The EXE Show

Taking place at the Royal Horticultural Halls in Westminster on 8th and 9th of June, this show promises to be HOT. Surprise, surprise, EXE will be there along with all the other key players in the software development industry. We'll be keeping you posted as we get nearer to the event, but for the moment take a look at the advert on page 46 for more details.

Free ticket worth £10

EXE readers can register now for a FREE ticket worth £10. So don't miss out on the chance to attend this event. Fax your name and address to EXE on 0171 287 0710 or write to Tim Macpherson at Centaur Communication, St Giles House, 50 Poland Street, London, W1V 4AX to guarantee your place.

Discount on Seminars: EXE readers qualify for a 25% discount on the seminars being held at The EXE Show. Ring Suzanne Chamberlain on 0171 287 5000 for details.

Access User Group

The Access User Group has just been launched. EXE readers can join at the special introductory rate of £58.16. Or for £142.28 a corporate membership of up to five named members. The group offers Access users a variety of benefits including the opportunity to work with Microsoft on the direction of Access products, a bimonthly newsletter, national seminars and a user group forum on Cix. Call Nicky Gordon on 01242 256549 to enrol, or for more information write to Access User Group, Stokesley House, 53 Prestbury Road, Cheltenham, Gloucestershire, GL52 2BY.

Don't forget to quote 'EXE' for the special rate.

Windows 95 Show Competition Winners

It was nice to see so many of you at the Windows 95 show. Here's the list of lucky people who won the prizes on offer:

Ian McAlamney, Reuters/Visual Basic
Jill M Bowen, Southern Electric PLC/Visual FoxPro
Anita Hancock, MTM Argus/Access, Access Developer Kit
Simon Hopper, Green Gables/Visual C++ 2.0 Professional

Prizes for the above were kindly donated by Microsoft.

Nick J Grant, Abbey National; Viky Spence, Five Flyte; Alan Jones, Troy Technical Design Ltd; R. Anderson, HFT all won Borland's dBASE for Windows, donated by QBS.

Win free CD-ROM Game

This month's game is Rise Of The Robots (The Directors Cut) the hot new interactive release from Time Warner. The game includes music by Queen's Brian May. Rise Of The Robots is tipped to be a revolution in combat games because of its unique system that adapts to and learns your style of play. It certainly looks good - the graphics are 3D and film quality. Send in your postcards right away but please remember that we don't draw the winner until the end of the month. First one out of the bag then will win.

EXE TOKEN No8 April

The Book Page

Something for Everyone

Since we launched The Book Page two months ago, many Subscribers Club members have taken advantage of big discounts we have been able to pass on to you. Since then, several more IT book publishers have been in touch and this month we welcome Prentice Hall from whom we have selected these four titles. Thanks to them too for supplying the Crossword Prize this issue.

Database Developer's Guide with Visual Basic 3
by Roger Jennings
1130 pages

Normal Price: £41.67
Price to You: £31.25

Bonus Disk

Expert techniques for building powerful and efficient database applications. This best-seller includes coverage of Access, dBASE, Paradox and Retrieve. Among the topics you will find creating front ends for client/server database systems, establishing security in multiuser databases, and developing custom controls. The disk includes complete VB, utility and other sample apps.

Object-Oriented Programming in C++
by Naba Barkakati
666 pages

Normal Price: £27.50
Price to You: £20.63

A complete tutorial for object-oriented programming. The first book to help experienced programmers learn OOP. It explores all the features of C++, including classes and virtual functions. Offers a platform independent approach and doubles as a convenient reference manual.

Win32 API - Desktop Reference
by James McCord
1490 pages

Normal Price: £46.30
Price to You: £31.25

Bonus CD-ROM

This doorstopper is the most comprehensive coverage of all Windows NT API functions. Learn the nuts and bolts for app development in the Windows NT environment. Reference information for each and every function, macro message, notification message, transaction and structure. Excellent cross-referencing. Examples of the most-used functions and messages. CD-ROM includes Windows NT demos and tools.

Developing PowerBuilder 4 Applications - 3rd Edition
by Bill Hatfield
1000 pages

Normal Price: £35.00
Price to You: £26.63

Bonus Disk

This is a comprehensive book on PowerBuilder by a certified PowerBuilder Instructor. Users receive detailed instruction through practical examples and clear advice. Uses an easy-to-understand style to teach PowerBuilder techniques thoroughly. Covers the database server and all back-end development issues. The disk includes all source code from the book and several powerful applications, including an on-line help app.

YOUR ORDER

Tick		Normal	Your Price	Copies	Total
☐	Database Developer's Guide with Visual Basic 3	~~£41.67~~	£31.25	___	£_____ (+ diskette: VAT included)
☐	Object Oriented Programming in C++	~~£27.50~~	£20.63	___	£_____
☐	Win 32 API - Desktop Reference	~~£46.30~~	£34.73	___	£_____ (+ CD-ROM: VAT included)
☐	Developing PowerBuilder 4 Apps	~~£35.50~~	£26.63	___	£_____ (+ diskette: VAT included)
☐	Heavy Metal OLE 2.0 Programming	~~£38.99~~	£31.20	___	£_____
☐	Unauthorised Windows 95 Developers Resource Kit	~~£38.99~~	£31.20	___	£_____
☐	Free $tuff From the Internet	~~£18.99~~	£15.20	___	£_____
☐	Oracle Performance Tuning	~~£25.95~~	£19.50	___	£_____
☐	TCP/IP Network Administration	~~£22.00~~	£16.50	___	£_____
☐	sendmail	~~£24.50~~	£18.40	___	£_____
☐	UNIX in a Nutshell	~~£ 7.50~~	£ 5.60	___	£_____

You can get the blurbs on this group of books in February or March issues of EXE.

Shipping @ £3.50 per order — £ 3.50

Grand Total — £

Name:_____
Address:_____
_____
_____
_____ Post Code:_____

Reader Number: ☐

PAYMENT OPTIONS

Cheques or official purchase orders only, please. Unfortunately, we are unable to accept credit card payment for The Book Page just yet - please bear with us for the moment on this one.

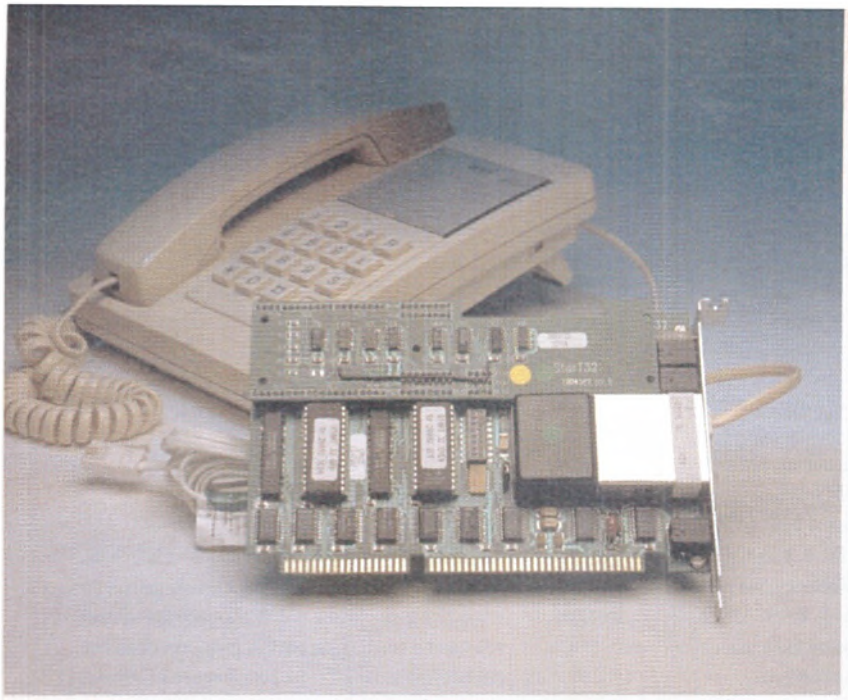
Please tick or fill the relevant boxes below

☐ I enclose a cheque for £_____
(payable to Centaur Communications Limited and drawn on a UK bank)

☐ I enclose a company purchase order. Please send me an invoice.

