November 1995

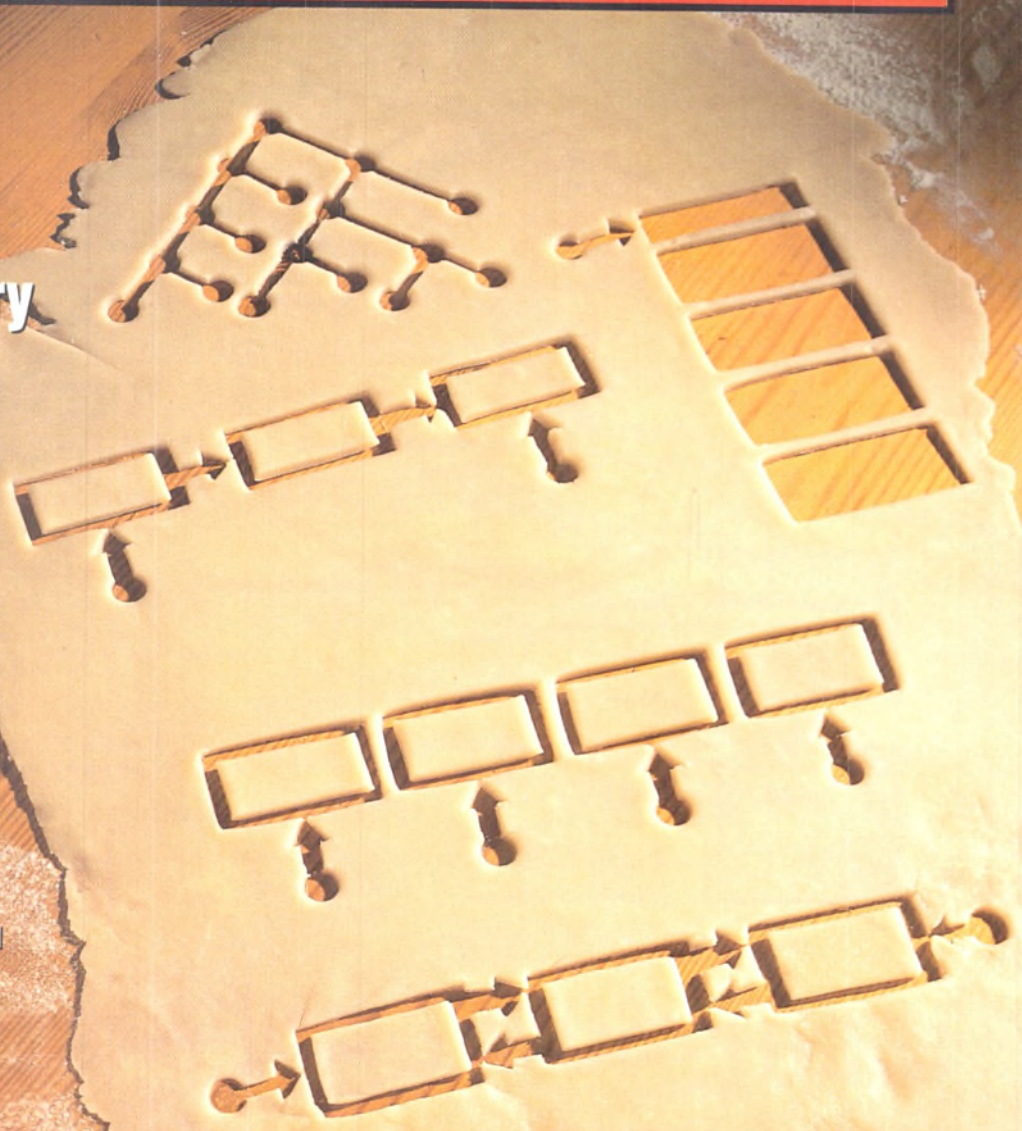
EXE

£3.20

Subscribers Club
see page 70

The Software Developers' Magazine

STL
A template library for ANSI C++



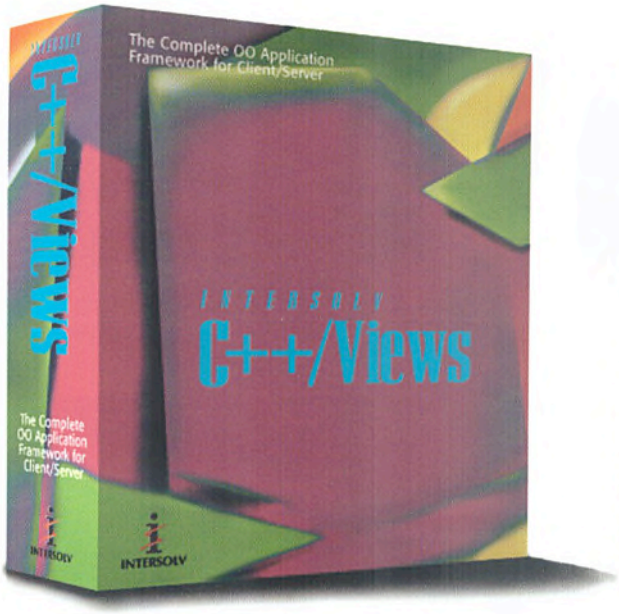
An old algorithm for a new language

Discover Linux

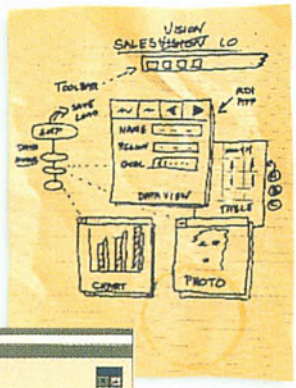
Jules' superlative Windows 95

Dynamic arrays in Delphi

9 770268 687008 11



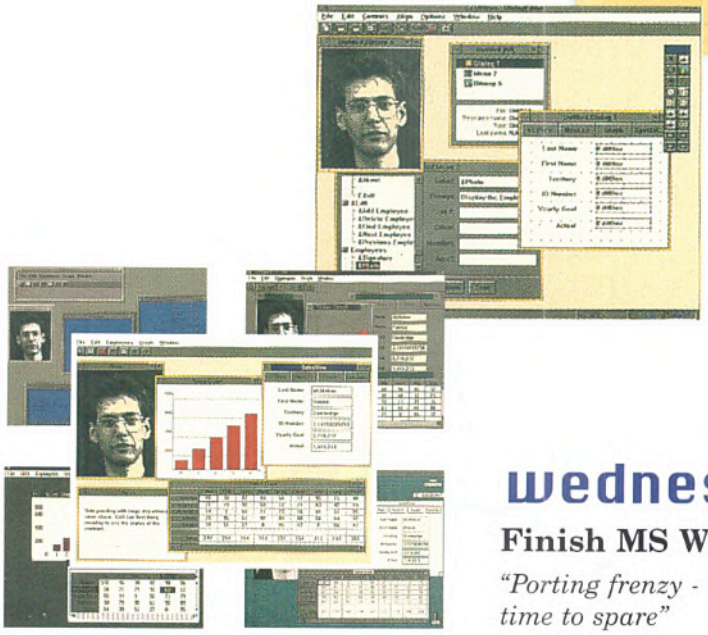
When the Deadline Demands Action...



monday....

The assignment

"OH NO! I agreed to get this application built by Wednesday... on Windows, Motif and OS/2."



tuesday....

Use C++/Views visual interface builder & C++/Browse

"I'm already ahead of schedule! I've got my dialogs laid out - I just have to finish the menus. Then, I'll use the class browser to create my classes and attach them to the dialogs and menus."

wednesday....

Finish MS Windows version

"Porting frenzy - hit the deadline with time to spare"

The complete OO Application Framework

...INTERSOLV C++/Views Delivers!

Need to develop client/server applications faster? INTERSOLV C++/Views offers an object-oriented application framework that helps your teams quickly deliver, deploy and maintain client/server applications.

C++/Views provides state-of-the-art class browser and GUI painting capabilities. This powerful combination lets you build reusable components and develop high-performance applications quickly.

Jump-start your C++ Development

INTERSOLV C++/Views provides a comprehensive solution that combines a rich, extensible class library with an intuitive visual development environment. The C++/Views class library holds over 110 object classes for UI design, data management, event processing and more. You can adapt any class to meet your needs.

Make Your OO Applications Fully Portable

With C++/Views, you can develop mission-critical applications on virtually any platform, for any platform. You can fully exploit each native environment - C++/Views portable resource format gives your applications a consistent look and feel across Windows, NT, OS/2 and OSF/Motif. Use C++/Views and you'll discover how quick and easy OO development can be.

For more information call 01727 812812 or fax 01727 869804

INTERSOLV

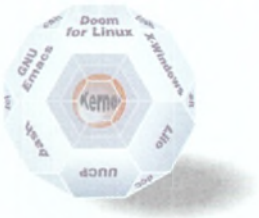
EXE
The Software Developers' Magazine

EXplodE
http://www.exe.co.uk

COMMENT

NEWS

REVIEWS



FEATURES

TECHNIQUES

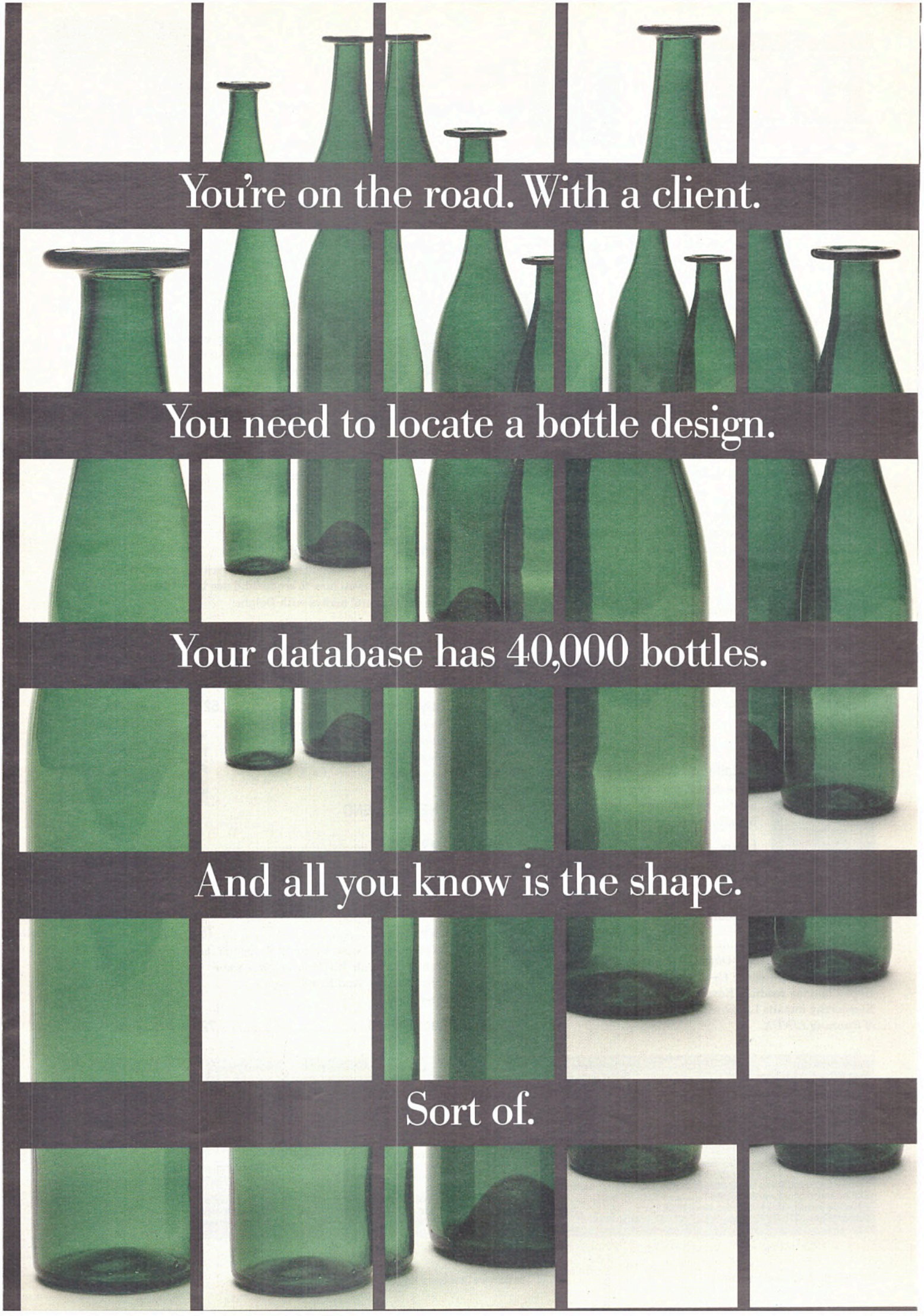
THE BACK END

Editor: David Mery
Staff Writer: Roland Perera
Contributing Editor: Will Watts
Editor - EXplodE: Cliff Saran
Production Editor: Mark English
Production Manager: Kate Adams
Publisher: Sandra Inniss-Palmer
Printer: St Ives (Roche) Ltd.

Display Advertising Manager: Marc Warren
Senior Sales Executive: Steven Miles
Sales Executive: Kathryn Tracey
Sales Manager - EXplodE: Mark Parker
Marketing & Promotions: Suzanne Chamberlain
Office Administrator: Jacqui Ramrayka
Front Cover Photo: David Pritchard
Repro & Typesetting: Ebony

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 Telephone: 0171 287 5000
Advertising email: stevenm@dotexe.demon.co.uk
Subscriptions 0171 439 4222 exts. 2212/2213/2214 (Fax: 0171 439 0110)

EXE is available 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 editorial@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.
 ISSN:0268-6872


You're on the road. With a client.
You need to locate a bottle design.
Your database has 40,000 bottles.
And all you know is the shape.
Sort of.

Perfect.

The human mind has the ability to recall information visually. But can a client/server database advance that far?

Well, IBM's exclusive Query By Image Content (QBIC) technology for DB2 should open your eyes. It actually lets you locate a range of records based on shape. Or colour. Or even texture. Which means faster access to the data you need – including multimedia – to make better business decisions.

Can your software do this?

Not that such advances should surprise you. IBM has led the way in databases as long as there have been databases. (Indeed, DB2 is at work in over 90% of the *FORTUNE 500*.)

DB2 has a history of offering useful innovations for information management – with tools that let you do everything from managing your database remotely to setting up a database to exist in several places at once. And while DB2 stands out from other databases, we designed it to fit very comfortably into any business environment.

DB2 meets open industry standards, so it will run on many popular systems. And it's been engineered so your database will deliver both speed and efficiency from any size server. From two users to more than 100,000. From megabytes of information to terabytes. To make your business information more accessible, just contact Donna Ratcliffe on 01705 498151, or visit our web site at http://www.software.ibm.com/software/data.html

You'll find out why, for so many companies, DB2 is the shape of things to come.

IBM

Solutions for a small planet

✎ CIRCLE NO. 371

SoapFlakes

Hot skills

I remember thinking Physics O-level was difficult. Two years later I realised the A-level was difficult, the O-level had been easy. I would presumably have changed my mind about that too, if I'd gone on to do a degree in Physics. (I didn't, because I wanted to get a job after graduating.)

Learning curves always look steeper from the bottom. This is a particular problem in our industry, where there is a constant high turnover of skills. If your skill set, as you get more experience, is not pro-actively managed, you'll find new graduates looking more valuable (and cheaper) than you. Unfortunately, managers - especially those without a technical background - don't understand this. So here are some rules of thumb you might want to show them.

1. *Learning syntax is easy, learning concepts is hard.* If you know COBOL, you won't have too much trouble with RPG, and if you know one OO4GL GUI tool, you won't have much trouble learning another. But going from COBOL to an OO4GL GUI tool is hard: you have to learn fundamentally different concepts such as object-orientation and event-driven programming.
2. *Climb learning curves one at a time.* The sinking feeling I had when trying to learn Z80 assembler on a TRS-80 was because I was struggling with concepts, context (like 'to display something on the screen, write to this area of memory') and syntax at the same time. Learning theory says: break things down into two-week segments, get tested at the end of each segment and take things at your own pace.
3. *The second person up a curve has a much easier time, because there's someone at the top to pull them up.* No-one has yet built an on-line help system that is quite as effective as being in contact with an expert.
4. *If developer's productivity is correlated with anything, it's correlated with the number of environments a developer knows* (this is from Ed Yourdon). This is an empirical finding that probably has to do with concept learning again.
5. *The best way to learn is by doing something real.* There's a limit to what you can get out of books and case studies - at some point they have to let you loose.

My company uses a set of 'techniques' to teach concepts such as data modelling or functional decomposition. Getting qualified in a technique involves some reading, real work alongside someone who is already qualified, and a rigorous viva. Tool skills get special attention. Our policy is: (a) the more the merrier; (b) give the first person to learn a tool the most time; (c) there'll always be a better tool out next year; and (d) developers love new tools, managers hate version 1.0 - this time managers are right.

Finally, we distinguish between people who want to get a job done (project managers) from people who care about skill development and career progression. Trying to fit these different priorities all in the same head doesn't work.

Easy for us to say: we started from a clean sheet and our core business is to grow people. But to keep up with the game, you're going to do more learning after the age of 21 than before. The least you can expect is for your organisation to support you.

Laurence Holt is CEO of Quidnunc, a 40-person Client / Server consultancy based in West London. Contact him at laurenceh@quidnunc.com.

Choosing a language

The choice of language for a development project is very complex and subtle: all too often the choice made is made for the wrong reasons.

Since C is quite a common language taught in University and college courses, and it is often viewed as being 'better' as a language by novice programmers, it stands to reason that there is a significant pool, and wide spectrum, of expertise in C floating in the job market. This makes it both easy and economic for companies to employ programmers with these skills.

Unfortunately one often encounters systems which would have been considerably easier to develop, debug and maintain had they been written in an appropriate language, rather than a language which the company, simply considering labour costs, found cheap in the short term. It is foolish to use a screwdriver to drive nails into wood but this is exactly what many software developers do by using the wrong language. Choosing C because it is 'cheap' in labour terms and easy to recruit for is a bad idea for significant developments. It may be that choosing another language will reduce development costs because the language actually helps to avoid errors during the development. C still has a significant role to play. My main concern is that it is turning into the 'Swiss army knife' of the software community, and that one day we will realise that we are better off using a monkey wrench to undo the plumbing than struggling to make a Swiss army knife do the job.

C and C++ have become *de facto* standards and virtually every programmer has some skill in one or both languages (and this varies considerably). If pressed to pick the number one programming language for producing code which is (a) obfuscated beyond recognition and (b) prone to bugs which are hideous to track down, I'm afraid C would be right up there on top of my list. I have witnessed some heinous crimes committed in the name of C. Well-written C may contain subtle errors; badly written C will cost your company dearly.