Simply fax 0171 437 1350 (with purchase order) or post (with cheque) this form with your order to this address:

EXE Book Page
Centaur Communications Limited
Freepost 39 (WD 1414/29)
St Giles House, 50 Poland Street, London W1E 6JZ



Voice Processing
Interactive Voice Response
Computer Telephone Integration

- WINDOWS DRIVER FOR WINDOWS V3.1
- SUPPORT FOR .WAV FILES
- VISUAL BASIC CUSTOM CONTROL
- TSR TOOL KIT FOR MS-DOS AND OS2 V2.x
- SPEECH RECOGNITION
- 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

ST32

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

A half length card for ISA Bus IBM compatible PCs

NEW!

Conformity to TAPI

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:

- CALLING LINE IDENTIFICATION (CLI)
- FAX AND DATA MODEM OPTION

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

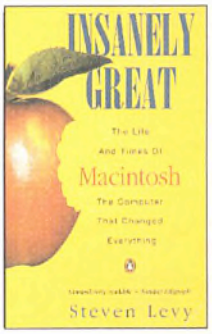
Staria Limited (Ask for Brian Atherton)
TEL: 0989 768 687 FAX: 0989 768 980
13 Market Place, Ross-on-Wye, Herefordshire HR9 5NU

APPROVED
S/2562/3/P/502756

CIRCLE NO. 072

Book review

Insanely Great reviewed by **Cliff Saran**



'I have essentially accessed another world... an ephemeral territory perched on the lip of math and firmament.' Steven Levy thus described his first, enlightening experience of a new computer that generations henceforth would fondly refer to as the 'Mac'. On that day, November 1983 in Cupertino, the genesis would spawn for a compassionate celebration of the greatest inspiration to the goal of computing for all. So great, in fact, that Steve Jobs described it as 'insanely great': an apt title for Steven Levy's biography of the Macintosh.

On reflection, nine years earlier in *Hackers*, the story of the true heroes of the computer revolution, Levy's final acknowledgement was to Steve Wozniak for creating the Apple II. Without it, he professed, a clean copy of the manuscript would have take an extra year on typewriter.

The story begins some 40 years earlier. In the wake of Nagasaki and Hiroshima, Vannevar Bush, then the vice president of MIT, wrote a far sighted article describing the culmination of visual, textual and audio knowledge onto what we now know as the 'desktop'. Doug Englebart, working at the Stanford research institute in 1963, fleshed out this vision and became the father of windows. He went on to invent that most famous of rodents.

In the 70s, while the nation 'discoed', to use Levy's own sardonic expression, PARC redefined computers. Palo Alto Research Center, the R&D arm of Xerox was the place to be. PARC, he says, was computing's very own Camelot. And it was here, as the legend goes, that Smalltalk, the Alto, the Star and the GUI were invented.

Wryly, Levy takes an almost comical view of the eventful day when Jobs and the team from Apple took a visit to PARC and saw the Alto for the first time. Compared to the Great Train Robbery, he considered Apple's was the slickest trick of all. 'They walked away with something more valuable than treasure... a paradigm.'

So, as the story goes, they set out to emulate what they saw at PARC with a 68000-based machine christened 'Lisa'. But Levy attempts to set the record straight on Apple's act of coping PARC. Bill Atkinson worked on the graphics routines that would later become QuickDraw. He needed an algorithm for clipping and ended up inventing it.

At a cost of $50 million to the company, Lisa was an expensive mistake for Apple. With a goal of cheap computing, Jef Raskin started a low key project codenamed 'Macintosh', away from the prying eyes of Steve Jobs. Levy portrays Jobs more as the driving force than the visionary 'father of the Mac' that he is often considered. He praises Raskin for providing the powerful vision of a computer whose legacy would be low cost, high utility and friendliness. But as the Lisa project wavered, it was clear that the Mac was the most interesting thing at Apple and 'Jobs took it over.'

Jobs insisted that the Mac be beautiful. In Levy's words, the design of the case would become as recognisable as the Volkswagen bug. Job's dream was a dream shared by all who worked on the Mac. They were revolutionaries, 'on a mission from God.' They were also artists but, as Jobs insisted, 'real artists ship.' He wanted the machine to ship with only 128 KB of memory. He almost didn't go with the new Sony 3 1/2" floppy drive and rejected a hard disk on the grounds that the resulting noise from the internal fan it would require would be 'inelegant'. He was also behind the lack of cursor keys on the keyboard. Far from creating an insanely great machine, the Mac was simply insane. In a sense, the Mac team were overexposed to what Levy called Steve Job's Reality Distortion Field.

Seriously under powered and overpriced, the Mac sold poorly when it was finally launched. Levy revealed 'that ad' in the third quarter of the 1984 Super bowl almost didn't run. Apple's board had canned it, but the ad agency couldn't resell the slot in time. The man who saved the Mac in the end was Paul Brainerd of Aldus which produced a new package called PageMaker. With PageMaker and Apple's new LaserWriter, the world of desktop publishing became Apple's oyster.

Verdict: Golden delicious.

Title:	*Insanely Great*
Pages:	*312*
Price:	*£7.99*
Author:	*Steven Levy*
Publisher:	*Penguin*
ISBN:	*0-14-024492-1*

Debugging The Development Process reviewed by **Edward Kenworthy**

Writing Solid Code, Maguire's first book, was the best book on practical software development that I had ever read. Regular readers will recall the glowing review it received. It was with some anticipation, therefore, that I awaited the arrival of his next, *Debugging The Development Process*. A book that promised to do for my project management skills what *Writing Solid Code* had done for my coding skills. It would, according to the back cover, show me how to energise software teams, why I should kick a star programmer off my team and, marvel upon marvel, how I could deliver on schedule without overwork! The book would be worth buying just for that last secret if for nothing else...

Debugging The Development Process has a wealth of helpful suggestions and ideas. Some of them - like using email to communicate progress - I'm not convinced about. I prefer the human touch, although I would agree with his arguments against the traditional methods of monitoring progress. Other ideas and arguments, even the most trivial and tiresome, are belaboured. Either Maguire was convinced his readers needed to have the ideas hammered home Chinese water torture style, because they're difficult or possibly important, or he was padding out a book with a short delivery schedule designed to take advantage of the success of its far more worthy sister. I'm of the more cynical inclination myself.

Overall I come down against recommending this book. It feels as if the ideas presented were written down as Maguire thought of them, almost as if his notes for a book were taken, tidied up a bit and then printed. Repetition is the order of the day and the book seems to have completely bypassed the editing process. There's a lot of dead wood that desperately needs pruning away.

Verdict: Not Recommended.

Title:	*Debugging the Development Process*
Pages:	*182*
Price:	*£21.95*
Author:	*Steve Maguire*
Publisher:	*Microsoft Press*
ISBN:	*1-55615-650-2*

The prerelease was second in the EXE Developers' Challenge.

Think what you could do with the real thing.

DELPHI COMES TO QBS

3D Graphics Tools	£115.00
3D Widgets	£85.00
Agility for VB	£149.00
†Aware/VBX	£110.00
†CodeBasic	£129.00
CodePrint Pro	£79.00
Crescent NetPak Prof	£129.00
†Crescent PDQComm	£95.00
†Crescent QuickPak Pro	£140.00
Crescent VBDD Datadict.	£129.00
Crescent Xref for VB	£79.00
Crystal Reports Pro	£295.00
Crystal Reports Server	£850.00
†Custom Control Factory	£40.00
DataDirect Dev Toolkit 2	£495.00
DataDirect MultiLink/2	£289.00
DataDirect ODBC Pack	£345.00
†Data Widgets	£90.00
Desaware CCF Cursors	£30.00
Desaware Cmn Dlg Toolkit	£30.00
†Designer Widgets	£90.00
Distinct TCP/IP SDK Vis.Ed.	£179.00
†First Impression	£199.00
†Formula One	£199.00
EXTools/VB Professional	£255.00
FXTools/VB Standard	£145.00
Gantt/VBX	£189.00
†Grid VBX	£70.00
†Graphics Server SDK	£199.00
†HighEdit SDK	£195.00
†ImageKnife VB	£165.00
†ImageKnife Pro	£245.00
†ImageStream	£199.00
Integra/VDB for VB Server	£499.00
Integra/VDB for VB Desktop	£199.00
Intelligent Paper	£125.00
†LeadTools/VB 4.00	£459.00
†List & Label (VB)	£275.00
†List & Label (VB, C/C++, Pas)	£389.00
M.4 VB	£155.00
†MediaKnife	£230.00
†MicroHelp Comms Library	£109.00
MicroHelp Compression Plus	£170.00
MicroHelp Fax Plus	£170.00
†MicroHelp Muscle	£119.00
MicroHelp Network	£79.00
MicroHelp Spellpro	£119.00
MicroHelp Uninstaller	£49.00
MicroHelp VBTools 4	£105.00
†MicroHelp VBXRef	£79.00
†MicroHelp Viewer	£79.00
Microsoft Visual Basic Pro	£225.00
Open Mail System (VBX)	£189.00
Rocket	£195.00
Schedule/VBX	£189.00
†Spread VBX	£165.00
†SpyWorks	£115.00
†Tab Pro	£70.00
†Tab VBX	£49.00
TMS Tools	£99.00
†ToolThings	£125.00
†TrueGrid Professional	£129.95
VBAssist	£135.00
VB-Cert	£99.00
†VBCompress	£95.00
VBDB Code/Form Gen. (Intro)	£99.00
†VB Language Manager	£130.00
†VB Language Manager Pro	£195.00
VB ProjectWorks Std 1 user	£149.00
VB ProjectWorks Pro 3 User	£225.00
VB ProjectWorks Pro Unl. user	£895.00
VB ProjectWorks Pro Extra User	£75.00
VBtrv	£185.00
VExe VB Consolidator	£89.00
†Version Stamper	£120.00
Versions/VB 1.1	£75.00
Visual Tools Suite	£275.00
†VisualWriter Pro	£199.00
†VSVBX	£40.00
†VSView	£89.00
†VT Speller	£109.00

†Maestro Pro for Visual Basic	£785.00
†Notes Netware Server	£435.00
†Notes OS/2 Client or Server	£435.00
†Notes Transporter (5 server)	£2375.00
†Notes Windows Client	£435.00

Borland Delphi Desktop	£179.00
Borland Delphi Client/Server	£call
Conversion Assistant	£call
Distinct TCP/IP SDK Std	£375.00
†Doc-To-Help (1 user)	£255.00
†First Impression	£199.00
†Formula One	£199.00
†Graphics Server SDK	£199.00
Greenleaf Commlib	£245.00
†InstallShield for Windows	£319.00
LeadTools Express 4.0	£1380.00

†LeadTools Pro 4.0	£765.00
LightLib Business Pro	£299.00
NeoImage API (16 bit)	£240.00
Open Mail System C, C++	£379.00
Orpheus	££135.00
MicroHelp Compression Plus	£170.00
Mobius Business Builder	£195.00
Mobius Drawkit	£95.00
Mobius WinG Sprite Kit	£95.00
Silverware Win Comms Kit	£219.00
Visual Tools Suite	£275.00

Cryptor (Standalone)	£99.00
FlagShip (FoxPro under UNIX)	£call
FoxFix for DOS and WIN	£149.00
Mac Rubel's Power Dev. Lib	£125.00
Microsoft Foxpro 2.6a Pro	£279.00
Microsoft Foxpro 2.6a Std	£79.00
MIX (Single User)	£95.00
NetLib Network Library	£189.00
Raidar debugger	£175.00
ReFox Decompiler	£295.00
†SilverFox Comms (Win/ DOS)	£249.00
SymScript	£145.00
The FEEL Editor	£85.00
TimeOut	£145.00
Visual FoxPro Standard	£call
Visual FoxPro Professional	£call

Adv. xBase Server SDK	£199.00
Adv. xBase Server (5 user)	£725.00
Adv. xBase Server (10 user)	£1195.00
Bandit Linkable Reporter	£199.00
†Blinker Linker	£179.00
CA Clipper 5.x	£299.00
CA Exospace	£95.00
Causeway 386 DOS Extender	£145.00
†Class(y)	£109.00
†Clip-4-Win	£195.00
†ClipSQL	£585.00
†Comix RDD	£99.00
dBest Printer Drivers	£195.00
†dGE Graphics	£189.00
†Escape Printer Library	£159.00
†Fast Text Search for Clipper	£70.00
†FAXual.lib	£79.00
Flexfile	£99.00
†FUNCky	£169.00
HighClass	£79.00
†MrDebug Clipper debugger	£99.00
NetLib Network Library	£189.00
†NovLib Library	£189.00
†R&R Report Writer/DOS	£195.00
†Rescue Decompiler	£259.00
SilverClip SPCS Comms	£249.00
Scrutiny	£109.00
†Summer '93 Code Optimiser	£159.00
†Telepathy Clipper Comms	£159.00
†T-BASE Graphics DOS or Win	£399.00
WBrowse for Clip-4-Win	£69.00
xCase	£299.00

Borland C++ v 4.5	£275.00
Borland PowerPack	£69.95
Borland Visual Solutions Pack	£59.00
†CodeBase or CodeBase++	£235.00
CodeBase multi-platform	£699.00
Distinct TCP/IP SDK Pro	£525.00
Distinct TCP/IP SDK Std	£375.00
Formula One 32 bit	£199.00
†Graphics Server SDK	£199.00
Greenleaf Comm++	£195.00
Greenleaf Database Library	£195.00
†High Edit SDK	£195.00
Integra/VDB for C++ (Desktop)	£199.00
Integra/VDB for C++ (Server)	£499.00
LeadTools 4.0 WIN 32 bit	£999.00
†LeadTools Professional 4.0	£765.00
List & Label (C/C++)	£389.00
Microsoft Visual C++ Prof v2.0	£295.00
MS VC++ Prof v2.0 (upgrade)	£149.00
NeoImage API (16 bit)	£240.00
NeoImage API (32 bit)	£290.00
VBtrv for C++	£275.00
Watcom C/C++ (CD)	£199.00
Zinc App Framework	£call

PowerBuilder Desktop 4	£489.00
Erwin for PB Enterprise	£2395.00
FUNCky for PowerBuilder	£175.00
†PowerFrame App. Fr. Library	£305.00
†PowerFrame App. Sec. Lib.	£1150.00
PowerFrame Navigator Object	£99.00
PowerFrame Object Analyser	£99.00
PowerFrame TabFolder	£99.00