One of the major difficulties for C programmers to tackle in any significant development is that of memory leaks. One possible solution, garbage collection (currently being investigated for a future inclusion in C++), does not dispense with this problem, but can alleviate the suffering. Unless your system is particularly desperate for run-time then I would say garbage collection is more of a blessing than a curse. If compiler writers could put a switch on their systems to compile with or without garbage collection I am sure it would be welcomed by a great many people. Certainly if C++ is to be imbued with garbage collection, I would expect this switch feature to be present on all C++ compilers.

I would encourage developers to consider the choice of language as carefully as they consider the choice of design tools and methods, word processors, communication packages, hardware, etc. Do not simply opt for C 'because it's there'.

Mark Bools is a software engineer currently working on quality engineering and software / business process improvement within Siemens Traffic Controls Ltd.

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

VISUAL BASIC 3 ADD-ONS

These will still be available for some time to come.

Comms - Async

Fax Plus for Win	£175
FaxMan SDK	£390

Comms - Network

Apiary Dev Suite for NetWare	£156
Distinct TCP/IP - Visual Edition	£265
dsSocket 1.1 (Intro)	£65

Database

ADE/VBX	£350
CodeBasic 5.1	£140
List & Labels for VB	£279
VB/ISAM MU	£150

Graphics - Charting

Chart FX 3.0 (16-bit only)	£210
Charting Tools for Win - VB	£180
Real-Time Graphics Tools - VB	£300
VBGraphix	£270

Graphics - Image Files

Image SDK Plus/VBX 2.0	£250
ImageKnife/VBX Std 2.0	£200
ImageMan/VB 3.1	£239

Grid Controls

Data Widgets	£99
Grid/VBX	£75
Spread/VBX	£171

Multi-Function

Borland Visual Solutions Pack	£59
Designer Widgets 2.0	£99
Muscle (Win)	£125
VBlite 1.0	£130
VBTools 4.0	£115
Visual Developer's Suite	£216
WinWidgets/VBX	£160

Sundry Components

CADControl	£365
d-Barcode VBX/DLL	£94
VB/Magic Controls	£120
Visual CAD Dev Kit	£520

Sundry Controls

3D Graphics Tools 4.0	£130
EDI-VBX 1.0	£705
Gantt/VBX	£195
MediaKnife/VBX	£290
VBX Artist	£240
Visual Instrument Panel Cntrls	£150
VSView/VBX	£105

Text Editor Controls

HighEdit 3.0	£199
TX Text-Control Standard	£180

Tools

JET Inspector 2.0	£475
SpyWorks-VB 2.0	£100
TMS Tools 1.1	£99
VB Compress	£99
VBAssist 3.5	£140
VB/DLL 2.05	£165
VERSIONS/VB 1.1	£135

C & C++ FOR DOS

Comms

Essential Comm 5.0	£265
MagnaComm/DOS	£205
SilverComm "C" Asynch 4.02	£195

Database

c-tree Plus 6.4b	£565
D-ISAM	£495
SoftFocus Btree/ISAM	£75

Graphics & GUI

3D-Ware Prof	£99
Fastgraph 4.0 (Ted Gruber)	£195
GX Graphics 3.0	£155
MetaWINDOW-DOS 4.4	£199
Zinc Engine & DOS Key 4.1	£634

Maths & Scientific

C/Math Toolchest & Grafix	£45
Huge Virtual Array & NAT 3.0	£215
Science, Eng & Graphics Tools	£115

Screen

C/Windows Toolchest	£45
Greenleaf Datawindows 3.0	£225
TCXL UI for DOS/Win 6.2	£135

General & Systems Libraries

GX Sounds	£200
MTASK	£215
TTSR Ram Res Dev Sys 2.03	£100

Tools

C-DOC Pro 6.0	£275
C-Vision for C/C++ 4.0	£145
CodeCheck (Professional)	£475
PC-Lint for C/C++	£135
Source Print+ 5.5	£235

PROGRAMMING tools

Ada	Assemblers
Basic	C/C++
Comms	Cross Dev
Custom Controls	Database
Debuggers	Delphi
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.

C & C++ FOR WINDOWS

Comms

Async Pro for Win	£135
COMM-DRV/LIB 16.0	£105
Greenleaf CommLib 5.2	£235

Compression

Crusher! Win DLL w/Source	£235
Greenleaf ArchiveLib	£210
PKWare Data Comp Lib for Win	£175
TCOMP/Multi-Platform 2.12	£105

Database

Borland Database Engine 2.0	£205
CodeBase 5.1	£248
CXBase Pro	£500
DataBoss for Windows	£410
DBTools.h++ for ODBC	£340
Greenleaf Database Library 4.0	£180
List & Labels for Win (Pro)	£395
POET Personal SDK 3.0	£169
Raima Database Manager	£380
Spread/VBX++	£171
Velocis	£340

Graphics - Charting

Charting Tools for Win 2.0	£180
Essential Chart for Win	£320
GraphiC/Win 7.0	£360
Graphics Server 4.0	£245
Real-Time Graphics Tools	£360

Graphics - Image Files

AccuSoft Image Lib/Win 5.0	£610
Ad Oculos (Image Analysis) 2.0	£325
ImageKnife Pro 2.0	£280
ImageMan	£399
LEADTOOLS Prof 5.0	£675
Light Lib Images Pro	£315

Graphics & GUI

3d Graphics Tools 4.0	£240
ProtoGen+	£190
RWCanvas.h++	£340
WinGKS	£575
WinMaker Pro 6.0	£725
Zinc Engine & Win16/32 Key	£634

Maths & Stats

IMSL C Numerical Libraries	£495
IMSL Math Module for C++	£495
Math.h++ 4.1	£340
Money.h++	£340

Sundry Components

HeapAgent	£420
TG-CAD Prof 5.5	£770
Tools.h++ 6.1	£340
WinWidgets++	£240

Tools

CC-RIDER for Win16	£250
KPWin++	£600
SOMobjects Dev Toolkit	£220
Visual Parse++	£289

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

MS Fortran PowerStation Pro 4	**£443**
MS Visual Basic Std 4.0	**£72**
MS Visual Basic Prof 4.0	**£365**
MS Visual Basic Ent 4.0	**£715**
Microsoft Visual C++ 4.0	**£368**
Borland C++ 4.5	**£280**
Borland Pascal with Objects 7.0	**£250**
Delphi	**£279**
Delphi Client/Server	**£806**
Paradox for Win 5.0	**£280**
Turbo C++ for Win 4.5	**£68**
Turbo Pascal 7.0	**£94**
Visual dBASE 5.5	**£285**

WITH FULL TECHNICAL SUPPORT

C ++ COMPILERS

Borland C++ 4.5	£280
Microsoft Visual C++ 4.0	£368
Salford C/C++ Win Dev	£395
Symantec C++ 7.0	£199
Turbo C++ for Win 4.5	£68
Watcom C/C++ 10.5 (Intro)	£139
High C/C++ for Ext-DOS/Win	£620
Microsoft Visual C++ 1.52	£70
Salford C/C++ DOS Dev	£195
Turbo C++ 3.0	£66

VISUAL BASIC FOR DOS

MS Visual Basic for DOS Std	£96
MS Visual Basic for DOS Prof	£230
NetPak Pro for DOS	£135
PDQComm 2.62	£65
db/Lib Prof 3.0	£195
VB/ISAM MU	£150
Graphics Workshop	£105
GraphPak Pro	£99
Compression Plus	£79
Printer Plus	£130
ProBas 7.1	£210
QuickPak Prof 4.19	£145

VISUAL BASIC 4

Visual Basic Enterprise 4.0	£715
Visual Basic Professional 4.0	£365
Visual Basic Standard 4.0	£72
ButtonMaker	(Oct) £75
Designer Widgets 2.0	£99
Erwin/Desktop for VB4	£599
Spyworks-VB Prof 4.0	(Q4) £195
VB Assist 4	(Oct) £140
XRef 2.0	(Oct) £105

BASIC LANGUAGE

CA-Realizer 2.0 (OS2&Win)	£179
GFA-BASIC for Windows	£80
PowerBASIC Pro (Win&DOS)	£210
BBC BASIC-86 Plus	£75
TrueBASIC Std 3.0	£88

OLE CONTROLS

OLE Controls will be supported by most Windows tools over the next few months. We have the widest range available!

Comms - Asynch

Comms Lib 3.0	(Oct) £115
PDQComm 3.0	(Oct) £145

Comms - Network

Distinct TCP/IP Visual	(Nov) £265
NetPak 2.0	(Oct) £145

Compression

Compression Plus 4.0	(Oct) £175
DynaZIP-32 3.0	(Oct) £210

Database

Controls for Btrieve 3.0	£180
Crystal Reports Pro 4.5	(Oct) £295
MyData Control 1.0	(Oct) £115

Graphics - Charting

Chart FX 3.0 (16&32-bit)	£315
First Impression 2.0	(Oct) £180
Graphics Server 4.02	(Sep) £235

Graphics - Sundries

Accusoft IFL/OCX 16	£415
d-BarCode/OCX	(Dec) £149
Dazzle/OCX	(Q4) £420
MetaDraw	(Oct) £240
Resource Manager/OCX	(Q4) £195
Schedule/OCX	(Nov) £195

Grid Controls

Formula One OCX 3.0	£180
Grid/OCX	(Q4) £75
Spread/OCX	(Oct) £171
TrueGrid 4.0	(Q4) £155

Maths & Scientific

ProMath OCX	(Q4) £130
QuickPak Scientific 3.0	(Oct) £145

Multi-Function

Aware/OCX	(Q4) £110
OLETools 5.0	(Oct) £145
QuickPak Pro 4.0	(Oct) £180
VBlite OCX	(Q4) £130
VSView/OCX	(Oct) £145

Multi-Media

FXTools/VB OCX	(Q4) £290
MediaKnife/OCX 1.0	(Q4) £299
VideoPlay/OCX 1.0	(Q4) £72

Sundry Controls

Calendar Widgets	£99
Custom Control Factory 4.0	(Q4) £40
Drag-IT/OCX	(Dec) £240
StorageWorks 1.0	(Q4) £100
Tab/OCX	(Oct) £75
Tlist 3/Pro	£195
VersionStamper 4.0	(Q4) £100
VisualSpeller OCX 2.0	(Oct) £115
VS-OCX	(Oct) £110
VSFlex/OCX	(Oct) £145

Text Editor Controls

ALLText HT/Pro 4.0	(Oct) £335
VisualWriter 3.0	£180

OS/2 WARP

Borland C++ 2.0 for OS/2	£329
High C/C++ for OS/2 3.3	£435
IBM C Set++ FirstStep for OS/2	£89
IBM VisualAge C++ 3.0 w/Docs	£364
Watcom C/C++ 10.5 (Intro)	£139

News & Views

NEXT LEVEL RAD?

How would you like to generate an entire application at the touch of a button? Complete with Browse, Form & Report procedures for every selected file!

Wouldn't it be nice if you could produce versions for Windows 3.x, Windows 95 & Windows NT, all using the same VBXs. It would be even better if all these versions were multi-threaded & *fully compiled* with small executables.

Can your present tool do all this? Of course not! Then you better call us to find out more about

CLARION FOR WINDOWS 1.5

All you have to do is define the data dictionary containing file & relationship definitions, with optional pre-formatting preferences for window & report controls.

Then the Wizard of Wizards - the *App Wizard* - generates complex applications with no intervention, including multiple-order browse lists selectable by tab; synchronised parent-child dialogs with a form for the parent on one tab, and a listbox displaying related child records on another, etc.

Unlike most Wizards, changes in the data dictionary migrate via live links to the application file. You can concentrate on the data model and business rules, and let Clarion do the grunt work. This is real RAD!

The best news is that we are virtually giving it away at **only £189** until the end of the year. Call us right now for further details.

ODBC DRIVERS FOR WINDOWS 95 SPECIAL OFFER

The Visigenic ODBC DriverSet includes drivers for Windows 3.x, Windows 95 & Windows NT. Databases include Informix, Sybase, Oracle, Ingres & Microsoft SQL Server. Available for **only £85** until the end of the year (on-line docs, limit of 5 copies per customer site).

XY-QUERY FOR OS/2

is the fastest and easiest tool for accessing your DB2 data. It is object-based, using notebooks, toolbars, etc with Query Builder which hides the complexity of SQL. Too many features to cover here - suffice to say that if you use OS/2 and need to query DB2, call us now for full details. **Only £140.**

Mayhem!

Some people thought it was never going to happen. Some people thought it already had. But finally, on (whatever the date was), Windows 95 was launched.

Never before in this business has a new product been so eagerly awaited. For years, hints and leaks were coming from Microsoft about their wonderful new operating system. Eventually, some real specifications emerged, then beta copies.

The beta programme was the largest the industry has ever seen. While version after version crashed and burned, as deadline after deadline was missed, people were asking, if it ever were to be finished, who but the beta testers would be left to buy copies?

And then, finally, amid the most expensive blaze of publicity we've ever witnessed, it was launched.

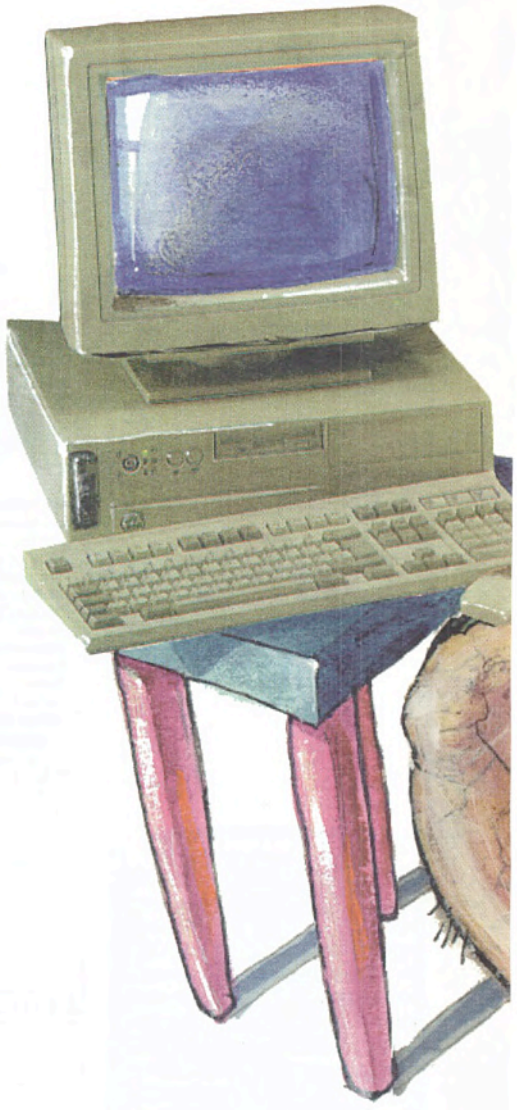
It's difficult to talk about Windows 95 without a list of superlatives. It was incredibly late, it was comprehensively re-specified, it was ridiculously hyped. On the eve of its launch, the TV news was awash with items about it. The following evening, the same news programmes were full of stories about what a waste of money it was. Yes, Windows 95 was hyped, but hype itself is not enough to explain the remarkable level of interest.

The buying public, I think, is desperate for something new in computers. For the last two or three years, the marketplace has been steadily stagnating, due to a combination of massively rising costs and a dwindling pool of ideas. There can't have been many people who bought Windows 95 who didn't know exactly what it was six months ago. But hope was triumphing over experience, and that same public wanted to be surprised.

That was the key to Microsoft's marketing strategy. Never mind that the thing had been downspecified times without number. Never mind that it's less compatible with legacy code than NT. Do something big, something glamorous, and the public will think they're being sold the novelty they're crying out for.

It could have worked, but two things went wrong. Firstly, the software itself didn't work properly. Within weeks, users were grumbling about trashed disks, about peripherals that wouldn't install, and about programs that wouldn't work. In the same record numbers that users bought the software, they were now uninstalling it.

The second thing that went wrong was, thanks to its size, Microsoft defined where the marketing battle lines were to be drawn. The other companies were perfectly happy to pick up the gauntlet that Microsoft threw



down. 'See the hourglass?' asked IBM. 'Hey, it's not there,' answered the chuffed user. 'Where do you want to go today?' asked Microsoft. 'Good question; how about tomorrow, though?' countered Apple, conveniently ignoring the fact that Apple has achieved nothing of commercial note in the last five years. Parts of the software which should have been a big part of the image were being devalued in public: CompuServe attacked MSN, for example, and everybody remembered DoubleSpace.

It doesn't really matter that IBM and Apple are as devoid of ideas and adequate programming skills as Microsoft is. The nature of the market is that nobody has to sell any more machines or packages; they just have to make the competition sell less. Microsoft, by design or accident, has few friends, and so is everybody's competition.

Most users don't need anything more sophisticated than vanilla Windows, but because most users already hadit, Microsoft wasn't selling any more. If Microsoft really believed in the potential of MSN, it could have shipped the software free to anyone who wanted it. That wouldn't have helped much, because a steady trickle of products wouldn't have created the excitement that the market so desperately craves. No, what created the excitement was an old product in new packaging, lots of glitzy adverts, and years of whetting the market's appetite.

In this day and age, Microsoft really can't be censured for having no ideas, for making software that doesn't work properly, or for delivering it late. Nobody has any ideas, nobody makes software that works properly, and everyone is delivering late. But again, that's not the issue. The perception is that, because nine out of ten users who expressed a preference prefer Microsoft, what's bad for Microsoft is good for everyone else.

That perception is dead wrong. The battle for the desktop was fought out years ago. Microsoft won, and there's nothing anyone can do about it. At this point, picking on Microsoft doesn't do anyone any good, because nobody has anything that works any better. This kind of strategy can only kill sales for everyone, because once the buying public starts asking critical questions, it will find that nobody has any real answers.

Once upon a time, the big players in desktops co-operated with each other. Sure, there was healthy competition, and brand identities, and all the other things that help to carve up a market. But now, Windows 95 has done something which this industry has never had to deal with before. Now the big players are at each other's throats, not caring how badly injured they are, so long as they do some damage to the competition in their desperation to sell anything at all. That's the real novelty in Windows 95 - that while there was something to compete with, everyone was happy, but now the pickings are scarce everyone is terrified of a facelift with nothing inside.

It is to Microsoft's credit that it has never attacked its competition in this way. It may even be that Microsoft will win by pointedly not getting involved in this infighting. It doesn't have to get involved; it can sell anything it makes. The only question is: can it sell enough?