Adv. xBase Server (Due in April)	£call
App:Convert2VO()	£145.00
App:HelpConstructor()	£145.00
App:VBX	£145.00
CA VO (Clipper user special)	£359.00
Class:VBX	£219.00
†Formula One	£199.00
FUNCky for VO	£229.00
†Graphics Server SDK	£199.00
Grumpfish Clip-4-VO	£call
†HighEdit	£195.00
†ImageKnife Pro	£245.00
LightLib Business Pro	£299.00
Looking at CA-VO (Book)	£45.00
†MediaKnife	£230.00
SilverWare Win Comms Kit	£219.00
Six/VO	£call
Visual Tools Developers' Suite	£275.00

Barcode Library (DOS)	£389.00
†Borland dBASE for Windows	£255.00
Borland ReportSmith PC	£69.95
Borland ReportSmith SQL	£195.00
†BugTrak (1 user)	£135.00
BugTrak (5 user)	£359.00
BugTrak (unlimited user)	£1259.00
†DemoQuick	£390.00
†Doc-To-Help (1 user)	£255.00
†ED for Windows v 3	£145.00
†ED for Windows +DOS	£199.00
†EZ-Install for DOS	£195.00
†InfoModeller DeskTop	£339.00
†InstallShield for Windows	£319.00
InstallShield for OS/2	£495.00
InstallShield for NT	£495.00
Microsoft Access	£275.00
Microsoft Access Dev. Toolkit	£289.00
Microsoft Office	£395.00
Microsoft Office Pro (+Access)	£520.00
MultiEdit Pro + Evolve	£155.00
OnCmd XBase for OS/2	£189.00
OnCmd XBase/OS/2 runtime kit	£295.00
OutReach VBX Enabler	£call
†PC-Install for Windows	£129.00
PowWow (5 user)	£295.00
†R&R Rep Wtr Win/xBase 6.0	£195.00
†R&R Rep Wtr Win/SQL 6.0	£295.00
†SOS Help! for Win Info Author	£195.00
†Versions for Windows	£195.00
WatCom SQL Developers	£260.00
WigWam Duo (Cix & CIS)	£89.00
WigWam Pro (Cix, CIS, 'Net...)	£179.00
WigWam Solo (Cix or CIS)	£59.00
†ZipitUp Software Protection	£99.95
ZipitUp Pro (WIN & CDs)	£call

ORDER BY WIRE!

You can now pick up detailed product information, and even place orders, via email. Product information will be sent back to you within a few minutes of you requesting it (subject to Internet traffic).

To use the service, send a message to info@qbss.co.uk (or INTERNET: info@qbss.co.uk from CompuServe). The message should contain the following lines in its body:

help
end

In return you will be sent detailed instructions describing how to use the service. Note that messages are dealt with automatically, and should only contain valid commands. The service is free.

†We have demos on our BBS for these products. Call 0181 747 1979, connect 2400 up to 28,800 baud, 8 data no parity 1 stop, to pick them up.

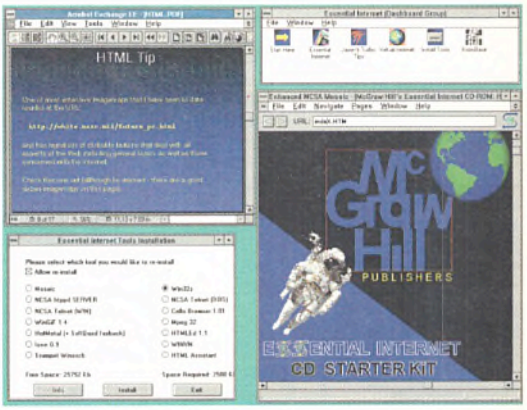
Call us today on 0181 994

QBS Software Limited
10 Barley Mow Passage, London W4 4PH
Ph: +44 181 994 4842, Fax: +44 181 994 3441
BBS: +44 181 747 1979, L/server email: info@qbss.co.uk
(Send email containing the word 'help' on a line by itself in the body of the message. Reply contains instructions for full Internet contact with QBS.)

*Prices, correct at time of going to press, are subject to change. All trademarks recognised.

CD-ROM review

Essential Internet Starter Kit - reviewed by **David Mery**



Coming from McGraw-Hill you would expect the Essential Internet Starter Kit to contain at least a book. There is none or should I say no *printed* book. The *Essential Internet Starter Kit* comes as a CD-ROM with an additional floppy disk. But amongst its 105 MB of data, two electronic books can be found. They are stored in Adobe Acrobat .PDF format. An Acrobat reader is installed during the setup process. At the beginning I couldn't find the registration number, but it is present and printed on the sealed envelope containing the CD-ROM. I find Acrobat quite a good reader to browse a book online, but a paper edition is more versatile. The main interest of having an electronic version is to perform searches. For that purpose the Verity search engine is also included.

The two books, *The Essential Internet Information Guide* and *The World-Wide Web, Mosaic and More*, are both written by Jason J Manger. They appear quite complete. However, it will take you a long time to read these books online. The first one is 673 page long while the second is 582 pages. I must admit that I didn't read them completely. On the other hand I really enjoyed the short *Turbo Tips,* a collection of tips, on HTML, Mosaic, Winsock, Httpd and Cello. I discovered some nice URLs and found some good tips on composing HTLM pages.

The only printed text that comes with the package is a one page offer from CityScape, and the CD-ROM cover disk which describes the installation process. One important point that needs to be emphasised is that this CD is for Windows. Internet can be accessed from any type of computer. Acrobat readers exist for DOS, OS/2 and Macintosh. But the CD-ROM contains only a Windows Acrobat reader and mainly Windows software. There is however, a DOS version of NCSA Telnet for DOS users.

On the software side, the CD-ROM does include all the *essential* software. By that I mean it does have most basic tools to access the Internet and publish Web information but it is by no mean exhaustive. So who should be interested by this CD-ROM? My impression is that it is most suitable for people that do not yet have an Internet connection and either are going to or just want to have a gist of what it's all about. What is HTML? What can be done with it? The CD will let them wander around the Web with most of the tools they need without the need to download these utilities for themselves. They save in time and money: these files are big. There is no need to download them when they are on CD-ROM already. *The Essential Internet Starter Kit* contains everything needed to create HTML pages, publish them and view them on a local machine or on a LAN.

Most of the software on the CD-ROM is shareware. So if you intend to use it for more than just an evaluation you'll have to register the ones you use. The price of £39.95 covers only the books, the Adobe license and the distribution of the tools. It doesn't cover their use. *Essential Internet Starter Kit* contains one stack, the Trumpet Winsock 2.0c and two Web browsers: Mosaic 1.00a-010 and Cello 1.01a. It doesn't *even* have NetScape, which is my favourite browser at present. To run the version of Mosaic provided, the 32 bit extensions of Windows must be installed. If you haven't already done it, they can be installed from the CD-ROM. From the version of Mosaic included it seems that tools present are up to date. But if you buy this CD in six months time then you should check that it has been updated.

For composers of HTML pages there's a choice between HoTMetaL 1.7, HTMLed 1.1c and HTML Assistant 1.4M. After authoring some pages, you can then compose your first Web service with NCSA htppd server V1.3Pre.

The other tools on the CD-ROM cover reading news (WinVN 0.92.6), telnet (NCSA Telnet Win Beta 3), displaying GIF files (WinGif 1.4) and playing MPEG movies (MpegPlay 1.61). The JPEG viewer LView is mentioned in one file but doesn't make it to the CD-ROM. What about ZIP, TIFF and QuickTime files or all the other weird formats you can encounter on the net? Nothing. It would have been a definite plus for such a CD-ROM to include all these hard to find tools that let you display, listen and convert the less standard formats.

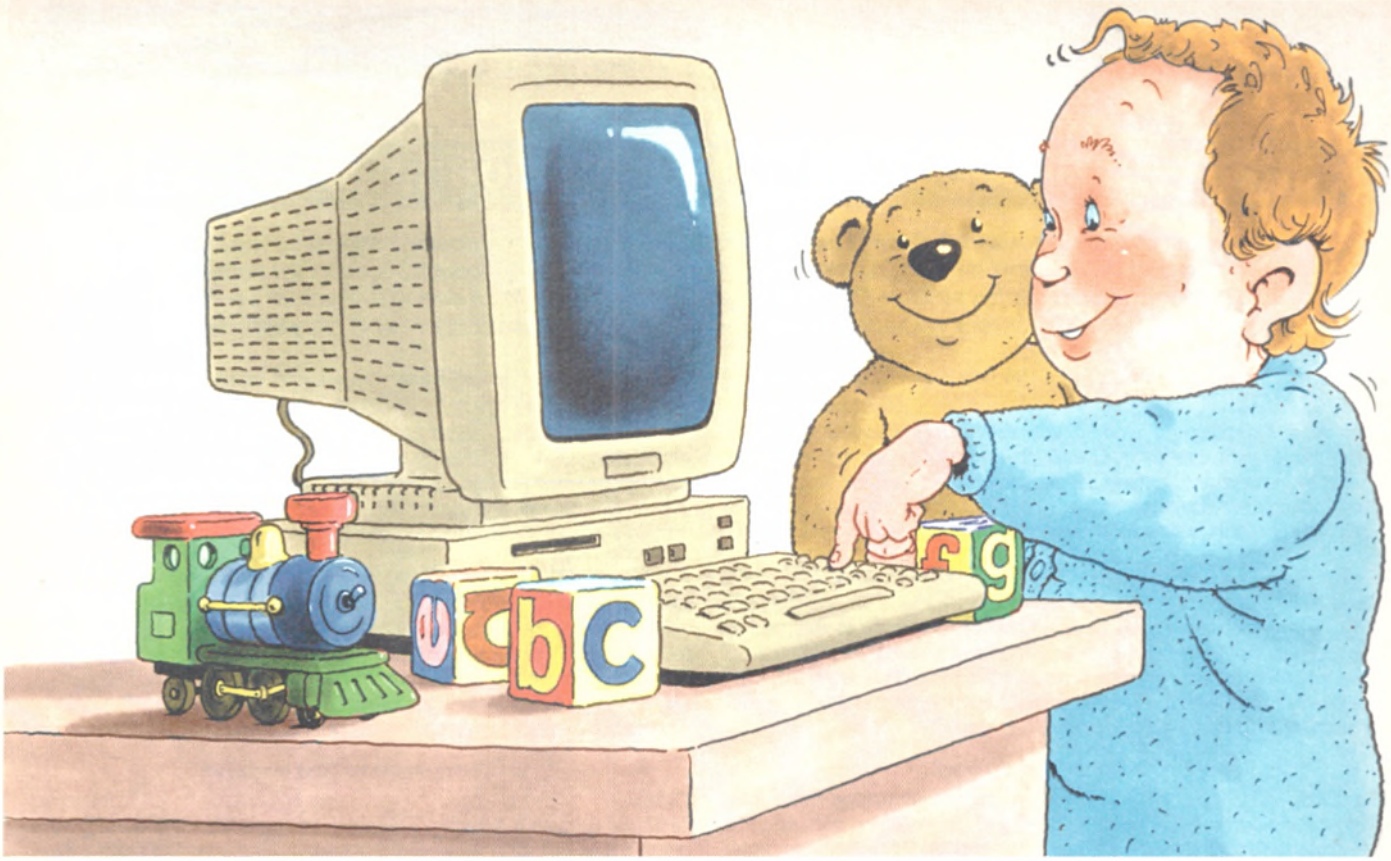
The weirdest thing about the *Essential Internet Starter Kit* is the presence of the Icon language on the CD. But what is its relationship with the Internet? I have no idea. Why would new users to the Internet suddenly want to develop in Icon? Not a clue. My only wild guess is that during the production stage of the CD, Icon was put there instead of LView. It look so unrealistic that I find it hard to believe. But that's the only explanation I could come up with anyway.

So you have everything you need to start browsing local HTML files. If you want to get online then you must get an account from an Internet provider. An offer from CityScape is included. It is presented as a free 80 hours trial, but you need to give your credit card details when you register. You have the option of cancelling after the free trial is up. It looks like McGraw-Hill made a deal with CityScape very late. A separate disk is provided to access CityScape. Even so, it includes Trumpet Winsock already present on the CD-ROM and WinWeb, yet another Web browser. The version of Winsock on the disk (1.0A) is much older than the one on the CD-ROM (2.0c).

According to McGraw-Hill's press office, so far there's no intention to update the tools present. Also this Windows version is the only one planned. I was disappointed. I thought it could be useful to both regular onliners and newcomers. But if you already have a Web browser and a TCP/IP stack, don't bother.

Verdict: If you're using Windows, this CD can help you get started on the net but it should have been more complete.

Title:	*Essential Internet Starter Kit*
Content:	*1 CD + 1 floppy disk*
Price:	*£39.95*
Author:	*Jason J Manger*
Publisher:	*McGraw-Hill*
ISBN:	*0-07-709142-6*



Software protection is childsplay with DESlock.

Millions of pounds is lost every year through the mis-use of software and even piracy. Policing of the industry is difficult, and sometimes ineffective or non-existent abroad. Therefore the emphasis falls on you to protect your own investment. You need to create and distribute more secure software products.

Thankfully now the whole process of software protection is made easy with the latest range of security products from Data Encryption Systems, the UK's leading software protection specialists.

DESlock and DESkey are the easy way to implement software protection, combining to create an impenetrable level of protection against illegal use of in-house and third party software. And it's so much quicker.

Until the advent of DESlock, you would have spent hours implementing software protection. But now it's as easy as typing a one line command or using the menu driven set-up programme. It will take you seconds not hours. And you don't even need to access any source or .OBJ files.

It really is childsplay.

DESlock is designed for DOS and Windows software and can also protect networked applications using only one DESkey.

All DESkeys are transparent in use and therefore do not interfere with the normal operation of the computer and its peripherals. This leaves your customer free to legitimately use your protected software any way they choose.

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.

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

Data Encryption Systems Limited

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

Telephone 0823 352357
Fax 0823 352358

CIRCLE NO. 074

Software developers... Isn't it time you found G.I.S.???

If you haven't already discovered, there is a truly exciting new enabling technology sweeping the IT industry. It is called G.I.S., short for Geographic Information Systems and it incorporates current software engineering philosophy, such as OO, RDBMS and WYSIWYG within a data visualisation context. Check out these six current vacancies to see how you could become part of the G.I.S. phenomenon.

GIS Applications Programmers (No GIS experience required)

To £20K **Central London**

This is a fun place to work if you are a highly motivated young programmer with fluency in C and Visual Basic under MS-Windows. The client is a young, hugely successful MapInfo partner able to offer challenging new development work for an extremely wide range of commercial clients, many of whom are themselves new to GIS.

Software Engineers (All levels)

£15K to £30K **Wiltshire**

No GIS experience needed here, either. Demonstrate to this rapidly expanding systems house that you have sound experience of applications programming in embedded C and/or C++, using Yourdon design discipline within a real-time UNIX environment and your career is on track for a career in real-time GIS-based C3i.

OO Systems Developer

To £30K **Cambridge or London**

With or without experience of GIS, this IT consultancy and systems integrator will consider you if you have developed OO systems under UNIX, ideally within a Utility such as Electricity, Water or Gas. Your skills profile must include fluency in C++, Smalltalk or Smallworld Magik programming.

Technical Programmer (Oracle and Windows skills)

To £30K plus car **Thames Valley**

If you enjoy 'leading edge' challenge, this client offers challenge in abundance. The task is to build a new generation GIS which incorporates multi-media. OO and Oracle relational databases, within the latest windowing environment. To qualify you must be fluent in C and C++ in both a UNIX and PC environment.