It could be that the industry is fighting for scraps from Microsoft's table, and it could be that Microsoft is on the floor with the rest of them. The success of Windows 95 will decide, but given that Windows 95 was released unfinished, years late, I suspect the latter. If so, the scraps will hardly be enough to keep everyone alive. ■

Jules is a programmer, but he won't be for much longer. If you're quick, you can phone him on 01707 662698, or you can email him as jules@cix.compulink.co.uk.

develop to advantage

Build better applications

Add performance, functionality & versatility

MKS Source Integrity

MKS Source Integrity is the most comprehensive configuration management solution for client/server, cross-platform development with powerful new features:

- Easy to use GUI creates fully functional project in minutes.
- Sandboxes for personalised workspace.
- Visual differencing & visual merging
- Seamless integration in Visual C++, Visual Basic, Borland C++, PowerBuilder, File Manager

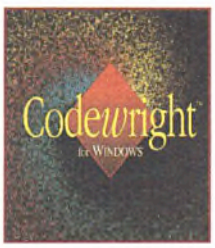
Call for multi-license prices - single user £359

Doc-To-Help

Use Word 2 & Word 6 to create Windows Help files effortlessly. Add hypertext links to existing documents and prepare Help files and printed documentation from the one source. Convert Help files to text, graphics etc. components, to bring existing projects into Doc-To-Help usage. Multi-user licenses available. £259

WatcomC/C++ version 10.5

Watcom C/C++ delivers a professional, cross-platform 16-bit and 32-bit development system that produces the fastest executables for the widest range of platforms. Using the same integrated development environment, you can target all platforms including Windows NT, Windows 95, Windows 3.x, OS/2 Warp, extended DOS and more. New features include: Blue Sky's Visual Programmer, Windows 95 Ready, MFC 3.0 for Win32 and direct support for popular revision control packages. Special offer only £139



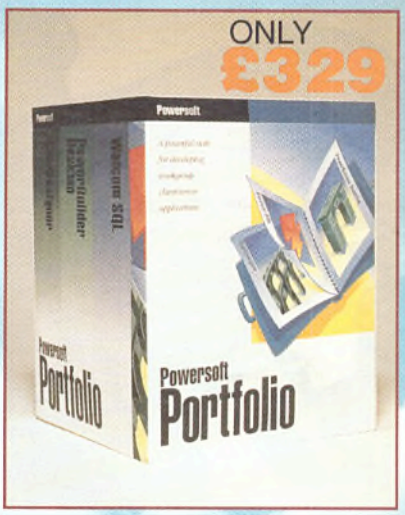
CodeWright Professional 3.1 - Programmer's Editor

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/95. £159

Powersoft Portfolio

3 Powerful tools seamlessly integrated to boost your client/server development

StarDesignor
Data Construction made easy
PowerBuilder Desktop
Powerful intuitive development
Watcom SQL 3-User
SQL Database power right out of the box



Blast off with the new Powersoft Portfolio from the acknowledged leader in client/server development tools. For the first time, everything you need to start defining, building, and deploying serious workgroup applications has been brought together in one powerful, affordable suite.

The Suite for Serious Workgroup Applications. Only £329
All products also sold separately.

BoundsChecker Professional

BoundsChecker Professional redefines automatic error detection for C/C++ developers using Windows 95 or Windows NT. Professional Edition introduces breakthrough technologies to capture even more information, with extended API compliance checking for all three Win32 implementations. Integration into the VC++ environment, enables BoundsChecker to be used at all stages of development.

CodeBase

CodeBase the fully xBase compatible RDBMS, is extremely flexible and available for C, C++, VB, Delphi and Borland Pascal. CodeBase is single/multi user and client/server compatible, and it is supported on a wide range of platforms including DOS, Windows 3.x, Win 95, NT, OS/2, Macintosh, SCO, AIX, HP, Solaris, UnixWare, SunOS etc... Free CodeReporter and CodeControls included.

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

TO ORDER CALL 0171-833-1022
FAX 0171-837-6411
System Science, 1-6 Bradley's Close, White Lion St. London N1 9PN

SYSTEM SCIENCE

your first choice for better development tools

CIRCLE NO. 373

EXE Challenge '96

EXE is now planning next year's challenge. This will take place in February at the Windows Show organised by IT Events at Olympia, Kensington. Competing teams will battle against each other to design a piece of software for a charity and become the EXE Challenge Winner 1996. Not only will you be able to demonstrate your skills, but there's also the opportunity for software vendors to show the world just how good their tools are! Please call Suzanne Chamberlain on 0171 287 5000 to register for an information pack.

Rapid adoption of RAD

The Dynamic Systems Development Method (DSDM) Consortium is keeping up the pressure for the industry to adopt a standard Rapid Application Development (RAD) method. The user-driven Consortium, comprising some 450 members, aims to provide a common definition of RAD to improve the development process. Version 1 of the DSDM method was completed earlier this year; version 2 is to be launched at the London-hosted European RAD Conference in December. To find out more contact the DSDM Consortium on 01233 661003.

Take your partner by the URL

Netscape Communication's Development Partners Programme appears to be taking off, with several thousand developers already members. The programme is intended to make life easier for developers creating applications for, and with, the Netscape application platform. A number of core components are included: a subscription to an electronic newsletter; on-line access to documentation on Java and Netscape client and server APIs; technical support; early access to Netscape development tools; and discounts on software. You can apply to become a 'partner' at http://home.netscape.com/comprod/development_partners.

New pickings for Full Moon

Apple recently outlined enhancements to what it politely refers to as its 'developer relations efforts'. According to an IDC report, the Macintosh development platform compares favourably to Microsoft Windows in a number of respects, including higher software revenues and lower development costs. The main change Apple has instigated for UK developers is the appointment of Full Moon Software Distribution (01628 660242) as the exclusive resellers of Apple's development tools, which should speed up the process of obtaining products previously only available from the US.

Novell to sell UnixWare

SCO and Novell have agreed that SCO will purchase Novell's Unix business. SCO will also license Novell's NetWare Directory Services and other NetWare 4 technologies as the basis for networking services to be incorporated into future SCO products. Particularly, it plans to merge the SCO OpenServer Release 5 and Novell's UnixWare 2 product lines to create a unified Unix operating system that will contain integrated NetWare-based networking services.

Initially, however, SCO Open Server and UnixWare will be supported and enhanced individually. Releases of both that will sport integrated NetWare services are due in the first half of 1996. The summer of the same year should also see a beta of the proposed merged operating system, with a software migration toolkit that will allow developers to get started on targeting the new platform. SCO claims that the merged Unix, when released in 1997, will offer binary compatibility with existing applications written for its parent operating systems.

The agreement between SCO and Novell has come at a significant time as the industry is attempting to consolidate a standard 64-bit API for Unix. (64-bit operating systems allow the flat addressing of up to 16 exabytes of RAM). To this end Hewlett-Packard has formed business relationships with both SCO and Novell to release a series of merged Unix products that will define an evolutionary path to 64-bit networked Unix on the HP/Intel architecture.

It's the computer's fault!

Earlier this year, MORI conducted a survey on behalf of Motorola on the perception of IT in Britain. Some of the results should brighten up software developers. 'It's the computer's fault,' one of the most common excuses made by companies who have provided wrong information, was offered by less than 10% of the respondents who felt uncomfortable with using IT (see chart). For the research, 339 adults (over 16) were interviewed.

Barriers to using IT *(Source: Motorola Ltd. and MORI)*

	General Public	Non IT users	Heavy IT users
Computers make mistakes	7%	8%	6%
Computers break down	1%	1%	-
Concerned by confidentiality	6%	5%	9%
Dislike IT	7%	9%	3%
Prefer personal contact	33%	22%	52%
No experience of IT	33%	41%	15%

The most surprising point was not mentioned in the survey itself but raised by A. William Wiggenhorn, President of the Motorola University: 'IT companies need to train the consumer.' By that he is admitting that the latest software and hardware *is* complex to use because of all its fancy functionality. For the consumer of tomorrow to be able to use (and hence buy) the product in development today, he'll need some training! Motorola is on 01753 575555.

The corporate game

The Disneyland Paris Convention Centre was the venue for IBM's first European Technical Interchange, which ran last month between 2nd and 6th October.

The Interchange was IBM's first major conference since its acquisition of Lotus. And Jim Gant, Vice-President, Solution Developer Operations, didn't miss the opportunity to tell the 2000 or so delegates why $3.52 billion was money well spent: 'Lotus Notes is the way of the future for us.' Just a week after this conference, Jim Manzi resigned as CEO of Lotus Development and Senior Vice-President of IBM.

Gant wants to see greater distribution of information to developers. To this end, IBM is putting together a new Solution Developers Programme, to commence January 1996. The aim is to combine the 20 or so existing Developer Assistance Programmes into one, with the focus on commercial software development. Check out *http://www.austin.ibm.com/developer/* for the latest information.

No IBM conference would be complete without a word on the rattle in Seattle. Gant's thread of attack took the view that nobody would commit business-critical applications to v1.0 of anyone's operating system. He spoke briefly of a fully-functional Lotus SmartSuite for OS/2 and pulled out a figure of 12,000 native OS/2 applications - a bit paltry. Gant blamed the marketing of an otherwise excellent product. My interpretation of this is: if Microsoft will have spent an estimated $500 million launching Windows 95 this year, IBM *has* spent $3.52 billion prolonging the existence of OS/2.

Cliff Saran

The Telecom show happens every four years in Geneva. This year's event was very different from previous ones. With the convergence of the telecommunications and computing industries, the stands looked much more glamorous than they used to be. All the traditional fun of computing shows was to be found: dancers, singers, big screen attractions, freebies... All in all, 189,671 visitors attended, not counting children.

The convergence was not obvious on all the stands (far from it), but this was definitely a trend of this year's show to the extent that one of the keynote speaker was Andy Grove, President and CEO of Intel. The convergence goes both ways. On one side, the big telecomm operators are using more and more computer techniques (eg voice switching over IP experiments) and offering new services (eg video on demand, multimedia, online shopping). On the other side, the computing industry wants a slice of the huge and growing telecommunications market (eg call processing on the PC, video-phone applications). Let's have a peek at a small selection of stands...

The best display of Computer Telephony Integration (CTI) applications was definitely on the Dialogic stand (tel. 01252 844525 or *http://www.dialogic.com*). This US company, founded in 1983, designs and produces hardware and software components for CTI. Its stand was hosting some 25 partners making use of its technology. Eight of them (Apex, C3, Expert Systems, MediaSoft, Nsoft, Parity, Stylus and Voysys) were showing new versions of toolkits aimed to ease the development of voice processing and call routing software. Applications such as fax-on-demand, interactive voice and fax response, voice mail and electronic mail reader could be seen (and heard).

Next stop: Versit. It's a 'global initiative' by Apple, AT&T, IBM and Siemens, the aim of which is to release general interoperability specifications for *free* public use. It's a middle of the road approach between a formal standardization process which can take a long time and many compromises to get anywhere, and 'standards' defined by *one* company. Among the specifications available to download (*http://www.versit.com*) is GeoPort Universal Network Port, a 2 Mbit/s serial interface designed to replace the limited V.24 serial port in all CTI applications. To interface to GeoPort, operating systems or applications will need specific drivers. PowerMacs are currently equipped with this bus, but PCs are not yet. Versitcard, another free to use specification, defines how to exchange information such as business card details between two, otherwise incompatible, systems. It is based on OpenDoc's Bentograms and, for palmtops, on 'Simplegrams'. Additional work is to be done to extend Versitcard to allow information exchange on the Net.

Briefly, here are some other attractions. Nokia demonstrated cellular data transfer over GSM via an infra-red phone connection. The software was based on Microsoft's still unannounced IrDA protocol stack for Windows 95. Ericsson presented ODBC- and MAPI- based software to control its phones from a PC (via a serial link). Microsoft itself announced ISDN support for Windows 95, and TAPI on NT. Chorus presented a report on the increase in the use of Unix micro-kernels instead of proprietary OSs in the embedded market. Hewlett-Packard showcased the OmniGo 700, a modified HP200LX integrating Nokia's data card and support for a Nokia phone. Also in partnership with Gemplus and Informix, H-P announced a 'Personal Information Card': a smart-card holding multiple databases.

David Mery

Object Expo Europe, this year hosted by the QEII Conference Centre from 25th to 29th September, is the place to go if you have more than a passing interest in object technology. My curiosity was certainly piqued as I made my way to Westminster.

The first person I met was Pierre Haren, the founder of the French company ILOG. ILOG publishes a suite of C++ software components designed to solve real-world problems involving constraint satisfaction, resource allocation, task scheduling and rule-based process supervision. These sorts of problems are very general and arise in such diverse domains as traffic monitoring and crew allocation by airlines.

Two of the packages Haren explained in detail were ILOG Server, an 'object server' architecture for real-time groupware, and ILOG Talk, a Lisp-based, object-oriented language compatible with C++. ILOG Talk is normally interpreted but also compiles to C, allowing it (in Haren's words) to 'surf' existing compiler optimisations.

I then met with Iain Gavin, Product Manager of Rational Software. Rational has been through a bit of an upheaval recently, with Jim Rumbaugh joining its ranks last year to work with Grady Booch on a unified object methodology. The unified method is almost ready, a Request for Comments on version 0.8 having been issued by the Object Technology User Group. Iain explained the rationale behind the version number: it was intended to suggest that Jim and Grady are *almost* there, but not quite!

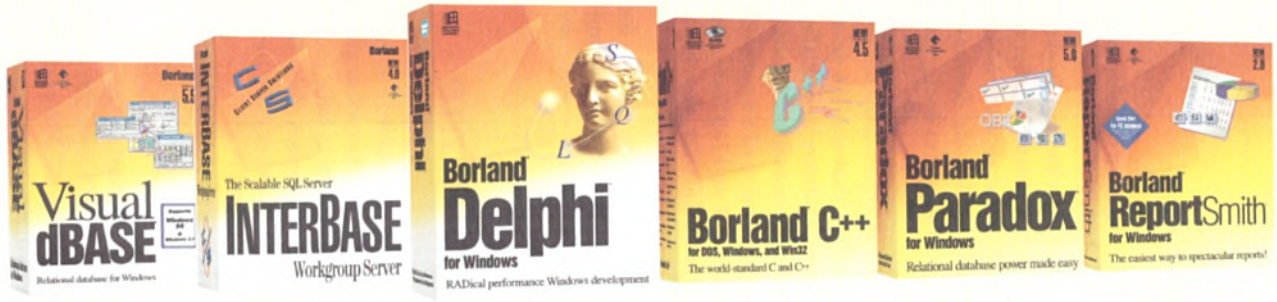
The new version (3.0) of the Rational Rose/C++ CASE tool does not yet enable hybrid models, although it now supports iterative development with OMT as well as Booch - the user can switch between two alternate 'views' of the same underlying object model.

At last I had an hour or so to peruse the exhibits on show in the labyrinthine QEII Centre. Cadre were showcasing their vastly improved CASE tool ObjectTeam for OMT, which as well as actually working, now provides features like version control at diagram, sub-project and project levels, support for project 'phases', and some Cadre-specific extensions to OMT.

A quick stop at Sun's stand revealed how Unix tools are still a few steps ahead of their PC equivalents: SunSoft's Visual Workshop for C++ supports incremental linking, a fast template instantiation scheme, and 'Fix and Continue' debugging. This latter feature allows you to modify source code at run-time and continue debugging without rebuilding. The compiler is also capable of 'threading' your code during the optimisation phase to allow independent code sequences to run concurrently on SMP systems.

To close the day, Russell Prince-Wright, Director of European Operations at ParcPlace, explained to me ParcPlace's vision of the 'virtual' corporation, an organisation that exposes secure 'business objects' to its partners, such as vendors or suppliers, over the Internet.

Roland Perera



Puts you in control of your business data

Visual dBASE® combines leading edge visual tools with a flexible yet easy programming language to deliver the most productive database for business professionals and applications developers.

The scalable SQL server

This high-performance SQL server is specially designed to make it easy for you to scale PC and network applications up to the robust world of Client/Server.

RADical performance windows development

Delphi™ is the next generation visual development tool. It combines an optimising native code compiler, visual RAD tools, and a scalable database engine, making it the fastest way to the fastest applications.

The high productivity C and C++ toolset

Borland® C++ targets DOS, Windows 3.1, and Win32 from a single development environment. Buy now and after Windows 95 ships you'll receive a free Borland C++ 4.5 update.

Easy and integrated database power

Superior ease of use makes Paradox® a favourite desktop database with end users and developers alike. It's the best integrated companion for users of office suites.

Spectacular reports from any data

It's the easiest way to create spectacular reports. Seamless connectivity to every major SQL and PC database lets you work directly and interactively with "live" data.

Increase your productivity with these companion products:

- **The RAD Pack for Delphi**
  All the development tools you'll need for maximum productivity and Rapid Application Development.
- **Visual Solutions Pack**
  A collection of more than 30 custom controls for you to plug into your favourite visual tool.
- **Delphi/Link for Lotus Notes®**
- **Paradox Developer Tools**
- **Paradox Runtime**
- **Visual dBASE Compiler**
- **Borland Database Engine**
  High performance database engine for Windows applications.

A single-minded focus on developer productivity

Borland has consistently listened to software developers and, as a result, we have come up with a complete range of development tools. Each product, a component of our development toolset, is designed with one objective in mind... making programming easier so that you become more productive.

Visual tools, reusable code, incredible performance

Borland tools give developers what they want… ease of use, innovative features, speed, and reliability. Take the Graphical Data Modelling in Paradox that maintains your entire Applications' Table Relationships or the Two-Way-Tools™ in Visual dBASE that create code for you automatically while you design visually. Or Borland C++, the only tool that lets you reuse VBX controls in both 16 and 32 bit applications. Or the blazing speed of Delphi that lets you create .EXEs that are 10–20 times faster than interpreted code.

Superior tools for DOS, Windows, and Windows 95 development

Whether you're developing mission-critical custom applications, or shrink-wrapped software, Borland offers the most complete selection of high-performance, high-productivity programming tools. Combining power, innovation, and ease of use in an open development environment, there's a Borland programming tool that's right for you. Call us, or see your local reseller to find out more about the exciting development products shown here. And don't forget to ask about our many productivity enhancing "companion products".

Borland Connections - your partnership with Borland

The Borland Connections scheme is designed to support relationships with organisations using Borland products, including Corporate organisations, Resellers, Training Companies and System Integrators. Members are provided with a special range of goods and services to ensure that their work with Borland products is as productive as possible.

Visit your reseller, or for more information please call

0990 561281

Internet: http://www.borland.com/ • CompuServe: GO BORLAND

CIRCLE NO. 374

Borland
Making Development Easier

REGISTER NOW!

Borland Developer Roadshow Seminars and Workshops • for more information please call 0990 561281

REGISTER NOW!

DEVELOPMENT JUST GOT EASIER

AS/9544/JP

OneDriver ODBC

NobleNet Inc has launched OneDriver, a generic client-resident ODBC driver which eliminates the need for client machines to have one driver per database used. By centralising database-specific drivers on the server, OneDriver reduces system administration costs. The OneDriver SDK is priced at £3,850 per server (NT/Unix), £150 per client - call Personal Workstation Software on 0171 231 0333 for availability.

Persistent Objects with VB

A VBX version of Gamelon, a persistent object library from California-based Menai Corporation, shipped last month. The product, a follow-up to Menai's existing C/C++ libraries released last year, allows programmers to 'embed' simple object databases into VB code - no more time wasted creating custom file formats! Database features include object nesting and tracking, indexing, and navigation. Call Menai on 001 415 6175730.

Syntropy CASE tool

The *Syntropy* OO design methodology from Object Designers at last sees a CASE tool that offers support for the approach. SyntropyCASE, from tool vendors SoftCASE, currently supports only the 'essential' and 'specification' models, event tables and the data dictionary. A release later this year will complete the functionality. Call SoftCASE on 01202 700415 or Object Designers on 01279 816846 for information.

Last minute Notes

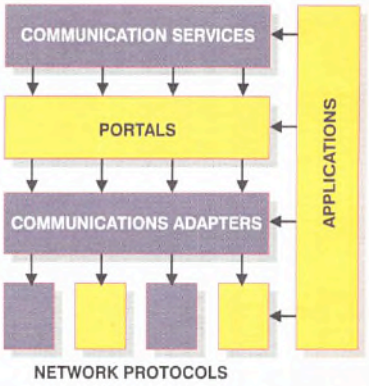
Lotus has announced Notes 4.0, to be available at the end of the year. This new version integrates cc:Mail and Notes and features many performance improvements (field level replication, sliding windows communication protocol, etc). The whole combined product can be controlled by LotusScript, with LotusScript:Data Object focusing on the Notes Object Model. Agents have been added to sort and select. Lotus is on 01784 455445.

Multi-Edit for Windows 7.01

Multi-Edit, American Cybernetics' programmer's editor, has been enhanced in a new version 7.01 to provide many new features, including hex mode editing, rulers, code 'templates', long filename support, background compilation, modeless search/replace dialogs and more. Current users of Multi-Edit 7.0 will automatically receive a free update; otherwise it's available from Rhino Publishing (01302 364861) for £159.

Comms without tears

Rogue Wave Software, publisher of DBtools.h++ (reviewed in June's *EXE*) and Tools.h++, has released a new network communications library for C++. Net.h++, an object-oriented interface to network services, is portable between Unix (SunOS, Solaris and HP/UX), Windows and Windows NT.



Net.h++ is layered into a three-tiered architecture, with each layer built on the other and the top one providing the highest level of abstraction. The C++ developer can write code that is both protocol-independent and cross-platform, and has the option of working directly with the socket classes or programming more abstractly in terms of 'services' (see diagram).

The lowest layer, the *Communication Adapter* layer, encapsulates existing procedural APIs (such as BSD sockets) making the existing functionality easier to use. The middle, or *Portal* layer, is a portable interface to diverse network communications services. Finally, the *Communications Services* layer is the most abstract, and leverages the Portal layer to provide transport-independent services.

Net.h++ 1.0 is distributed in the UK by Hypersoft Europe, tel. 01273 834555. Pricing starts at £399, including source code, although Tools.h++ is a pre-requisite for Net.h++.

Powersoft's new Portfolio

Powersoft Portfolio, a suite of client/server products for workgroup development, was announced by Powersoft. The toolset allows users to design a database with StarDesignor, build a two-tier client server application with PowerBuilder Desktop and deploy the application in a workgroup environment with Watcom SQL Server.

StarDesignor enables the creation and maintenance of databases for 50-odd DMBSs. Reverse-engineering of existing databases is possible, allowing document generation and 'retargeting' to different DBMSs. The tool also supports declarative referential integrity.

PowerBuilder Desktop, combines GUI building tools with an object-oriented programming language. It also offers the 'DataWindow', an object that shields developers from SQL by interfacing directly with the back-end database, and a 'Data Pipeline' features to access and transfer data between heterogeneous databases.

The remaining component of the Portfolio is a three-user version of Powersoft's own RDBMS, Watcom SQL Server, for Windows platforms, OS/2 and NetWare. The suite of products should be available by the time of publication for £329, a saving of about 50% over the cost of purchasing them separately.

Version 5.0 of PowerBuilder, for all current platforms, should appear during the first half of 1996. It will support 3-tier partitioning, OCX controls and native compilation. To find out more about Powersoft's products, try http://www.powersoft.com/.

Phar-out offerings for real-time

Phar Lap Software just released TNT Realtime DOS-Extender and TNT Embedded ToolSuite 8.0, Realtime Edition. The DOS-Extender provides standard realtime features such as priority scheduling and inter-thread communication, while allowing developers to build multi-megabyte, 32-bit protected-mode DOS applications as with the standard version of the DOS extender. Host computer requirements are a PC running MS-DOS, Windows or OS/2; target hardware may include custom design boards, PC 104 systems and standard PCs.

The Embedded ToolSuite is a realtime kernel with features such as threads support and semaphore signalling from Interrupt Service Routines (allowing more code to reside in normal threads rather than in ISRs).

Both tools support 32-bit compilers from Borland, and Microsoft, embedded cros-debugger versions of CodeView and Turbo debugger, and a subset of the Win32 API (over 200 calls). Realtime DOS-Extender is priced at $995; Embedded ToolSuite costs $4995. Non-real-time versions of both tool sets are available at a lower cost. Call Hitex on 01203 692066.

keeping you
ONE DESIGN AHEAD

There are loads of tools for the i960® processor from Intel and other leading companies. There are profiling compilers, operating systems, in-circuit emulators, advanced debuggers, low-cost, fully-functional evaluation kits and many other powerful tools.

Our 16 code-compatible, 32-bit processors give you real value and include choices with up to 150 MIPS. They're all part of an upgrade path to future innovations, such as our new i960 RP processor, a complete I/O subsystem on a chip that unlocks the capabilities of PCI.

http://www.intel.com/embedded/i960/
To learn more about the i960 processor, its development tools and our "No Run-Around" technical support, look us up on the Internet. Or call 01793 431155 for a free CD-ROM quote reference MP0844E.

To get the job done faster, choose from our 243-piece tool set.

intel®

CIRCLE NO. 375

Letters

We welcome short letters on any subject that is relevant to software development. Please write to: The Editor, EXE Magazine, St. Giles House, 50 Poland Street, London W1V 4AX or email **editorial@dotexe.demon.co.uk** Unless your letter is marked 'not for publication', it will be considered for inclusion. Letters may be edited.

Do What I Mean

Dear Sir,
Robert Sproat (EXE letters, October '95) regrets that Francis Glassborow used a bad example of code, but then proceeds to suggest a change that would make matters even worse.

He says that K&R's decision to use different operators for assignment and equality was 'one of the all-time great language design gaffes', and 'there's no sensible reason for not having a dual purpose assignment/equality ='. I'm afraid, unless you are going to throw C's design philosophy out the window, there is a very good reason why a dual-purpose operator is impossible!

The C language definition says (if not in these exact words) 'a statement may consist of an expression', 'an expression is an operator and its operands', and 'an operand may consist of an expression'. From this, a C compiler is unable to deduce semantics from syntax, as the following code snippet shows:

```
a=3;
x=4;
y=5;
a=x=y;
```

If we have a dual-purpose operator, what is the final value of `a`? Is it 3, FALSE, or 5? For that matter, is it anything at all, given that the first three statements could be equality tests! After all '`3==4;`' is a valid, if useless, C statement. The warning about unintended assignment is saying 'what you've done is legal; but do you really mean it?' It is not the job of the compiler to second-guess the programmer and assume he did not mean what he typed.

Where K&R made their mistake was to use two symbols that could easily be transposed by accident. They should have used, for example, `=` and `EQ`. And these aren't the only operators to suffer this problem. What about:

```
a=5;
a=-1; /* no white space */
```

Is the final value of a 4, or -1? Both are valid interpretations of the standard (although that is ANSI's fault, not K&R's).

By the way, I use a language with C-like syntax, and the dual purpose `=`. The only way I can save the result of an equality test is to use the alternative `EQ` operator, as assignment takes priority over equality if confusion might arise.

Anthony Youngman
Gravesend

In functional languages, such as LISP, the different semantic of the assignment and the equivalence is obvious and made clearer syntactically.

Free lunch

Dear Sir,
I'm using quite a few freeware tools developed by the Free Software Foundation. I have recompiled them under DOS. My machine is only a 386 with 4 MB of RAM and a 120 MB hard disk. So no hope of running all the wonderful free Un*x systems at full speed. I was wondering if you were aware of any free alternative to DOS?

John Novik
Internet address supplied

First, you can run a full Unix system on such a system: see the Linux review p. 63. On the point raised in your letter, you'll be pleased to know that there is such a project, called *Free-DOS*, currently under development.

To quote the FAQ: 'Free-DOS wouldn't be much good if it were not compatible with MS-DOS. While there may be some subtle differences in commands between Free-DOS and MS-DOS, these are intended as enhancements, and should not affect most users. Certainly, you might have to tweak a batch file once in a while, but the DOS/NT kernel being used by Free-DOS is planned to be 100% compatible with MS-DOS at the most fundamental level.'

All files regarding Free-DOS (status report, manifesto, coding standards...) can be downloaded by ftp at the site *sunsite.unc.edu* from the subdirectories of */pub/micro/pc-stuff/freedos*. A Web page can be accessed at *http://sunsite.unc.edu/pub/micro/pc-stuff/freedos/freedos.html*.

MoonWater

Dear Sir,
With regard to Will Watts' SoapFlake (*EXE* October '95) on MoonWater, can I point out that Ameol, the CIX offline reader (OLR), gives help on disabled menu items and has done for a long while :-)

Steve Palmer (CIX Staff and Ameol author)
Internet address supplied

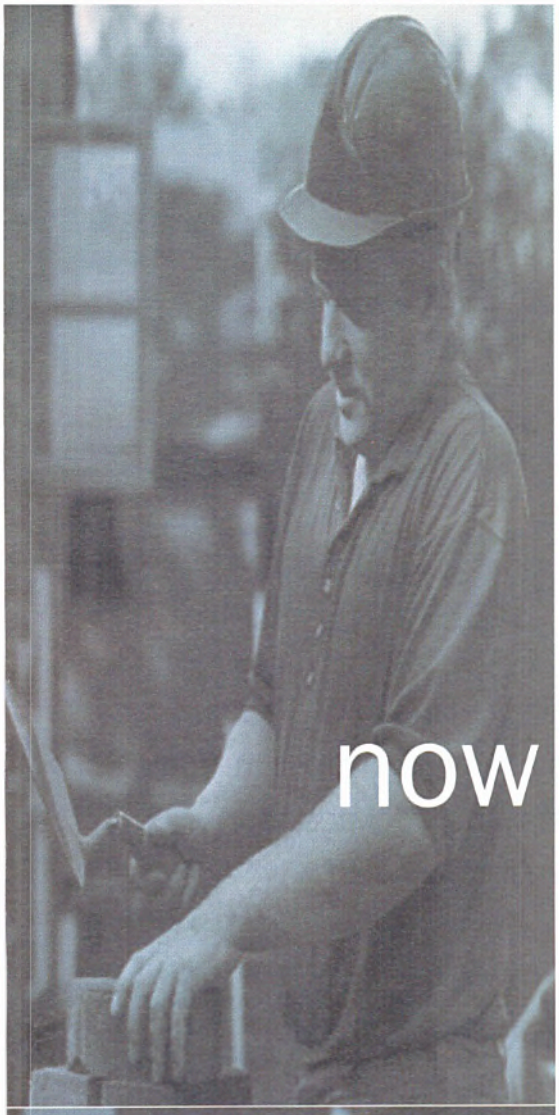
It does give contextual *hints* on greyed menu items but there isn't any contextual *help* on these items, greyed or not (pressing F1 on a menu item doesn't do anything) - Ed (Ameol user).

Data Warehousing

Dear Sir,
What the hell is 'Data Warehousing'? I have asked six people, and have received *seven* mutually incompatible answers. The only common thread seems to be that (a) it's very important and (b) it's very expensive. Can you enlighten me?

Bill Hoyne
Dublin, Ireland

Dunno, beats me - Ed. More seriously, it consists of consolidating all the data of a company in one database. This extra database is organised more logically to respond quickly to most usual queries. It needs to be kept in synch with the live data. It seems a good way to sell more software and hardware to solve a problem residing somewhere else.



Introducing Microsoft® Visual Basic® 4.0. Looks like the high-productivity development tool you know, but now enables teams to create multi-tier, scalable, 16- and 32-bit client-server solutions.

Now, with the Remote Automation technology in the Visual Basic programming system, Enterprise Edition, your development team can quickly build scalable, maintainable applications and deploy them across enterprises big or small – *using a tool they already know.*

now quick and easy tackles large and complex

And with support for Windows® 95, Windows NT,™ and Windows 3.1 operating systems, the new Visual Basic 4.0 Enterprise and Professional Editions let you migrate to 32-bit at your own pace, while preserving your investment in existing code, skills and technology.** Best of all, now you can even create your own open, reusable OLE components such as OLE DLLs and OLE automation servers. So your applications won't just get faster, they'll actually get smarter by design.

To obtain more information about Microsoft Visual Basic 4.0 or any of our other Visual Development Tools, please visit our web site at **http://www.microsoft.com/devonly**.

Enterprise-wide multi-tier solutions*

Partition applications across networks*

High-performance remote data access*

Large-scale team development*

32-bit with built-in 16-bit support

Open, reusable OLE component creation

Open, integrated development environment

New data-bound OLE controls

To order your copy or get details of your nearest Microsoft Reseller, simply call Microsoft Connection on 0345 00 2000. Alternatively, contact one of the following Resellers below:

• Grey Matter Software 01364 654100. • Programmers Paradise 0161 728 4177.

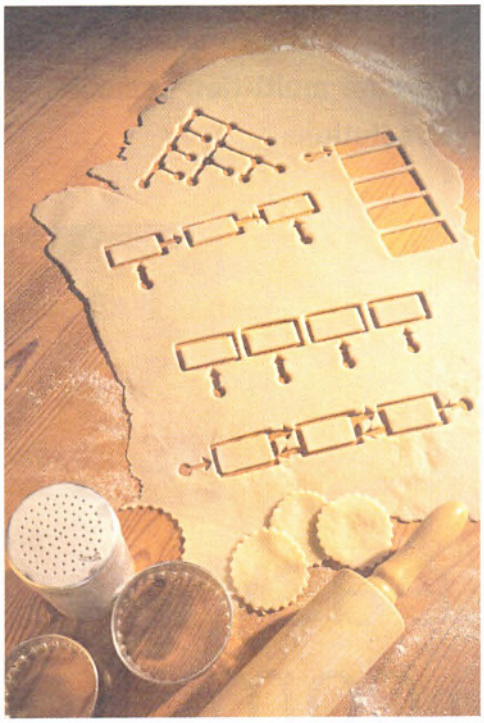
• QBS Software Limited 0181 994 4842. • System Science 0171 833 1022.

✎ CIRCLE NO. 376

Microsoft

WHERE DO YOU WANT TO GO TODAY?™

*Features available only on Enterprise Edition. **32-bit development requires a 32-bit operating system.



Standard Template Library

The last major addition to the ANSI/ISO draft standard for C++ made generic data structures and algorithms part of the language. **Barry Smith** gives you a head start with the STL.

Apart from a robust and functional string class, the first requirement of every C++ developer is a good set of data structure components. Until recently, C++ has lacked a standard for components like dynamic vectors, linked lists, stacks and queues. The commercial libraries that have emerged to fill the gap have struggled with some of the fundamental tensions underlying good object-oriented design in C++.

The Standard Template Library (STL) is an attempt to address and resolve these tensions. Originally written by Alexander Stepanov and Meng Lee at Hewlett-Packard's research laboratories in Palo Alto, California, the STL has made dramatic strides over the last year in its acceptance by the C++ community.

As the last major change to the current C++ draft standard, the STL was accepted as part of the standard library in July 1994. The original implementation by Stepanov and Lee has been augmented by a number of other free and commercial versions. Few compilers exist that do not support any STL implementation, and most compiler vendors are planning to provide built-in STL support in the near future. Vendors of existing class libraries, such as Rogue Wave, have announced migration paths to make their existing libraries compatible with the STL architecture.

The STL includes several different types of component:

- *containers*, such as vectors and lists, which manage collections of objects;
- *iterators*, providing the means to traverse through;
- *algorithms*, defining generic computing procedures like binary searching;
- *adaptors*, providing interface maps, eg from vector or list to stack;
- *functors* or function objects, encapsulating unary or binary functions;
- *allocators*, encapsulating a memory model and allocation strategy.

The STL attempts to provide a type-safe data structure toolkit that minimises the implementation burden on the developer, and supports the construction of generic and reusable - but efficient - algorithms.

Memory allocation

By avoiding direct use of the C++ `new` and `delete` operators in favour of `allocator` objects that encapsulate all the details of a given memory model and allocation strategy, the STL attempts both to insulate itself from platform-related memory model issues and also, ultimately, to allow memory allocation strategies to be isolated from container and collectable object implementations, rather than hard-wired into them.

STL implementations come with at least a default allocator predefined, and this exposes `allocate` and `deallocate` methods to grab memory from the free store and release it. The Hewlett-Packard reference implementation also includes `far`, `near` and `huge` allocators that work with Borland C++'s 16-bit memory models.

In theory, STL data structure components are supposed to be parameterised with respect to their allocators, so that different memory allocation strategies can be used with the same generic types. For example, an allocator might be defined that allocates objects in a persistent data store; another might use shared memory or Windows NT memory-mapped files; and another, thread-local storage.

For normal purposes, though, you will want to use the default allocator, which in effect maps allocations to the normal `new` and `delete` operators. The current C++ standard mandates a new language feature: default template parameters. Until compiler vendors implement this feature, the huge

```
class Object {
public:
    // ...
    virtual int compareTo(Object* pObj) = 0;
};

class A : public Object {
public:
    // ...
    virtual int compareTo(Object* pObj) { /* ... */ }
};

class B : public A {
public:
    // ...
    virtual int compareTo(Object* pObj)
        { return getNumber() - ((B*)pObj)->getNumber(); }
private:
    int getNumber();
};
```

Listing 1 - Beware of unsafe casting

potential of STL allocator objects will remain unrealised.