Software Developer (GIS Core functionality)

£25K plus car **Central London**

You must be highly proficient in C++ programming within current MS-Windows technology to qualify for this newly created position. Some experience of cartography, spatial data or GIS is beneficial but not essential. Plans include an imminent move to Windows 95 and NT, placing the product at the 'cutting edge' of desktop GIS.

Software Development Manager

£30K **Reading**

The brief calls for a 'hands-on' working manager who can manage and direct the efforts of a small team of programmers whilst handling the more intricate elements of a new GIS product development project. Your CV must indicate fluency in C and C++, experience of Windows development tools and GIS.

Concurrent Appointments International

To apply for one of the above vacancies contact Alan Carnell on (01582) 712976 or mail him your CV in the strictest confidence to the address shown. All applications will be acknowledged.

27 Field Close, Harpenden, Hertfordshire A15 1EP. UK National Tel: (01852) 712976 Fax: (01582) 764858 International Tel: +44 582 712976 Fax: +44 582 764858

NOVELL

The World of Pervasive Computing

Today there are 80 million users of Novell services and applications, such as NetWare, UNIX, WordPerfect and Quattro Pro. By the end of the decade Novell will reach out to more than one billion users. Novell and its partners are creating a world of pervasive computing that connects people with other people and the information they need, enabling them to act on it anytime, anyplace.

Software Engineers

Based Hungerford • Excellent Benefits Package Including Car

Due to continued expansion we require Software Engineers to work on Advanced Access Application and Network System Software. This is a rare opportunity to be involved in the development of leading edge products through the complete product life cycle for shipment into the global market.

The successful candidate will be a competent 'C' or 'C++' programmer and a user of structured programming techniques. It would be desirable if this person has a good knowledge of networking as this role will have a strong bias towards the development of networking system software.

If you are interested in this exciting opportunity based at our European Development Centre in Hungerford, where salary will be commensurate with experience, then please send your CV to the Novell Team at CTA International Search & Selection, Staverton House, 3-5 Easthampstead Road, Wokingham, Berkshire RG11 2EH, quoting reference NUK-148

PROGRAMMERS WANTED

Help to write the future of Interactive TV!

We have a number of vacancies for programmers to augment a team working on the development of an interactive TV system. Applicants should be very experienced in one or more of the following areas:

- 68000
- H8
- 'C'
- 'C++'
- 'Visual C' or 'Visual C++ for Windows'

Formal qualifications are less important to us than genuine ability and enthusiasm. Salary will be by negotiation. The successful candidates will be working in our client's offices in Chiswick.

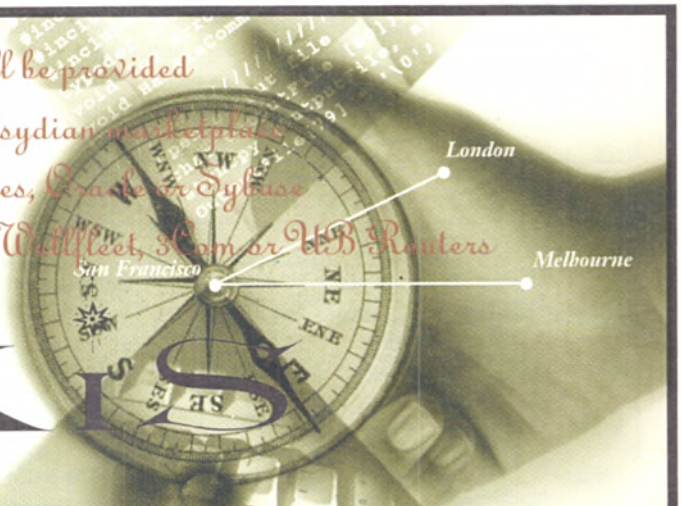
Please reply with CV to:

IR

David Levy,
Intelligent Research Ltd,
89, Constantine Road,
London NW3 2LP

email: DavidL@intrsrch.demon.co.uk

SAP R2/R3 *R/2 to R/3 training will be provided*
SYNON/2E *anxious to work in the Obsydian marketplace*
UNIX, C, RDBMS *using either Ingres, Oracle or Sybase*
INTERNETWORKING *using Cisco, Wellfleet, 3Com or NetBuilder Routers*



AXIS USA

Axis Consulting International, Inc. is one of the leading consulting firms in north america concentrating on projects in the above technologies. We are highly respected in the industry for our business ethics, professionalism and in-depth knowledge in obtaining USA work permits (H1 visas). Our company is currently seeking experienced professionals committed to excellence and eager to work on leading edge projects within the United States. We offer contract and permanent positions.

Axis offers excellent hourly rates for contract positions and outstanding salaries and benefits for permanent employees, including tax-free allowance. We handle all visa formalities.

Mail or fax your resume or call one of the divisional managers direct in our San Francisco office:

Linda Jackson SAP Division
Chris Sodergren Internetworking Division
Mike Shive Synon/CASE Division
Steve Re Client/Server Division

One Maritime Plaza, 24th Floor, San Francisco, California 94111 Tel 1 (415) 434 2947 Fax 1 (415) 434 2951
Glen House, 200 Tottenham Court Road, London W1P 9LA Tel (071) 323 4770 Fax (071) 916 1520

JOB
C++/MS WINDOWS

LOCATION	SALARY
SUSSEX	TO £25K

Our client has a niche in the financial market designing and manufacturing digital voice recorders. They are searching for a highly professional Software Engineer skilled in Borland C++/MS Windows development, with particular emphasis on OO design skills. Experience of a client-server environment and database design would be of interest. In the last few years our client has experienced sustained growth and is seeking the right candidate to progress with the company and become a key player within the organisation.

REF: EXE/10

JOB
BORLAND C++ PROGRAMMER

LOCATION	SALARY
W. LONDON	TO £23K + BENS

This is an excellent opportunity for a talented Borland C++ Programmer to work on the new development of finance-based systems. Candidates should have upwards of 18 months experience using Borland C++ and OWL (preferably v2). Experience of technical design and of the implementation of at least one major project using Borland tools would be advantageous. Any knowledge of Borland Database Engine/Framework, Windows comms API or TCP/IP would also be of interest. An extremely competitive salary and benefits package is on offer.

REF: EXE/11

JOB
VISUAL C++/MULTIMEDIA

LOCATION	SALARY
CAMBS	£18K-£27K

This CD-ROM market leader requires additional Software Engineers to work on a variety of CD-ROM titles. Visual C++/MFC Developers of all levels are required. There is also a vacancy for a 'C'/DOS Programmer to cross-train to Visual C++ under Windows. The Apple Macintosh team are also recruiting at present. Any experience developing CD-ROM products or graphics software will be of interest, but is not essential. All successful applicants will have the opportunity to work with, and where necessary trained to use, the latest Multimedia technology within an established and successful CD-ROM company.

REF: EXE/12

JOB
VISUAL C++ PROGRAMMERS

LOCATION	SALARY
YORKSHIRE	TO £25K

A leading developer of communications and networking solutions is currently looking for two Programmers with Visual C++ and MFC skills. Applicants should be educated to degree level and must have at least one years postgraduate experience in 'C' and C++. Any knowledge of communications or networking protocols would be useful, as would any exposure to Unix. Candidates must be able to demonstrate an ability to keep learning the latest technologies. Excellent career prospects and challenging development projects are on offer.