Type safety and polymorphism

One strategy for building a C++ data structure toolset is to maximise flexibility and reuse by implementing polymorphic *intrusive* collection classes and matching iterator classes. This typically involves heavy use of virtual function dispatch to manage collections of objects all derived from a single *cosmic object* class. Users of Smalltalk or of the NIHCL C++ library will recognise class `Object`. MFC users will be familiar with their own cosmic object class, `CObject`, and Rogue Wave users with another: `RWCollectable`. Let's call this the polymorphic or 'Smalltalk' strategy.

The problems with this approach revolve around compatibility, type safety, and - to a lesser extent - performance. Performance can suffer because of the need to rely extensively on virtual functions, although in many cases this is not an insurmountable problem. More serious is the lack of common ground between competing cosmic object classes: you have to build your own business objects around one or other architecture and forgo use of the other libraries.

Lack of type-safety, though, is the key problem. It forces the developer to cast objects to and from pointers or references to the base collectable types.

Heavy use of virtual functions within the collectable classes themselves can lead to very unsafe casting, as the typical example (based on the design of a commercially available library) in Listing 1 shows. The intention in this example is to allow each derived collectable class to implement its own notion of comparison, and so allow the creation of generic sorting and searching code within the architecture.

Unfortunately, to do this, the `compareTo()` override will almost certainly have to downcast the supplied `Object*`. What happens now depends on the exact run-time type of the operands. If the left operand (message receiver) is an `A` and the right operand (message argument) is a `B`, the comparison may well be safe, but it will ignore the `B` implementation of `compareTo()` altogether, and possibly return a nonsensical result. If the left-hand operand is a `B` and the right-hand operand an `A`, the invalid downcast to `B*` in `B::compareTo(Object*)` will probably result in a memory protection fault. To contain errors of this type, it's necessary to use extensive run-time type checking and type switching, or else be very certain that the operands will be of exactly the same type.

The alternative to the polymorphic strategy is the type-safe strategy. The obvious type-safe approach is to code each data structure and collection class afresh for each new collectable type, along with all the associated iterators and functions to implement the various algorithms, such as searching or sorting, which will be required for that class. Fortunately, it's possible to use C++ templates to eliminate much of the tedium involved in doing this (and with care even the code bloat can be addressed!).

This is the course followed by the STL, and also by a few other libraries. The STL includes template classes for three types of ordered collection: `vector`, doubly-linked `list`, and `deque` (a double-ended queue structure optimised for efficient pushing and popping at both ends). There are also associative container templates providing efficient key-based access: `set` and `map` for unique key collections and `multiset` and `multimap` for non-unique key collections.

Although there was a last-minute proposal from a group of contributors, including STL researcher David Musser, hashed collections are not officially part of the standard. There is however a draft hash table specification and a pilot implementation of hashed collections consistent with the STL architecture.

The STL fragment in Listing 2 shows a vector of integers, initially three elements in size, allocated on the program stack. With minor differences relating to the properties of the particular data structure, the interface for `list` and `deque` matches the `vector` interface. Here, vector-specific array indexing is used to initialise the initial three elements, but standard methods equally applicable to `lists` and `deques` are used to test and display various characteristics of the collection, expand the vector dynamically, access the first and last elements, and remove items.

Apart from the fundamental container types, other data structures are emulated in STL by the use of container *adaptor* template classes.

```
#include <stl.h>
int main()
{
    vector<int> vec(3);
    vec[0] = 2;
    vec[1] = 4;
    vec[2] = 6;
    assert(! vec.empty());
    assert(vec.size()==3);
    cout << "first item = " << vec.front () << endl;
    cout << "last item = " << vec.back () << endl;
    vec.push_back(8); // grow dynamically
    assert(vec.size()==4);
    cout << "last item = " << vec.back () << endl;
    vec.pop_back ();
    assert(vec.size()==3);
    cout << "last item = " << vec.back () << endl;
    return 0;
}
```

Listing 2 -Vector of integers allocated on the stack

```
stack<list<int> > s;
s.push(99);
s.push(66);
s.push(33);
s.push(0);
while (!s.empty())
{
    cout << s.top() << endl;
    s.pop();
}
```

Listing 3 - A stack based on a linked list

Thus, it's possible to define stacks, queues and priority queues based on `vectors`, `lists` or `deques`. Listing 3 shows a stack based on a linked list. The container adaptor defines an interface mapping for the stack-specific `push()`, `pop()`, `top()`, `size()` and `empty()` operations in terms of whichever underlying type is used to implement the stack, in this case a `list`.

Iterators

One of the key ideas behind the STL is a generalisation of C++ pointer semantics: the notion of an *iterator* object. STL iterators represent a 'position' in an abstract data structure. STL containers all expose a typedef, `iterator`, which is an alias for the iterator type of an iterator object suitable for traversing that container, and another typedef, `value_type`, which is the type of the values contained within the data structure.

STL iterators may or may not be pointers (actually they usually are, but you don't need to know this). Like a C file pointer, which may point 'one past the end' of the file, an iterator may point one past the end of an STL container. STL collections all provide a `begin()` method, returning an iterator pointing to the first item in the collection, and an `end()` method, returning a past-the-end iterator.

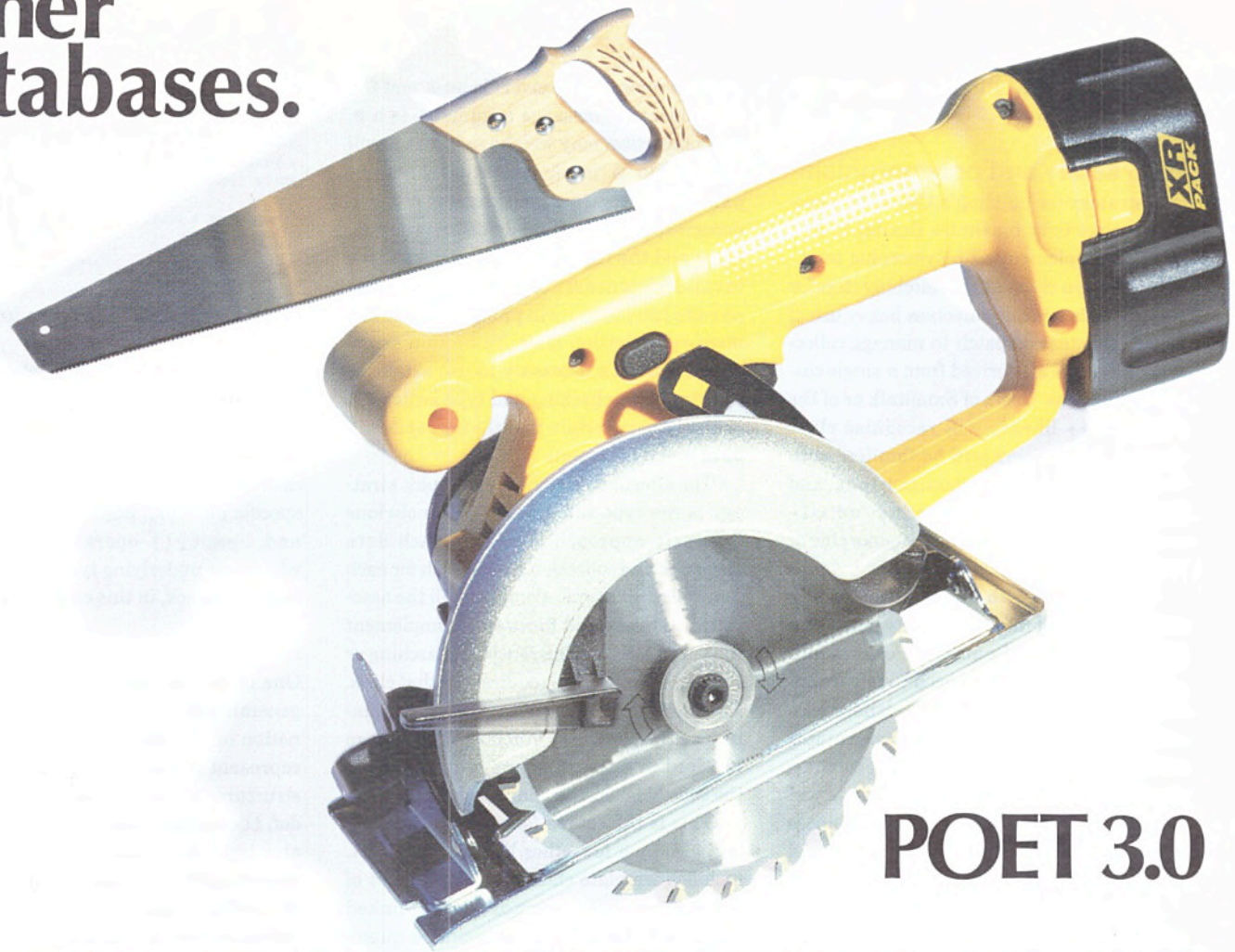
Like normal pointers, STL iterators can be advanced using `operator++()`, and compared using `operator==()`. STL containers guarantee that iterators returned by `begin()`, and subsequently advanced with `operator++()`, can be dereferenced to yield an object convertible to the collection's `value_type`, so long as the iterator never tests equal to the past-the-end iterator.

Most STL algorithms are defined in terms of ranges of objects specified using pairs of iterators, where the first iterator is the starting point, and the second is an end-point iterator reachable from the first iterator by the repeated application of `operator++()`.

Iteration over any STL container, `c`, can be carried out in this way (using `vector<int>` as an example):

```
vector<int> c;
// ...
vector<int>::iterator i = c.begin();
vector<int>::iterator e = c.end();
while (i != e)
```

Other Databases.



POET 3.0

Finally. The Object Database That Delivers More Platforms, More Features and More Performance Than Any Other Database.

It's easier when you choose the right tool for the job. Pick POET 3.0, the third generation of the world's best-selling object database. POET 3.0 is tightly integrated with C++, accelerating development and simplifying software maintenance.

More Platforms

POET 3.0 transparently networks PCs, Macs and workstations. Scale from a laptop to a workgroup to an enterprise server effortlessly. And only POET 3.0 supports all these operating systems:

- Windows, Windows for Workgroups, Windows NT and Windows 95
- Macintosh OS • Novell NetWare
- OS/2 • UNIX • NextStep

More Features

POET 3.0 includes POET Developer, an integrated, GUI-based workbench. Plus a toolset of system services, so you can focus on building your application.

Integrate with applications like Word and Excel using OLE 2.0. Create SQL-like queries with POET's OQL.

And with the ODBC driver, you can combine POET databases into existing environments.

More Performance

POET 3.0 offers the power of object databases and the reliability of relational systems. You get the POET Administrator, an industrial strength database manager that includes user authorization, online backup and transaction rollback.

POET 3.0's object server architecture maximizes performance and eliminates processor dependencies. And POET 3.0's unique navigation abilities speed client responsiveness.

There's no better object database tool than POET 3.0. For a limited time, get the POET 3.0 Personal Edition for just £149*. And cut the toughest object database problem down to size in no time.

POET 3.0 PERSONAL EDITION FOR £149

CALL 0181-317-7777

POET 3.0 is Distributed in the UK by:
Silicon River Limited.
106-108 Powis Street,
London
SE18 6LU.
Telephone: 0181-317-7777 Fax: 0181-316-7778

CIRCLE NO. 377

*Excludes Delivery & V.A.T.

POET Software
More Platforms. More Features. More Performance.

1992 BYTE AWARD OF DISTINCTION

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1993 GOLD

OBJECT DATABASE ODMG MANAGEMENT GROUP

```
{
    cout << *i << '\t' << endl;
    i++;
}
```

Compare this with iteration over a C array:

```
typedef int* iterator;
int c[100];
// ...
iterator i = c;
iterator e = c+100; // one past the end
while ( i != e)
{
  cout << *i << '\t' << endl;
  i++;
}
```

Apart from the method of getting the begin iterator and the one-past-the-end iterator, the code is identical.

The STL classifies iterator objects into a 'requirements hierarchy'. This is different to an inheritance hierarchy: it is just a list of operations (and their associated semantics) which must be supported by each kind of iterator if it is to be used with an algorithm that requires an iterator of that type. The algorithms themselves are templatised with respect to the various iterator types, to allow the STL to supply the optimal implementation for a given iterator type.

Input & output iterators

The simplest kind of iterator is an *input* iterator. It only supports single-pass iteration over the underlying container. Input iterators must support dereferencing to yield a reference to the underlying type, but assignment to the dereferenced iterator need not be supported. An input iterator must support pre- and post-increment operators as per normal C++ pointer semantics. The STL also specifies that the equality and not-equal operators must be available for comparing input iterators, that a copy constructor and assignment operator be available, and that a copy of an iterator tests equal to the original.

Normal pointers used to read from C - style arrays can act as input iterators. The STL also supplies a class `istream_iterator` that implements input iterator semantics for input streams.

Output iterators are essentially like input iterators except that they support *only* assignment to the dereferenced iterator. They need not necessarily support the equality operator, and dereferencing and assignment must occur at each point before incrementing. This odd requirement is added so that the STL can provide the `ostream_iterator` class to provide output iterator semantics for output streams.

```
int vec[100];
int *iend = vec+100; // one-past-the-end
 // ...
int *ret = find(vec,iend,99);
if (ret == iend)
  cout << "search failed" << endl;
else
  cout << "found it!" << endl;
```

Listing 4 - Search a C array of ints for the number 99

Other iterators

Forward iterators relax the requirements on input and output iterators to allow multi-pass algorithms, but only in the forward direction.

In addition to incrementing, and the other requirements of forward iterators, *bidirectional* iterators support pre- and post-decrement operations. A bidirectional iterator is the native type of iterator provided by STL `list` containers (STL linked lists are doubly linked).

In addition to the requirements of bidirectional iterators, *random access* iterators support general increment and decrement operations in the form of `operator+=()` and `operator-=()`. To support testing iterators against beginning and past-the-end values after non-unitary increments, random access iterators add the requirement to support all the inequality operators: `<`, `>`, `<=` and `>=`. A random access iterator are the native iterator type supplied by `vectors` and `deques`.

Generic algorithms

The real innovation of the STL is not simply that it offers a flexible data structure toolkit and iterators for traversing the data structures. It is that it supports the implementation of efficient algorithms that are generic across well-defined subsets of those data structures, without the need for the intrusive derivation of collectable objects from common ancestor classes. STL generic algorithms rely on the template function language feature.

To understand the idea behind generic algorithms, consider the very simple, but common, use of a template function to replace the old C `max` macro:

```
#define max(x,y) (((x)<(y)) ? (y) : (x))
```

In place of the macro, we may write something like:

```
template <class T> inline
const T& max(const T& a, const T& b)
{
    return a < b ? b : a;
}
```

Apart from getting over the side-effect bugs that can occur as a result of repeated evaluation of the macro's arguments, what does this buy us? It adds type safety because it ensures that the two arguments are of the same type or that implicit conversions are valid. At the same time, inlining the template function avoids any function-call overhead; it's no longer than the original macro, and will not generate additional code. But best of all, it is completely generic across all types - built-in or user defined - that meet a simple requirement. In this case, the requirement is that the type has a meaningful definition of `operator<()`.

This is the fundamental idea behind STL: that if you define a standard set of requirements, then you can implement algorithms in template form that will work generically across all types that meet that set of requirements. And in fact, the implementation of `max()` above comes from the set of algorithms supplied with STL.

The bulk of the STL armoury of algorithms operates on sequences of objects within STL data structures. In order to make the algorithms generic, that is, independent of the type of the data structures, these algorithms are defined in terms of the iterator abstraction, and the algorithm requirements are specified using the iterator requirements hierarchy outlined above.

Here is an (almost trivial) example of a generic algorithm implementing a simple linear search. It is templatised in terms of `Iter`, which may be any iterator type satisfying the requirements of an input iterator (this would include any bidirectional or random access iterator), and `T`, which is the type of the values contained within the data structure:

```
template<class Iter, class T>
Iter find(Iter ibegin, Iter iend, const
T& target)
{
   Iter i = ibegin;
   while (i != iend && *i != target) ++i;
   return i;
}
```

We don't test `(i < iend)` in the `while` expression, because that would restrict the algorithm to random access iterators only; the not-equal test applies to all iterators apart from output iterators and is thus more generic. Notice also the STL convention that we indicate failure by returning the past-the-end iterator.

Having defined our `find()` algorithm (actually, it's one of around seventy algorithms pre-defined in the STL), we can use it with almost any data structure which can be traversed. The code in Listing 4 searches a C array of integers for the number 99.

STL's stock of algorithms runs from the simple `min()`, `max()` and `find()` illustrated in this article, through to much more complex algorithms, such as a those implementing set algebra on sorted data structures: `set_symmetric_difference()`, `set_difference()`, `set_union()` and `set_intersection()`.

Some STL algorithms perform boolean tests or other arbitrary operations on the elements over which they iterate. This is achieved by implementing them as tem-

plates with respect to another STL abstraction: function objects.

Function objects

Function objects, or *functors*, define an `operator()`. STL function objects are either unary or binary (that is, they are defined to accept one argument or to accept two). Just as iterators are generalisations of the C++ pointer, so functors are generalisations of the C++ function pointer. The STL includes numerous pre-defined functors, but you can easily add your own, or use ordinary pointers to C++ functions. One example is the binary functor `plus`, which is defined like this:

```
template <class T>
struct plus : binary_function<T, T, T>
{
    T operator()(const T& x, const T& y)
                 const {return x+y;}
};
```

The `plus` functor is derived from the `binary_function` base class, which takes three template arguments: the first and second argument types and the return type. For `plus`, these are all defined to be the same type, but they need not be for other functors.

The `plus` class implements one method, binary `operator()`.

To see how we might use this innocent looking object, we'll look at `transform()`, a useful STL algorithm that takes two input iterators representing a start and an end point for some data structure, an output iterator representing the start of a data structure into which the results will be written, and either a unary or a binary functor defining the transformation to apply to the input values.

If the functor is binary, `transform` expects a second input iterator pointing to the start of a second input data structure that is assumed to contain at least as many elements as the first. The binary `transform()` algorithm passes the corresponding pairs of values from the two input iterators to the binary functor and stores the functor's return value using the output iterator. The signature of the binary version of `transform()` is like this:

```
template <class InputIter,
    class OutputIter, class BinaryOp>
OutputIter transform
    (InputIter ibegin1,
     InputIter iend1,
     InputIter ibegin2,
     OutputIter obegin,
     BinaryOp op);
```

Suppose we had two data structures `x` and `y` containing `doubles`, and a third data structure `z` pre-allocated to accept as many `double` results. We could then take the sum of the corresponding values in `x` and `y`, and store the result in the data structure `z` using the binary functor `plus<double>`:

```
transform(x.begin(),x.end(),y.begin(),z.
begin(),plus<double>);
```

This one line illustrates the power of the STL. It replaces some kind of `while` or `for` loop with a much more succinct piece of code, and it does so without compromising efficiency because the code is implemented using inlined templates. What is most striking is just how generic the algorithm is: the input and output data structure types and the binary function object can all vary arbitrarily, and the same code will work unchanged.