REF: EXE/13

JOB
LOW-LEVEL WINDOWS

LOCATION	SALARY
N. HERTS	£17K-£25K

This young, fast-growing company with an international client-base is currently seeking an additional Software Engineer. Suitable applicants will have at least twelve months experience programming in 'C' or C++ in an MS-Windows environment. Solid Assembler along with experience of device drivers (particularly Windows printer drivers) are also sought for this exciting new position. The successful candidate will work on new product development in a dynamic environment using the latest Windows technology.

REF: EXE/14

JOB
REAL-TIME 'C' SOFTWARE ENG

LOCATION	SALARY
CAMBS	£20K-£30K

Our client is searching for highly professional and committed Software Engineers with a minimum of two years experience of developing embedded 'C' software under Unix. Experience of C++ and Object Oriented techniques would be an advantage. The ideal candidate will have some telecommunications experience across a split site. Experience of working in a large, well structured development environment would also be of interest. However, the most important factors are a sense of humour, team spirit and a strong commitment to the production of quality systems.

REF: EXE/15

CONTRACT VACANCIES - UK WIDE

Herts	**Visual C++ Analyst/Programmers**	3 months
London	**Windows/Visual Basic Programmers**	6 months
S.London	**Visual Basic Anal/Progs**	6 months
Berks	**Visual Basic Programmer**	4 months
London	**VC++/SDK Multimedia Developers**	3 months
Cambs	**VB/OLE Developers**	3 months
Surrey	**Windows/Financial Appls. All levels**	6 months
W. London	**Windows SDK/'C'Programmers**	5 months
City	**Windows SDK/'C'Developers**	4 months

REF: EXE/16

City	**Mac/Windows Text Handling Programmers**	6 months
City	**Text Processing**	6 months
Cambs	**Text Handling Processing**	4 months
City	**Apple Mac Multimedia Developers**	6 months
Cambs	**Apple Mac CD-ROM Developers**	6 months
Cambs	**Multimedia/CD-ROM Developer**	4 months
London	**Sybase/SQL Server Software Engineer**	4 months
Surrey	**Visual C++/MFC Finance Developer**	6 months
London	**Testing System Tester**	3 months

REF: EXE/17

London	**Visual C++/MFC Developers**	6 months
Surrey	**Visual C++/MFC Programmers**	6 months
Oxford	**Visual C++/MFC Programmers**	3 months
City	**Visual C++/SDK Multimedia Developers**	3 months
W. London	**'C'/C++ Progs x 2**	6 months
City	**C++/Banking Programmers**	6 months
London	**Oracle, SQL, Forms Developers**	3+ months
London	**Windows/C++ Programmers**	6 months
City	**Windows NT Senior Programmer**	3 months

REF: EXE/18

Logistix Recruitment Limited
Lamb House, Church Street
Chiswick Mall, London W4 2PD
Tel: 0181-742 3060
Fax: 0181-742 3061
email: logistix@atlas.co.uk

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/email a CV to us and we will contact you at a convenient time.

Logistix

the soft corporation

Specialists in Software Development Staff Recruitment

OOD/OOP, C, C++, VISUAL C++ — **ALL LEVELS**

As the market for Object Oriented skills gathers pace we have a number of clients designing systems in diverse application areas including: **Multi-media, DTP, Telephony, LANs, Electronic publishing, On-line information Feeds, Finance and Banking** in both a UNIX and DOS environment.

Positions available vary from traditional Programmer/Software Engineer and Analyst/Programmers to Designers/Senior Software Engineers in the overall strategic direction for end-user organisations.

£17-£35K + benefits — **REF: SC/01/EXE**

WINDOWS OR X-WINDOWS/BANKING — **ALL LEVELS**

Three 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), MFC, 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/02/EXE**

C AND C++ PROGRAMMERS — **ANALYST PROGRAMMERS**

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, 4GLs and Object Orientated Design. A good opportunity for a second career move.

£17-£25K + Banking benefits — **REF: SC/03/EXE**

CAMBRIDGE - MANY, MANY EXCITING OPPORTUNITIES

A wide variety of specialist, leading edge IT companies in areas as diverse as: **ROBOTICS, TELECOMMUNICATIONS, MULTI-MEDIA, GIS, BUSINESS MODELLING, FINANCIAL/TREASURY, EMBEDDED SYSTEMS** and **SOFTWARE/GUI RESEARCH/ MANUFACTURING** require high calibre software development staff at junior and senior levels. Technical skills required include: **C, C++, VISUAL C++, VISUAL BASIC, X-WINDOWS/MOTIF, GUI's, NT, TCP/IP/X25/X4000, PROGRESS, SAP, Relational Databases, INTERNET CONNECTIONS** and **ATM** (Communications not ATM machines). — **REF:/04/EXE**

INGRES/ORACLE/SYBASE/GUPTA/OOD AND OOP — **ALL LEVELS**

Additional experience of: SQL, Forms, C and C++ required. We currently have client companies including Management Consultancies, Systems Houses, Systems Vendors, Bank and Finance clients looking for candidates with: Relaticnal Database design, Database tuning, Systems Administration, DBAs, Pre/Post Sales and solid programming knowledge and expertise. Please call to discuss your particular requirements.

£18-£40K + benefits — **REF: SC/05/EXE**

C/C++/VISUAL BASIC - UNIX OR 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 incompetitive 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-£35K + Benefits — **REF: SC/06/EXE**

UNIX/VMS/MS WINDOWS/NT MFC/C/C++ — **ALL LEVELS**

A degree in computer or natural science, two years solid C/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 programmer 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-£28K — **REF: SC/07/EXE**

LONDON/HOME COUNTIES WINDOWS SDK/NT DEVELOPMENTS

Senior Development Engineers — **To £30K + benefits**

Analyst Programmers — **To £27K + benefits**

Strong programming skills in C or C++ and Windows NT are pre-requisites for these positions. Experience in some of the following areas is also required: MS-DOS 5.0, MS Windows 3.1, Windows NT, Windows SDK, MS C 7.0, MFC, Visual Basic, Visual C++ and Microsoft NT. Also desirable are Windows XVT libraries or networking skills. — **REF: SC/08/EXE**

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, C++, MFC, Windows (SDK, NT or X-Windows and Toolkits), Networking and Communications with companies offering technolgy based careers and management responsibility.

COMMERCIAL DEVELOPMENT: Using technical based skills already developed, but offering opportunites 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-£25K + benefits — **REF: SC/09/EXE**

VISUAL BASIC SKILLS MUCH IN DEMAND - PLEASE CALL TO DISCUSS — **REF: SC/10/EXE**

LEEDS - LOW LEVEL C++ WINDOWS COMMS DEV ALL LEVELS — **REF: SC/11/EXE**

LONDON COMMS SPEC X25, X400 £40-60K — **REF: SC/12/EXE**

C, C++/MFC - Countrywide — **REF: SC/13/EXE**

the soft corporation

Third Floor, 7-15 Rosebery Ave, London EC1R 4RP
Tel: 0171 833 2772 Fax: 0171 833 2774
email: jmcb@softcorp.demon.co.uk

NEW DEVELOPERS REGISTER

From April 1995 ASH associates will initiate a New register for dedicated software developers who seek a career path driven by technology rather than management.

We hear continually from companies seeking highly experienced designers at mid and senior level to design and code software. They offer realistic salaries and long term job security, some positions are targeted towards freelance designers.

So, if you have a minimum of 3 yrs software design experience and looking for your next challenge call us, discuss your needs and we will do our best to find the right position for you.

Current skills in demand include C/C++, Turbo Pascal, Windows-SDK, Assemblers 68K, 80X86 and Unix. Applications include Real Time Control Systems, Graphics, Multimedia, Medical Electronics, Comms and Defence.

Call James Hunt or Ron Cook on:

(01425) 475480

Fax (01425) 480807 or write:
ASH associates, First Floor,
39 - 41 High Street, Ringwood, Hants BH24 1UF.
Mobile (0831) 330305.