To get you started, the draft ANSI/ISO C++ standard can be retrieved electronically, and implementations of the STL already exist for many compilers. Check out the URLs given in the box alongside. ■

Barry Smith is a consultant specialising in object-oriented software design and development using C++. He can be reached by email as bksmith@cix.compulink.co.uk.

STL information and product guide

ANSI/ISO

The draft ANSI/ISO C++ standard is now publicly available, and this includes a formal specification of the entire standard library, including the STL. It is however less readable than the original STL documentation from Hewlett-Packard, written by Stepanov & Lee (see below). The preferred UK location to get the C++ standard is ftp://ftp.maths.warwick.ac.uk/pub/c++/std/WP; if that site is busy you can get it from ftp://research.att.com/dist/c++std/WP. The WP directory contains one set of .PS files (PostScript for dumping to the printer) and another set of Adobe Acrobat .PDF files. There is a README describing what files to download, it also contains information about valid usage. 'Not for distribution' means that it's not for you to print, bind and (for example) sell. Reading it to improve your own understanding of C++ is specifically allowed.

Hewlett-Packard

The Hewlett-Packard reference implementation of the STL, including documentation in PostScript format, is available at *ftp://butler.hpl.hp.com/stl* and at *ftp://ftp.cs.rpi.edu/pub/stl*. The *ftp.cs.rpi.edu* site is also the best place to look for the STL-like hash table implementations by David Musser et al. The STL documentation set includes *The Standard Template Library*, by Alexander Stepanov & Meng Lee, which is probably the best introduction and reference to the STL.

The H-P source code is copyrighted, but freely redistributable both privately and commercially with few legal constraints. It works unchanged with both Borland C++ 4.5 and 4.51 (except that you may need to disable Borland's `min` and `max` template functions by declaring the appropriate manifest constant). Parts of the library will build and work with Symantec C++ 7, but some of the source files need modification. Many other compilers will work with this implementation. Microsoft Visual C++ supports (and ships with) the STL as of version 4.0.

GNU

The GNU C++ library, *libg++ 2.7.0*, contains implementations of several parts of the new ANSI Standard C++ library including a version of the H-P STL source modified so that the GNU C/C++ compiler can build it. Currently, some of the modifications have sacrificed efficiency for compatibility with the compiler. But GNU support for STL should improve over time: the GCC team are actively modifying the compiler to take account of the advanced template code in STL. To download the GNU library code, look for libg++-2.7.0a.tar.gz at *ftp://src.doc.ic.ac.uk/gnu*.

Rogue Wave

Rogue's complete standard library implementation is due to ship very soon. A new version of *Tools.h++*, version 7.0, modified to integrate closely with STL and the rest of the standard library, is due in around January 1996. UK contact: Hypersoft Europe (01273 834555).

ObjectSpace

ObjectSpace sell *STL<ToolKit>* and *Systems<ToolKit>*, both of which provide supersets of STL that are portable across a large number of platforms and compilers. ObjectSpace's STL implementation is compatible with all the major PC and Unix C++ development environments, including Microsoft's 16- and 32-bit compilers, and even Unix cfront-based compilers. UK contact: Power Software (01224 622 202).

Modena

Another commercial implementation is *STL++*, by Modena Software. Modena does not appear to be active in Europe at this stage.

Web sites

There are a number of WWW sites with STL-related material. The best is maintained by David Musser at *http://www.cs.rpi.edu/~musser/stl.html*.
Also, look at: *http://www.xraylith.wisc.edu/~khan/software/stl/STL.newbie.html*, *http://www.objectspace.com* and *http://www.roguewave.com*.

Productivity Headache #69: Check-list items that don't check out.

When selecting a tool, you make sure it has all the "Hot Items" on your check-list. But how many will turn out to be not-so-hot?

Get a Codewright Programmer's Editor.

Take Templates. Not C++ templates, but "Syntax Expansion", "Construct Forms". Call them what you will, every decent programmer's editor since BRIEF has them. You type in "for" and the editor inserts the basics of a loop construct. Templates are a check-off item. But how many programmers actually use them?

Most programmers try to use templates, but turn them off after finding they get in the way. The templates put braces in the wrong place, or don't follow your style guide. If you can't change templates easily, you'll be more productive just turning them off.

The Cure:

Codewright EDITORS

Codewright fusion for Microsoft VISUAL C++

Codewright Professional Version 3.1

Imagine Templates that save you time.

Codewright Templates take the drudgery out of typing in language constructs. Codewright lets you quickly select from the most common styles of brace placement. If that's not enough, you can easily change the template definition on-the-fly. No need to learn a macro language, or compile anything. Now that's the kind of templates you can use!

Turn Templates into Power Templates.

Codewright Templates are not just usable. They have the power to do real work, too. They can pause to prompt you for a function name, insert the current date or some boilerplate from a file, even paste the contents of an environment variable. You can even execute any of 1000+ functions, or your own function from within a template.

Features that won't burst your bubble.

Codewright templates are an exclusive feature of Codewright Editors: Codewright Fusion, the editor that is a drop-in replacement for the editor in the Visual C++ environment; and Codewright Professional, the comprehensive editing solution.

For more information about this feature and other ways the Codewright Editors will make you more productive, call now.

Single-user License:

Codewright Fusion	Codewright Professional
149USD	**229USD**

Prices are plus shipping.

Just one of the many productivity solutions built 'wright in.

premia®
Premium Quality Software

Premia Corporation
1075 NW Murray Blvd., Suite 268
Portland, Oregon 97229 USA
Email: sales@premia.com Fax: +1-503-641-6001 Phone: +1-503-641-6000

In the U.K. contact:

Grey Matter, Devon
+(364) 65 41 00

Software Plus, Cheshire
+(928) 57 97 00

System Science, London
+(718) 33 10 22

A New Language for an Old problem

Is the class concept any good for algorithms that are older than most C++ programmers? **Leendert Ammeraal** finds out and, in the process, tells you everything about dynamic programming.

When I was asked to write an article related to C++, I found it difficult to make a choice. I did not like the idea of simply explaining some elements of the language, quoting my book *C++ for Programmers* and annoying you with things you already know. It seemed to me a better idea to take some application that not all of you may be familiar with. However, I then realised that, since C++ is such a general language, finding a program that illustrates the use of C++ is as strange a task as finding a story that illustrates the use of a text processor. Since I am now writing another book about algorithms and data structures, based on C++, I decided, after consulting my publisher, to borrow and adapt a section of it. I then looked in my manuscript for an interesting algorithm, the code of which was compact enough to be included here as a complete, ready-to-run program.

Dynamic programming

After wasting much time deliberating, I took an example that also illustrates the use of *dynamic programming*. It is my experience that many programmers are intrigued by this term because they have never heard it, and because the word 'dynamic' sounds promising and possibly revolutionary. If this also applies to you, I am afraid I must disappoint you.

Dynamic programming is not exactly a new way of programming but rather, like linear programming, a well-known subject of operational research which is in turn a branch of applied mathematics. If you have enough scientific interest and plenty of time, you can find out all about it in R. E. Bellman's classic book *Dynamic Programming*, published as early as 1957. The terms 'linear programming' and 'dynamic programming' are used more frequently by mathematically-oriented economists than by programmers. They are both optimisation techniques, used to maximise profits or to minimise waste, and so on.

The problem

This article only deals with the simplest possible dynamic programming problem, and, unlike Bellman's book, it is *not* primarily intended for mathematicians. You may find the program discussed here useful for various practical purposes, all related simply to adding up some non-negative integers. For example, suppose that, in some currency, we had plenty of coins with values of 1c, 5c, 10c, 25c, and we had to pay 42c. We would then immediately compute

$$42 = 25 + 10 + 5 + 1 + 1$$

if we wanted to use as few coins as possible.

With the coin values mentioned, we can obtain the optimal solution by using a *greedy* algorithm. This word indicates that, each time, we simply use the largest coin that does not exceed the remaining amount to be paid. However, a greedy algorithm would not give an optimal solution (ie one with as few coins as possible) if no coins of 5c are available. In this case, a greedy algorithm would give the solution:

$$42 = 25 + 10 + 1 + 1 + 1 + 1 + 1 + 1 + 1$$

with a total of nine terms. But there are only six terms in the optimal solution:

$$42 = 10 + 10 + 10 + 10 + 1 + 1$$

The wise lesson to be learned from this is that greed is not always rewarded with the best results. Instead of the last solution, we can write

$$42 = 4 \times 10 + 2 \times 1$$

simply as an abbreviation, *still counting the original terms*. So even in this abbreviated form we will say that there are six terms, not two.

Since our problem is not only about coins, we can better express it in the following more general form. A non-negative integer sum s is given, along with n positive integers:

$$d_0, d_1, ..., d_{n-1}$$

Each integer d_i may occur more than once as a term, and the sum of all terms must be s. We insist on using as few terms as possible in forming this sum.

We will use the following, slightly simpler, example in our discussion of the dynamic programming algorithm:

$$s = 18;\ n = 3;\ d_0 = 7, d_1 = 6, d_2 = 2$$

Again, the optimal solution

$$18 = 3 \times 6$$

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

x	$F(x)$	Selected terms
0	0	
1	*undefined*	
2	1	2
3	*undefined*	
4	2	2 + 2
5	*undefined*	
6	1	6
7	1	7
8	2	6 + 2
9	2	7 + 2
10	3	6 + 2 + 2
11	3	7 + 2 + 2
12	2	6 + 6
13	2	7 + 6
14	2	7 + 7
15	3	7 + 6 + 2
16	3	7 + 7 + 2
17	4	7 + 6 + 2 + 2
18	3	6 + 6 + 6

is different from the one we would have obtained had we been greedy, starting with the largest value, 7. In that case we would have found the solution

$$18 = 2 \times 7 + 2 \times 2$$

which is not optimal, because it consists of four terms compared with three in our optimal solution.

So much for the problem we have to solve. You may wonder what all this has to do with C++, but before discussing a program that solves this problem, we had better pay some attention to the idea behind it.

The algorithm

The technique of *dynamic programming*, which solves our problem very efficiently, is based on using a table (see above). Explaining this important technique in full generality would be beyond the scope of this article, so we will discuss it only in connection with our problem, using as little mathematical notation and terminology as possible. We define function $F(x)$ to be the minimum number of terms we need to form the sum x, using only the available integers as terms. For example, we have

$$F(18) = 3$$

in our last problem because we need at least three terms (6, 6, 6) to form the sum 18. The dynamic programming approach implies that we also solve the problem also for values smaller than the given sum s (18 in this example). For all values $x = 0, 1, 2, ..., 18$, in that order, we try to find $F(x)$, and we store these values in a table. Function $F(x)$ is defined only for those values of x for which there is a solution. There is one for $x = 0$:

$$F(0) = 0$$

because a sum of zero terms is zero by definition. With 7, 6 and 2 as the only possible terms, we cannot form a sum 1, so we will write:

$$F(1) = \textit{undefined}$$

Then follows the first value ($x = 2$) for which $F(x)$ is defined and non-zero. We have

$$F(2) = 1$$

because we can form the sum 2 as the single term 2. Before discussing how to find all the other F values we need, let us have a look at the table to the left in which they occur. (Recall that it is based on $s = 18$, $n = 3$, $d_0 = 7$, $d_1 = 6$ and $d_2 = 2$.) Besides the values $F(0)$, ..., $F(18)$, we also store the corresponding sets of terms. You can see left how this table is built.

Except for $F(0)$, each of the values $F(x)$, along with the corresponding selected terms, is found by using some previous values $F(u)$, where u is less than x. To see how this works, let's assume that we have advanced to some point in the middle. For example, suppose we want to compute $F(x)$ for $x = 12$, and that we have already stored in the table all $F(u)$ values for all values u less than 12. Then we can find $F(x)$ by selecting the smallest of the values $F(x - d_0)$, ..., $F(x - d_{n-1})$. If this smallest value occurs more than once, we take the first we encounter, although we could have taken any. After all, there may be several optimal solutions. In our example, this implies that we have to select the minimum of these three values $F(u)$, at least as far as they are defined:

$$F(12 - 7) = F(5) = \textit{undefined}$$
$$F(12 - 6) = F(6) = 1$$
$$F(12 - 2) = F(10) = 3$$

Since $F(5)$ is undefined, we only choose between $F(6)$ and $F(10)$, identifying $F(6)$ as the smaller value. As we know, $F(6) = 1$ means that only one term (6) is required to form the sum 6. The second of the three lines above tells us that we only have to add one term, 6, to obtain the optimal solution for $x = 12$. In other words, we find

$$F(12) = F(12 - 6) + 1 = 2$$

and the sum in question is 12 = 6 + 6. This may seem very trivial because of the low value of x, but it is not with higher x values. If all values $F(x - d_0)$, ..., $F(x - d_{n-1})$ are undefined, then $F(x)$ is also undefined. Our algorithm determines in this way that $F(1)$, $F(3)$ and $F(5)$ are undefined in our particular problem. To find $F(1)$, we would have to select the minimum of the three values $F(1 - 7)$, $F(1 - 6)$ and $F(1 - 2)$, if this were possible. But it is not, since $F(x)$ is undefined for negative x values. Consequently, $F(1)$ is also undefined.

Dynamic programming is a 'bottom-up' approach: we try to compute $F(1)$, $F(2)$, ..., $F(s)$, in that order. The resulting program would be very simple if only the value $F(s)$ were needed, since then we would only have to use an array of integers to store the computed values $F(x)$. As we are interested in the selected values d_i, we also have to store these for each value x for which $F(x)$ is defined.

Implementation in C++

Program *dynpro* (see Listing 1), shows my implementation of the algorithm. I have used a class, also called `dynpro`, which enables me to have all the work done by these three statements in the `main` function:

```
dynpro DP;
DP.solve();
DP.print();
```

The program has no limitations other than those imposed by the environment, and I would be ashamed if it had. For example, memory for the table is allocated dynamically after it is known how much is actually required. The constructor, called in the first of the above three-line program, performs all input operations. I could have used another member function, say `setup ()`, for this purpose, omitting the constructor, but since there is a destructor, it is more logical to have a constructor as well. I could, of course, also have omitted the destructor, because de-allocation of memory is done

Why gamble on your database design?

"Finally, after 20 years, there is a database design tool with real, practical usefulness"

Richard Finkelstein
Performance Computing

"InfoModeler is the ideal business analysis and modelling tool. It is fast, productive and easy to use.
Put simply, it helps me do a better job"

Colin Inglis
IS Controller, Adams Childrenswear

"InfoModeler's sound theoretical underpinnings make it ideal for the teaching of database theory and for practical system design"

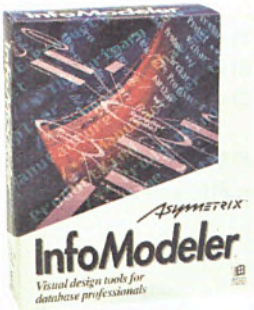
Graham Welford
Lecturer in Information Systems
University of Wolverhampton

Now there is a tool for business analysts and system designers that can transform the way you think about database systems.

It's a design toolkit that helps make sure your database applications truely reflect business requirements.

InfoModeler takes the guesswork out of business analysis and database design. It starts from basic principles - facts about the business process. Facts are expressed in terms everyone can understand - plain English sentences. From the 'factbase' you assemble InfoModeler automatically generates the database model. InfoModeler uses examples, again expressed in English, to validate your model, checking that the database design conforms to sound relational principles.

So there's no more guesswork.



at £99* there's no need to gamble!

Because InfoModeler automates routine, but essential, tasks such as normalisation, table generation and system documentation, development is faster, errors are fewer, maintenance is easier and a lot less expensive.

And at only £99* for the desktop version, InfoModeler is better value than ever.

Now, by using InfoModeler, you **can** be sure your database applications are correctly designed. To order call 01256 469460, or fax or post the coupon for a FREE information pack.

Asymetrix™
InfoModeler

UK Distributor:

ICS SOLUTIONS LIMITED

To order call 01256 469460

Please send me a **FREE** information pack

Name _____
Title _____
Company _____
Address _____
_____
_____ Post Code _____
Tel _____ Fax _____
EXE

Return to ICS Solutions FREEPOST BZ152
Kingsclere Road, Basingstoke RG21 2BR Fax 01256 842362

* Limited offer only, price excludes VAT & shipping
all trademarks acknowledged

CIRCLE NO. 382

anyway when the program terminates. However, we should be very careful with such practice.

Suppose that we wanted to use the `dynpro` class in a different program, in a function which defines a local instance of `dynpro`, as in this fragment:

```
void f()
{
   dynpro DP;
   ...
}
...
for (i=0; i<10; i++) f();
```

Here the destructor is essential to prevent memory leakage. If we omitted it, the dynamically-allocated table, although inaccessible, would remain in memory after each call, so there would be ten copies of it after the `for` loop. This is a common error because we are used to local variables being automatically released on function exit. Incidentally, if you plan to use the `dynpro` class in a program of your own, you should be aware that I did not provide a copy constructor or an assignment operator. You must therefore prevent `dynpro` objects, such as `DP`, from being copied (eg if passed to a value parameter); neither can you safely assign `DP` to another `dynpro` variable.

The style used in the `main()` function, with all the work delegated to member functions, is applicable to many programming problems. When programming in C, I usually make my `main()` functions larger, as you may have noticed if you happen to be familiar with my last book, *Programs and Data Structures in C*. You can actually find a C version of the *dynpro* program in it, which also works quite well, but I prefer this C++ version for reasons of readability.

This new version is also different from the old one in the way the selected terms belonging to $F(x)$ are stored. Recall that each value $F(x)$, except $F(0)$, is based on some selected value $F(u)$, where u is less than x, and $F(x) = F(u) + 1$. The terms that together form x are the same as those that form u, except that one new term is added. Instead of storing all these terms for $F(x)$, we need store only a reference to u along with the new term used for x. Since the information at position u may in turn be based on a previous position in the table, we have to follow a chain of table entries to find all terms required for x. Frankly speaking, I initially overlooked this efficient way of storing sets of terms. I did not think of it until I was writing this article and remembered about the LZW algorithm for file compression as described by T. A. Welch. *(Beware: the LZW algorithm invented by Terry Welch is copyrighted by Unisys. See the cover story of the May issue of EXE - Ed.)* This compression algorithm is based on a 'string table', where each table entry contains both the final character of a string and a link to another table entry for the preceding characters. Replacing characters with integers, I obtained exactly what I wanted for the present program. As for the table in question, I initially used a dynamically-allocated array of structures:

```
struct table{int F, ref, val;} *t;
...
t = new table[s + 1];
```

However, in view of the possibility of `s` being very large, I later decided to replace this single 'array' with three separate ones:

```
int *F, *ref, *val, ...;
...
F = new int[s + 1];
ref = new int[s + 1];
val = new int[s + 1];
```

This has the advantage that it also works if three separate memory blocks, of `(s + 1) * sizeof(int)` bytes each, are available, while one contiguous block for `t`, three times as large, is not. For example, some compilers running under MS-DOS have problems with allocating a single block larger than 64 KB. I have omitted checks to see whether memory allocation with `new` is successful, because they would have made the program less portable. Unfortunately, this subject is not as simple as it looks, due to the language still being subject to changes. You could add such checks using the exception feature if it's implemented by

```
// dynpro: Dynamic programming.
// To appear in: Algorithms and
// Data Structures in C++,
// L. Ammeraal, Wiley, 1996.
#include <iostream.h>
#include <limits.h>

class dynpro {
public:
  dynpro();
  ~dynpro()
  { delete[]d; delete[]F;
    delete[]ref; delete[]val;
  }
  void solve(), print();
private:
  int s, n, *d, *F, *ref, *val;
  void product(int m, int v)
  { if (m > 1) cout << m << " x ";
    cout << v;
  }
};
dynpro::dynpro()
{ cout << "Enter the desired sum: ";
  cin >> s;
  F = new int[s + 1];
  ref = new int[s + 1];
  val = new int[s + 1];
  cout <<
   "Enter n, followed by n integers:\n";
  cin >> n;
  d = new int[n];
  for (int i=0; i<n; i++) cin >> d[i];
}
void dynpro::solve()
{ int x, u, u1, i, i1, F1;
  F[0] = 0;
  for (x=1; x<=s; x++)
  { F1 = INT_MAX;
    for (i=0; i<n; i++)
    { u = x - d[i];
      if (u >= 0 && F[u] < F1)
      {  F1 = F[u]; i1 = i; u1 = u;
      }
    }
    if (F1 < INT_MAX)
    { F[x] = F1 + 1;
      ref[x] = u1;
      val[x] = d[i1];
    } else F[x] = INT_MAX;
  }
}

void dynpro::print()
{ if (F[s] == INT_MAX)
    cout << "No solution.\n";
  else
  { cout << "Solution:\n";
    cout << s << " = ";
    int k = s, v = val[s], m = 1;
    for (int j=2; j<=F[s]; j++)
    { k = ref[k];
      if (val[k] == v) m++; else
      { product(m, v); cout << " + ";
        v = val[k]; m = 1;
      }
    }
    product(m, v); cout << endl;
  }
}

int main()
{ dynpro DP;
  DP.solve();
  DP.print();
  return 0;
}
```

Listing 1 - The full 'dynpro' program

your compiler. You might at the same time add some other checks on the validity of the input data, omitted here for reasons of readability.

As should be clear by now, `F[x]` is equal to the number of terms that add up to `x`. These terms are found as the following elements of array `val`:

`val[x]`
`val[u1]` *where* `u1 = ref[x]`
`val[u2]` *where* `u2 = ref[u1]`
`val[u3]` *where* `u3 = ref[u2]`

and so on, until we have found `F[x]` terms.

I used the maximum `int` value `INT_MAX`, defined in LIMITS.H, as a code for *undefined*. This made it possible, with a given value `x`, to select the minimum of $F(x-d_0), ..., F(x-d_{n-1})$ as follows:

```
F1 = INT_MAX;
for (i=0; i<n; i++)
{ u = x - d[i];
  if (u >= 0 && F[u] < F1)
  { F1 = F[u]; i1 = i; u1 = u;
  }
}
```

If I had used, for example,

```
const int undefined = -1;
```

the test in the middle of the above fragment would have been slightly more complicated and less efficient:

```
if (u >= 0
  && !(F[u] == undefined)
  && F[u] < F1)
```

It is now time to see the program in action. Listing 2 shows the program running with one of our previous examples. When demonstrating a program, it is always good practice to also show an abnormal case, such as the one in Listing 3, which has no solution at all. Its simplicity makes it easy to verify this: The `print()` member function would have been considerably simpler had I displayed all the terms separately, writing, for example, 6 + 6 + 6 instead of 3 × 6. But then, when expressing a large number as a sum of small terms, the output would have consisted of very long list of such terms.

```
Enter the desired sum:
42
Enter n, followed by n integers: 3
25 10 1
Solution:
42 = 4 x 10 + 2 x 1
```

Listing 2 - A demonstration

```
Enter the desired sum: 11
Enter n, followed by n integers:
3
9 7 5
No solution.
```

Listing 3 - An abnormal case

Since you may find the `print()` function hard to read, let me give you an explanation. Instead of immediately displaying the first term, `val[s]`, I store it in the variable `v`, because identical terms may follow. The multiplication factor `m` indicates how many identical terms there are. As soon as a different term `val[k]` is encountered, then, with the previous term still stored in `v`, the product `m` × `v` is displayed in the form of two integers separated by a lower-case `x` letter, and followed by a 'plus' character. Since multiplying by 1 looks silly, I omit 'm ×' if `m` is equal to 1. When `F[s]` terms have been dealt with in this way, the value of the last `m` term has not yet been output and is still available in `v`. The final product `m` × `v` is therefore displayed after completion of the `for` loop.

The available terms can be specified in any order in the input. The terms in the output will appear in the same order. In the examples of this article, I consistently started with the largest term in the input, so that the largest also came first in the output.

Is C++ any good here?

Part of this article would have been the same if I had used C instead of C++. If you plan to switch from C to C++, you will be disappointed if you expect this to make every algorithm much easier to understand and implement. There are those for whom this is the case, but the advantages of C++ over C are not always spectacular. However, there is little reason not to use C++. As the present example illustrates, the C++ class concept will in most cases enable you to group data and functions together in a more pleasing way.

For example, suppose you had to write a `main()` function in C similar to the one shown at the end of Listing 1, displaying only three essential steps in a few program lines. Since you don't like global variables, you would have had to define them in your `main()` function. This would not have been as compact as the code shown in Lisiting 1. It would have looked like this:

```
int main(void) /* C solution */
{  int s, n, *d, *F, *ref, *val;
   setup(&s, &n, d, F, ref, val);
   solve(s, n, d, F, ref, val);
   print(s, n, d, F, ref, val);
   return 0;
}
```

Of course, you could reduce the number of arguments by grouping together some (or all) of them in a structure. But you will surely prefer the C++ practice of encapsulating data and functions in a class, once you get the hang of it. ■

Ammeraal, L. (1995)
C++ for Programmers, 2nd Edition,
John Wiley.

Ammeraal, L. (1995)
Programs and Data Structures in C, 2nd Edition, John Wiley.

Bellman, R. E. (1957)
Dynamic Programming,
Princeton University Press.

Welch, T. A. (1984)
A Technique for High-Performance Data Compression,
IEEE Journal, June 1984.

Leendert Ammeraal has been programming since 1960 and graduated in mathematics at Delft Technical University. He has written about 20 books on programming, in both Dutch and English, and is currently a lecturer at Hogeschool van Utrecht. You can contact him by email at mmeraal@ibk.fnt.hvu.nl.

This article is adapted from the book 'Algorithms and Data Structures in C++' to be published by John Wiley & Sons in April 1996 (ISBN: 0471-96355-0, price: £22.50). John Wiley can be contacted on 01243 779777 or by email at market@wiley.co.uk.

Programmer's Paradise® HAS ARRIVED !

Paradise No. | Discount Price | Paradise No. | Discount Price

Assembly Language

Phar Lap TNT Dos Extender
PLTN73100-E2 ..... £318

Basic Language

3DWidgets
SH3DM3100-E2 ..... £80
Data Widgets
SDDWW3100-E2 ..... £87
ImageKnife/VBX Professional Pack
MAIKP3100-E2 ..... £226
Microsoft Visual Basic Professional C.U
MSVVBP3100-E2 ..... £199
PDQCOMM for Windows
CRPCW3100-E2 ..... £90
QuickPack Professional for VB
CRQKPW3100-E2 ..... £130
Spread/VBX
FAVVS3100-E2 ..... £156
TrueGrid Professional for VB
APTGP3101-E2 ..... £95
VB Tools
MHVT43100-E2 ..... £102
VBAssist V3.5
SSVBU3100-E2 ..... £128
VisualTools Developer's Suite
VTVTD3100-E2 ..... £225
WinWidgets/VBX
LPWWV3100-E2 ..... £89
Microsoft Visual Basic Prof. 4.0 Upg
MSVPC0250-E2 ..... £109

C/C++ Language

Accusoft Image Format Library V.5.0 / WBX32
ACVT53100-E2 ..... £927
Borland C++ V4.5 and Database Tools
BOCDT8888-E2 ..... £399
CodeBase V5.1
SQCOD3200-E2 ..... £224
Greenleaf CommLib V5.1
GRCLA3100-E2 ..... £223
IBM C Set++ for OS/2
IBM30H1758-E2 ..... £379
LEADTOOLS Professional
LTLP43100-E2 ..... £610
Microsoft Visual C++ V2.0 Pro
MSVC28888-E2 ..... £288
ProtoGen+
PDPGP3100-E2 ..... £99
Symantec C++ V7.0
SYCW73100-E2 ..... £149
Turbo C++ Visual Edition
BOTCVW3100-E2 ..... £85
Watcom C/C ++V10.5 Comp. Upg.
WACCU888 ..... £129

Client/Server

Borland Delphi
BORDE8888-E2 ..... £156
Clarion for Windows 1.5
CLCWN3100-E2 ..... £185
IBM VisualAge
IBM17H7503-E2 ..... £1366
Power Builder Desktop for Windows V4
PSPBD8888-E2 ..... £449
ProtoGen+ Client/Server Suite
PDPCS3100-E2 ..... £763

Communications

PROCOMM Plus for Windows
DAPC23100-E2 ..... £70

Disk Management

MKS Toolkit (Dos & Win)
MKTKD3200-E2 ..... £189

Editors

Codewright
PCCW33100-E2 ..... £151
ED-the programmers editor for Windows
LXEDW0002-E2 ..... £131
Multi-Edit for Windows
ACME3100-E2 ..... £143

File Compression

Compression Plus 4
ETCPW3100-E2 ..... £167
Doc-to-Help
WTDH0250-E2 ..... £259
ForeHelp
FFFHW3100-E2 ..... £303

Help Authoring Tools &Multimedia

Help Magician Pro 3.0
SWIHM3100-E2 ..... £174
RoboHelp
BSR33100-E2 ..... £339
Windows Help Magician
SIWH23100-E2 ..... £153
Visual Install Builder
ODGVI3100-E2 ..... £256
Install Professional
KDINP3100-E2 ..... £219

Installation

PC-Install for Dos & Windows
TTPWD0250-E2 ..... £177

Linkers & Librarians

Blinker
BIBL313100-E2 ..... £175
Graphics Server 4.0
PPGHKR3100 ..... £192

EXCLUSIVE ! From Programmer's Paradise

VISUAL COMPONENTS™

FIRST IMPRESSION 32 Bit by Visual Componets
+
MS Visual Basic 4.0 Pro Comp Upg
= £296

FORMULA ONE 32 Bit
+
MS Visual Basic Pro Comp Upg
= £298

Microsoft®

Operating Systems & Environments

Windows'95 Upg
MS95C8888 ..... £59

Prototyping

Dan Brickilin's Demo II
LISBD3100-E2 ..... £123
Dan Brickilin's demo-it
LIDDW3100-E2 ..... £110

Sorce Management

Microsoft SourceSafe 4.0
MSMSS3100-E2 ..... Call
MKS Source Integrity for Windows
MKSID3100-E2 ..... £284

TCP/IP/NFS

Version/VB
SBVVBW3100-E2 ..... £118

Xbase

Chameleon TCP/IP w/Maintenance
NMCTM3100-E2 ..... £292
Crystal Reports Professional 4.5 32 bit
CSCRPW3100-E2 ..... £285
Flipper Graphics Library V6.0.
PWFLI0200-E2 ..... £226
FUNCky II Library
HJFL20250-E2 ..... £206
InfoModeler Desktop
ASIDI3100-E2 ..... £412
RAD PACK for Delphi
BRDRD8888-E2 ..... £110
Borland Visual dBASE V.5.5
BOVDB3100-E2 ..... £169
R&R Report Writer for xBase for Windows
CDRWW0250-E2 ..... £176

I Want a *FREE* Subscription to the No.1 source of leading products for the software developer!

EXE-10/95

Please send a catalogue to me

Name _____
Company _____
Address _____
_____
Post Code _____
Telephone _____ Fax _____

cut on dotted lines and mail in or fax to: 0161 - 7284017

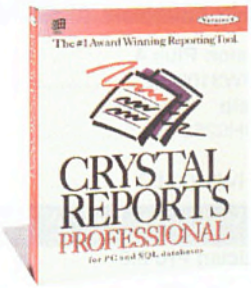
Programmer's Paradise (UK)Limited . Ist Floor, 217 Worsley Road
Swinton Manchester - M27 5SF
free ordering:0500 284177 - telephone 0161 7284177

Prices do include Vat or delivery. Please at time of order as prices are subject to chage.

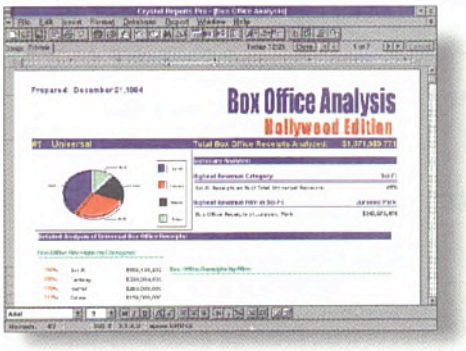
CIRCLE NO. 383

WHICH ONE HAS THE BEST REPORTING TOOL?

THEY ALL DO.



If Microsoft®, Borland® and 80 other software vendors have one thing in common, it's that they all know a great reporting tool when they see one. That's why they all include Crystal Reports in their software programs.



Crystal Reports' Report Engine DLL, full-featured OCX & VBX, and open APIs allow for fast, seamless integration into your database applications. And now, with Crystal Reports 4.5, you get hot new features that allow you to generate presentation quality reports with even greater speed and control.

The number one reporting tool is even better....now with 32-bit technology.

- 16 or 32-bit Report Designer & Report Engine. Great for Windows 3.1, 95, NT.
- 40 new engine calls (>80 total), full VBX and 16 or 32-bit OLE Control (OCX).
- 2-10 times faster report processing.
- 12 fully-integrated graph styles.
- Report Distribution Expert with free runtime.
- Easier to use interface. And much, much more.

Order Crystal Reports 4.5 today.

Crystal Reports also comes with another important feature: a money-back guarantee. Order version 4.5 today for **£299**, or call for special upgrade and Visual Basic user pricing.

Order today — and make the reporting tool of the experts your tool of choice.

CONTEMPORARY Software

Contemporary Software, Abbey View, Everard Close,
St Albans, Hertfordshire AL1 2PS
Tel: (01727) 811999 • Fax: (01727) 848991
CONTEMPORARY SOFTWARE A DIVISION OF INTERSOLV PLC

Call now and get Crystal Reports version 4.5 for just £299. TEL: 01727 811999

CIRCLE NO. 384

C++ Faux Pas

The C++ language has more holes in it than a string vest.
Roland Perera presents a list of potential traps and how to avoid them.

C++ HAS TWO FAIRLY ORTHOGONAL GOALS: maintaining backwards-compatibility with C, on the one hand, and being a strongly-typed, object-oriented programming language on the other. The former of these two aims tends to compromise the latter. Even expert C++ programmers have to keep their wits about them if they want to avoid the traps and pitfalls (or is it pits and trapfalls?) that beset C programmers.

Only some of the points made here identify true *faux pas*, to be avoided at all cost - the remainder is a hotchpotch of guidelines, recommendations and rules of thumb. Ignore them at your peril!

Don't rely on the order of initialisation of globals

The C++ language doesn't define the *order* in which statically-allocated global objects from different files are initialised. What *is* guaranteed is that all initialisation within a given translation unit will happen in order, and that global initialisation will have completed by the time `main()` starts executing. General tips to get around this problem: (a) minimise the order dependencies between objects' constructors where possible; (b) reduce the number of global objects by wrapping them in classes, passing more parameters, or using local variables; (c) put order-dependent objects in the same translation unit (*in* the correct order!). Note that this problem doesn't arise with variables or constants that don't require run-time initialisation.

Don't rely on the order of argument evaluation

C++ also doesn't define the order in which arguments to a function are evaluated, only that by the time the function body begins executing, *all* arguments will have been evaluated. So a call such as `f(a++,a++);` is in error if the caller relies on the increment operations happening in a certain order. This problem manifests itself in another, more subtle form: implicit `this` pointers are also subject to this undefined ordering!

```
cout << a++ << a++;
// is equivalent to:
cout.operator<<(a++).operator<<(a++);
// which is basically equivalent to:
write(write(cout,a++),a++);     // which a++ is first?
```

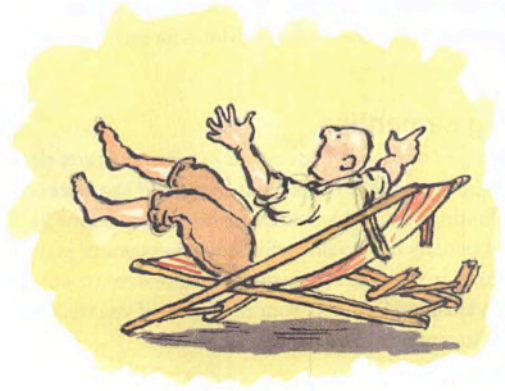
Avoid variable-length argument lists

Objects passed by value to a function via an ellipsis (...) are *bitwise*-copied onto the stack, bypassing the normal copy construction process (either memberwise copy or a user-defined copying operation). If the object's class has non-trivial copying semantics, these copies will likely have invalid state. As Marshall Cline, author of the C++ FAQ repository on the Internet, has put it, 'The technical term for this asymmetry is "ouch".'

Even if you manage to steer clear of this problem, variable-length argument lists should still be avoided for the simple reason that they're not type-safe. (Streams provide a type-safe and flexible alternative to the standard I/O library.)