In and Around WEST YORKSHIRE

We have clients currently seeking the following:

UNIX/SOLARIS & VMS Technical support Manager	**£25,000 + car**
Visual BASIC Programmers (2 req'd) for progressive UNIX/MS Windows client server environment	**£20,000**
Developer/Support for UNIX/SQL based systems in client server environment (INGRES useful)	**£20,000**
Junior 'C' Programmer with PC skills (includes 4GL training)	**£12,000**
4GL experienced Programmer for Visual BASIC, OMNIS 7 and 'C++' environment	**£14,000**
'C'/Assembler Programmer for serial comms	**£13,000**
UNIFACE 4GL Programmer and team leader	**£14-£22,000**
'C' Developer for SNA implementation in a LAN environment, working with PC, Windows, NT, Chicago and Netware	**£30,000**
'C' Programmers for challenging low level environment	**£20-£30,000**
'C' experienced Programmer to enter a Visual 'C++' development environment	**£17,000**

For your next career move around West Yorkshire telephone **Vincent Atherton** on **Leeds (0113) 250 4560** or write to:

AIREDALE RECRUITMENT

Realtex Hs., Micklefield Lane, Rawdon, Leeds. LS19 6AX

Airedale Recruitment

ADVERTISERS INDEX

ADVERTISER	PRODUCT/SEVICE	CIRCLE	PAGE
Aladdin	Security Dongles	046	22
Bits per Second	Graphic Tools	048	26
Borland	Development Tools	037	5
B-Trieve	Programming Tools	042	16
Citadel	Comms Library	041	12
Contemporary I	Development Tools	039	9
Contemporary II	Programming Tools	051	32
DES	Software Protection	074	83
Elverex	Development Tools	053	36
EXE Show	Development Exhibition	058	46
Full Moon Software	Object Master	054	39
Grey Matter	Programming Tools	036	2
Hitex	Embedded Systems	068	66
Hypersoft Europe	Programming Tools	066	69
Intasoft	Allchange	069	70
Intech Ltd	Communications	075	OBC
Intersolv I	Development Tools	040	11
Intersolv II	Development Tools	049	28
Magic Software	4GLs Development	045	21

ADVERTISER	PRODUCT/SEVICE	CIRCLE	PAGE
Magnifeye	Software Protection Device	047	24
NAG Ltd	Development Tools	043	18
Nu-Mega	Programming Tools	062	58
Personal Workstations	Development Tools	035	IFC
Popkin Software	Systems Architect	044	19
Powersoft Europe Ltd	PowerBuilder	075	IBC
QBS I	Development Tools	070	73
QBS II	Development Tools	073	81
Rainbow Technologies	Security Products	050	30
Readmar	Version Control	052	34
Ridgeway Press	Computer Manuals	065	66
Sequiter Ltd	CodeServer	060	50
Software Security	Security Dongles	061	56
Softwerk	Development Tools	071	75
Staria	Telephony	072	79
System Science I	Development Tools	038	7
System Science II	Development Tools	059	49
Visual Numerics	Productivity Tools	063	61
Zinc	GUI Library	055	40

Whatever NEXT

The Software Developers' Magazine

Rave on Graphics in the merry month of May

- Games writing in OS/2
- Visual Fox reviewed
- Manipulate those pixels
- Best Unix books round-up
- Bullet proof code
- The day after phone day...
- Abusing friendship
- Patent GIF grief

EXE: The Software Developers' Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Centaur Communications Ltd, St Giles House, 50 Poland Street, London W1V 4AX.

EXE Advertising/Editorial/Production: 0171 287 5000 **Advertiding email: stevenm@dotexe.demon.co.uk**
Subscriptions 0171 439 4222 (Amina x2252, Fiona x2212) **Facsimile:** 0171 437 1350 ISSN: 0268-6872
Subscriptions. EXE. is available only by subscription at £35 per annum (12 issues) in the UK: see subs card within this issue. The magazine is published around the 1st of the month. To subscribe or if you have a subscription query, please call 0171 439 4222 or write to The Subscriptions Manager, **EXE,** (address above). We can invoice your company if an official company order is provided. Back issues are available at £3.50 each
'A Subscription implies that this journal will be sent to the subscriber until one of the three expires' (AG Macdonell.)
Editorial. Address editorial enquiries and comments to The Editor, EXE, (address above) or email 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 issue.

Editor: Cliff Saran
Features Editor: David Mery

Production Editor: Mark English
Production Manager: Kate Adams

Display Advertising Manager: Marc Warren
Recruitment Advertising Executive: Marc Green

Display Sales Executive: Steven Miles

Marketing and Promotions: Suzanne Chamberlain
Publisher: Sandra Inniss-Palmer

Office Manager: Jacqui Ramrayka
Subscriptions: Fiona Mitchelson

Front Cover Picture: Tess Roberts
Printer: St Ives (Roche) Ltd.

Repro: Clerkenwell Graphics
Typesetting: Ebony

PowerBuilder Desktop

There's no easier, more powerful way to get to Client/Server

COMPETITIVE UPGRADE

NOW AVAILABLE

CALL 0171 833 1022

AT SPECIAL PRICES

PowerBuilder Desktop for Windows is the most powerful application development environment available for individual developers. Combining Client/Server architecture with powerful programming tools, PowerBuilder Desktop lets you easily create robust graphical Windows applications that access data from most popular desktop databases. That's not new, but Version 4.0 is!

Features, Features, Features Full support for MDI, drag and drop, DDE and OLE 2.0 containers and automation, configurable toolbars and a brand new addition - DataPipeline.

Data Migration: This new feature allows you to copy tables and data from your existing desktop database to a relational solution such as Watcom™. Save the pipeline as an object and you can open and execute it in your application.

Built-In single-user 16 or 32 bit Watcom™ Database Watcom™ SQL relational database provides stored procedures, triggers and other advanced server database features, allowing deployment of single-user Windows applications royalty free.

Great Add-ons for Desktop Developers The Advanced Developer Toolkit - new with Version 4.0. Includes a cross reference tool, stored procedure update facility, Install Disk Builder Utility, Application Library, NetWare Library, Pen Library and DataWindow SQL verifier.

FUNCky Library for PowerBuilder Version 4.0 - access over 500 functions.

PowerBuilder Version 4.0 Library for Lotus Notes - integrate Notes data.

PowerBuilder Desktop Version 4.0 reaches new heights of client/server development capability. See it and believe it.

For further information please call ***System Science on Tel: 0171 833 1022.***

Powersoft™

Building on the power of people

CIRCLE NO. 076

Features of *DIALAFILE*

- NO MASTER/NO SLAVE
- API AVAILABLE
- WORKS OVER RS232
- FULL SECURITY
- NO JOB LOGGING
- NO SCRIPT FILES
- UNATTENDED POLLING OF REMOTE SITES

intech

COMMUNICATIONS SPECIALISTS

fully programmatic automatic polling software

THE DIALAFILE SOFTWARE ALLOWS A CENTRAL COMPUTER TO POLL ANY NUMBER OF REMOTE SITES OVERNIGHT USING TELEPHONE LINES AND HAYES COMPATIBLE MODEMS. PROGRAMMATIC TRANSFER OF ANY TYPE OF FILE IS POSSIBLE. THE SOFTWARE HAS THE ABILITY TO KICK OFF JOBS ON RECEIPT OF FILES. FULL ERROR CHECKING & LOGGING OF ALL JOBS IS PROVIDED. DIALAFILE CAN HANDLE MANY STREAMS AND MODEMS.

- *FILE TRANSFER FOR PCs HP3000 UNIX VAX AS400*

CIRCLE NO. 075

for further information please contact Jackie Clough on 01204 699660