Don't return non-const references to private data

If you're going to do this, there's no point in making the data member `private` in the first place! There are two alternatives: returning the *value* of the `private` member (ie a copy of the data), or returning a `const` *reference* (or pointer) to the member. Which alternative you choose depends primarily on the logic of your application and efficiency considerations. You won't want to copy the data if either (a) the copying operation is computationally expensive, or (b) you actually want a 'handle' on the data in question, rather than a copy.



Remember 'The Big Three'

The Law of The Big Three, courtesy of Marshall Cline, can be stated thus: *If a class needs a destructor, or a copy constructor, or an assignment operator, it needs them all.* The principle is that as soon as you discover the need to do something special in a constructor, you'll probably end up *un*doing that action in the destructor. That special thing will probably also need to be done during copying, and for the same reason, during assignment. To make this idea a bit more concrete, consider the notion of 'remote ownership'. For example, a `String` class might have a pointer to an array containing the characters in the string. When we copy a `String`, the data in this array should be copied, not just the pointer to the first character: the `String` has *remote ownership* of the pointer referent.

Avoid pre-processor macros

Remember, the pre-processor expands macros before your compiler even gets a look-in. There are at least two points to be aware of. The first relates only to constants: instead of `#define BUF_SIZE 2000`, use `const int BUF_SIZE = 2000`. The advantage of a declaration like this over a `#define` is that a `const` is a symbol that obeys the normal scoping rules of C++, and which browsers, debuggers and compilers know about. As constants have static (file-scope) linkage, you can put this declaration in a header file as you would with a `#define` directive.

The second point concerns macro 'functions'. The problems with these textual transformations are manifest, particularly when an expression (rather than just a variable) is passed as an argument to the macro. If the formal parameter receiving the argument occurs more than once in the body of the macro, the argument form will be evaluated more than once! For example, given:

```
#define MAX(a,b) ((a) < (b) ? (b) : (a))
```

an expression such as `MAX(x++,y)` results in `x++` being executed twice (once for the `<` operator and once for the `?` operator). Use an inline function, possibly templated, instead:

```
template <class T> inline
T max (const T& a, const T& b) {return a < b ? b : a;}
```

Make destructors virtual

Generally speaking, a class that has virtual functions should also have a virtual destructor. Otherwise deleting an instance of the class through a *base class* pointer will not result in the correct destructor being called:

```
void f (Stream* pStream) {
    delete pStream;
    // may actually point to a derived class object
}
```

Remember to de-allocate arrays correctly

Let `X` be an array of some type `T`. As any C programmer will tell you, a subsequent occurrence of the name `X` actually resolves to a pointer to the first `T` of the array. So applying `delete` to `X` (assuming `X` was dynamically allocated in the first place!) will delete only the first element of the array, since as far as the compiler is concerned `X` is a simple `T*` pointer. Instead, use the `delete[]` operator to de-allocate the entire array - but, conversely, make sure you don't use `delete[]` on a single object.

An interesting (and dangerous) corollary of the fact that a pointer to a single object has the same type as an array variable is that, given a class `D` derived from a base class `B`, an array of `D`s is an acceptable argument to a function that expects an array of `B`s. If you then try using subscripts to access the array, and `sizeof(D)` is not equal to `sizeof(B)`...bang!

Avoid type-switching

A 'type switch', or *type case*, is a `switch` statement (or similar construct) that selects an action to take by querying the dynamic type of a variable using C++'s Run-Time Type Information (RTTI) mechanism.

Performing an explicit type switch violates the 'open-closed' principle: code should be *open* to extension, but *closed* to modification. (Robert Martin's book, reviewed in *EXE*'s September issue, explores this idea in depth.) In the example below, if `Shape::Draw()` is not declared `virtual`, code like that found in `DrawElements1()` is required. Such code is *not* closed as the addition of a new class derived from `Shape` will require a modification to `DrawElements1()` to support the new sub-class.

Alternatively, if the `Draw()` method of the `Shape` class is declared `virtual`, the function `DrawElements2()` is 'closed' to the addition of new `Shape` derivatives. The function would not have to be changed should new sub-classes of `Shape` be added - it is open to extension.

`DrawElements3()` shows how templates can be used to allow client code to be even more generic: now the same function can be applied to a `vector` of any object that supplies a `Draw()` method. (`DrawElements2()` only works with sub-classes of `Shape`.)

```
void DrawElements1 (vector<Shape*>& v) {
    for (iterator i = v.begin(); i != v.end(); i++) {
        Circle* pc;
        if (pc = dynamic_cast<Circle*>(*i))  // bad
            pc->Draw();
        if (typeid(**i) == typeid(Triangle)) // worse
            static_cast<Triangle*>(*i)->Draw();
    }
}

void DrawElements2 (vector<Shape*>& v) {
    for (iterator i = v.begin(); i != v.end(); i++)
       (*i)->Draw();
}

template<class T> void DrawElements3 (vector<T>& v) {
    for (iterator i = v.begin(); i != v.end(); i++)
       (*i)->Draw();
}
```

Be careful with constructor initialisation lists

For efficiency reasons, initialisation lists in constructors are generally preferable to assignment in the constructor body. In fact, initialisation lists are the *only* way of initialising reference and `const` members (you can't assign to a `const`).

There is one thing to remember. *The order of initialisation matches the class definition, not the order given in the initialisation list*: immediate base classes first (left-to-right), and then member objects in order of declaration as per the class body. So watch out for any order-dependent initialisations - these are candidates for inclusion in the constructor body.

Stick to a consistent overloading semantics

In a bit of code I saw recently, the programmer had overloaded `operator+()` for a `Person` class he had defined. Oddly enough, he had defined `operator+()` so that it *incremented* the `Person`'s age: so if `Jim`'s age initially were `25`, then evaluating `Jim + 1` would cause his age to become `26`. I say 'oddly enough' because the `+` operator traditionally has no side-effects; the `+=` operator is used to combine evaluation with assignment. Of course, in C++ you're free to overload `operator+()` (or indeed any other overloadable operator) to do a low-level format of your hard drive if you wish, but for a number of reasons it makes sense to stick to common-sense semantics.

Rules to bear in mind:

SYSTEM SCIENCE

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

Call for our new Sept/Nov CATALOGUE

Windows Development

Doc-To-Help - Int. ver	£259
Robo-HELP - help authoring	£345
Tools.h++, Linpack.h++, Math.h++	£call
SmallTalk/V Visual - ***new***	£1075
Zinc new 4.0 for Win & NT	£call
Graphics Server SDK new version	£195
Heap Agent & SmartHeap Win	£575
TCP/IP Development tools	£call
CodeBase or CodeBase++	£229
Greenleaf Commlib 5.2 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	£99
Crescent PowerPak Pro	£745
Designer Widgets or Data Widgets	£95
QuickPak Professional for Windows	£139
PinPoint/VB Rig Pro	£135/£89
Grid/VBx, Tab/VBx, and Tab/Pro - ***new***	£call
Visual Components Suite	£235
Gantt/VBX, Schedule/VBX	£call
Formula 1/ First Impressions OCXs	£195
Crystal Reports 4.5 OCX & VBX	£call

Windows 32

SmartHeap Win/32	£525
WinMaker Pro 6.0 - Win & NT	£349
Win Widgets/32	£295
VtoolsD for Device Drivers	£call
InstallShield for Win95 & NT	£call

Mathematics & Science

- Mathematica - DOS, Win, Mac, Workstations, Education & Student pricing.
- Derive - version 3
- Gauss 386i
- Origin windows scientific plotting.
- Lindo & What's Best Linear Programming

...many more for maths & stats.

UNIX

- SCO OpenServer 5.0 - new release
- SCO Dev. Systems.
- WordPerfect for UNIX.
- TCP/IP, NFS & X Window Servers

For FORTRAN

Lahey FORTRAN 90 - new	£625
NAG Fortran Library - PC/Unix	£call
MS FORTRAN-77 5.1 - DOS	£145
MS FORTRAN PowerStation -DOS/Win	£259
MS FORTRAN PowerStation 32 - ***New***	£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

TNTIDOS Extender SDK - 32 bit	£329
286IDOS Extender SDK - 16 bit	£329

Microsoft

Visual C++ 4.0 CD - Win 95 & NT	£call
Visual C++ 4.0 upg - with subscription	£call
Windows 95 & Win 95 Resource Kit	£call
Visual SourceSafe 4.0 ***New***	£call
Visual Basic 4.0 PE upg from VB PE	**£119**
Visual C++ 4.0 Ent. upg from PE	**£375**
Visual Basic Prof Ed. upg from SE	**£225**
Visual FoxPro ***New***	£call
Macro Assembler	£110
SQL Server 6.0 ***New***	£call
Windows NT 3.51 Server & Workstation	£call

Editors and Tools

CodeWright Prog Editor Win or NT	£159
Fusion-drop in editor for VC++ 1.x & 2	£99
InstallShield Windows or Win NT	£Call
PC Lint C/C++ - Lint code checking	£145
System Commander-multi-boot loader	£69
PC-Install-Win/Win32 Combo ***New***	£call
DemoShield - Build Windows Demos	£call
Winstall MultiPlatform - Install Tool	£325
DemoQuick-Win and now Win NT & 95	£call

MKS

MKS Toolkit of Unix tools	£195
Source Integrity Multi-Platform ***New***	£359
MKS Lex & Yacc for C, C++ & TP	£215

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

1-6 BRADLEY'S CLOSE, WHITE LION STREET, LONDON N1 9PN

SYSTEM SCIENCE

Your first choice for better development tools

Watcom

Watcom C/C++ 10.5 CD ***New***	£139
Watcom Fortran 77 - new ver 10.5 CD	£345
Watcom SQL - DOS,Win,OS/2	£195
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	£285
Borland C++4.5 & Dbase Frameworks	£415
Turbo C++ 4.5 Windows - ***New***	£69
CodeGuard for Borland C++	£69
Delphi/Delphi Client Server	£295/795
Delphi RAD Pack	£129
Visual Solutions Pack of Vbxs	£69
Turbo Assembler	£69
Borland Pascal & Turbo Pascal	£call

Nu-Mega

BoundsChecker Standard Ed - Win 95 or Win NT	£235
BoundsChecker Prof Ed - Win 95 or Win NT	£395
BoundsChecker Prof Ed Pack - both Win 95 & NT	£625
BoundsChecker Win 3.1, Win32s or DOS	£195
Soft-ICE - Win 95 (beta), Win 3.1, or DOS	£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. ***Now new ver 5.2***

OS/2

OS/2 Warp Connect - p2p network	£115
OS/2 Warp Win-OS/2 - CD/3.5"	£99/£115
VisualAge C++ - comp upgrade	£call
DB/2 for OS/2 - Single or client/server	£call

CLIENT/SERVER

Btrieve Dev Kit - Win, NT, DOS, or OS/2	£395
Scalable SQL for Netware	£call
PowerBuilder Portfolio - ***NEW***	£329
Component PowerPack for PB4 ***New***	£199

CASE

EasyCASE Professional Win 4.2	£645
EasyCASE DataBASE Eng for SQL	£call
InfoModeller Desktop	£call
Star Designor for Windows	£259

NEWS

- **BoundsChecker *NEW* Standard & Professional Editions** - SE has new GUI, new helpfile with full explanations and sample code for each exception and integration into VC++ 2.1 + IDE. PE has new **Compile Time Instrumentation** for more thorough testing, including C++ usage checking.
- **Visual Basic 4.0 Enterprise Edition** - with 16- and 32-bit compilers, OCX and OLE server capability, plus direct drivers for Oracle and SQL server - upgrades from VB Prof. Ed. are available.
- **Visual C++ 4.0 (DC)** - new for Windows 95 & NT, with MFC 4.0, and all new GUI. Upgrades are available.
- **Delphi** - VCL components from TurboPower: Orpheus for extra controls and validated data-entry. Asynch Professional for serial communications within Delphi.
- **PowerBuilder Portfolio** - great value bundle for good database design with PowerBuilder Desktop, Watcom SQL server (3 user) and Star Designor - all Win 95 compatible.

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

PROFESSIONAL TOOLS FOR PROFESSIONAL DEVELOPERS

FROM GREENLEAF SOFTWARE

CommLib. The premier communications library. £259 VERSION 5.2
A library of language-independent C routines that provide easy, reliable high speed communications for your PC applications programs. File transfer protocols include CompuServe B+; XMODEM, YMODEM, ZMODEM, ASCII file transfer protocol and Kermit. ANSI Terminal Emulation and NASI support.

Comm++. A communications class library for C++. £229 NEW VERSION 3.0
Comm++ supports object oriented programming through the use of inherited classes.The library was designed to make use of asynchronous serial communications in a C++ program as easy as possible. Features include XMODEM, YMODEM, ZMODEM, Kermit file transfer and CompuServe B+, ANSI and TTY terminal emulation. Multiport hardware support and provisions for portability.

Database Library. A versatile library of C programming tools. £199 VERSION 4.0
The Greenleaf Database Library is for programmers who want to develop their xBASE applications in C without the limitations of using one xBASE language. Level 2 database control, Level 2 Relational capability. The library is compatible with dBASE III, dBASE IV, Clipper, FoxBASE, FoxPro, and compatible xBASE DBMS products.

ArchiveLib. Robust data compression and archiving for C, C++. £229 VERSION 1.01
ArchiveLib is a new product from Greenleaf, a data compression library that provides a set of programming tools to let developers quickly and efficiently compress ASCII or binary data into an archival library for storage. It is 100% C++ library plus C and Visual Basic APIs, supports C++, C, Visual Basic and others through Windows DLLs. Multilanguage, multiplatform, extensible data compression and archiving.

What they support:
MS-DOS, Windows, Windows 95, Windows NT, Win32, Borland PowerPack 32 bit.
Comm++ and the Database Library also support OS/2.
ArchiveLib also has support for Watcom 32 bit extended DOS.
CommLib and Comm++ now have support for NASI and a Win32s Thunking Layer.
Products support all the major C/C++ compilers and Visual Basic.

Greenleaf products come with:
Full Source code, No Royalties, 60 day money back guarantee.
FREE Technical support and access to our BBS.
Professional documentation as with all our products.

Full information available through:
Citadel Software Limited
Coombe, Trewen, Launceston, Cornwall PL15 8QF
TEL 01566 86037 FAX 01566 86147
BBS 01566 86925 Prices exclude VAT and delivery.
Official European Agents for Greenleaf Software Inc.

GREENLEAF Software®

CIRCLE NO. 408

PREMIER LIBRARIES FOR DOS AND WINDOWS

- don't confuse operators that simply return values (eg `operator+()`) with those that have side effects (eg `operator+=()`);
- if operators are intended to manipulate some mathematical property, ensure that the obvious mathematical relationships between the operators are honoured (see example below);
- ditto for operators such as `operator[]()` and `operator*()` (eg `p[i]` should be equivalent to `*(p + i)`).

```
++x;    x -= -1;    x += 1;    x = x + 1;
```

These 4 statements should mean the same thing.

Be aware of the lifetimes of temporaries

The compiler creates temporary objects in two situations. Probably the most salient of these is when a function (or operator) returns an object by value: the returned value is maintained as a temporary object in case it is needed as an argument to another operation. If `c1`, `c2` and `c3` are objects of type `Complex`, then evaluating the expression `c1 + c2 + c3` will create a temporary object holding the result of `c1 + c2`, which is then passed along with `c3` to the second `+`.

The other situation that creates temporary objects is the initialisation of a `const` reference to anything other than an l-value of the reference's type: in this case the compiler implicitly invokes a suitable constructor to convert the 'source' type to the 'target' type. So statements like `const String& str = "hello";` are acceptable; the character array `"hello"` is used to initialise a temporary `String` object (via a `String` constructor) to which the reference is then bound. `const` references can also be initialised to temporary objects returned from functions, operators and explicit constructor calls.

So what's the problem with temporary objects? Well, their lifetimes are not as obvious as those of normal variables. Temporaries that result from expression evaluation are destroyed at the end of the statement; one bound to a reference is destroyed as soon as the reference goes out of scope. A caveat falls out of these semantics: don't return a `const` reference to a `const` reference parameter, as the parameter may have been initialised to a temporary which will be destroyed as soon as the parameter goes out of scope at the end of the function:

```
const CString& uh_oh (const CString& s) {return s;}

void main () {
    const CString& str1 = uh_oh("hello");
    cout << str1;                              // bang!
    const CString& str2 = "hello";
    cout << str2;                              // OK
}
```

Look out for implicit construction

As hinted at in the point above, constructors for a class `X` can be invoked implicitly in a variety of situations when the compiler performs a silent conversion from some type to `X`. The class must provide a constructor taking a single argument of the appropriate type. For example, if class `X` provides a constructor `X(int)`, then an integer can be implicitly converted (constructed) into a temporary `X` object when an `X` value is required. Implicit conversion like this is certainly handy, but reduces the chances of the compiler detecting genuine errors. ANSI C++ compilers will soon allow constructor declarations of the form `explicit X(int);` which will force the compiler to flag all implicit invocations of the constructor as errors. Until then - beware!

```
X f (X x1, const X& x2) {
     return 5;    // implicit construction
}

void main () {
   X x = 2;       // implicit construction
   f(5,6);        // implicit construction
}
```

Avoid old-style casts

In fact, this rule should read 'avoid casting at all', although in practice this is rarely possible. There are some rules of thumb, however, that should make casting rarer and safer. First, when you find yourself deriving all your objects from a common root or using `void*` pointers so that you can implement generic 'container' classes, try templates instead. Second, use the cast operators `const_cast<>`, `static_cast<>`, `reinterpret_cast<>` and `dynamic_cast<>` in preference to the old-style `(T)e` syntax - see the article *Type cast?* in September's *EXE* for details.

Don't throw caution to the wind exiting a process

Throw (and catch!) an exception instead, and allow `main()` to return. Calling the standard `exit()` function to return control to the calling process doesn't call destructors for local variables during stack unwinding. Any side-effects performed by your local variables' constructors, such as opening a file or dynamically allocating memory, cannot be tidied up by corresponding destructors before the program exits. (See Francis Glassborow's article *Tidying up*, p52.) Statically-allocated objects *are* destroyed properly when `exit()` is called.

A similar point relates to the use of the standard C functions `setjmp()` and `longjmp()` defined in SETJMP.H. Using `longjmp()` to effect a non-local goto doesn't call destructors for stack-allocated variables. Use `catch` and `throw` instead.

Don't violate the 'Substitution Principle'

The Substitution Principle is the maxim that derived class services *require no more* and *promise no less* than corresponding base-class services. In other words, if `Derived` is a 'proper' sub-class of `Base`, a client function `f(Base&)` should work correctly when passed a `Derived` object instead of a `Base` object. 'Improper' inheritance prevents code from being 'closed' against modifications and forces programmers to introduce run-time type checks or 'capability queries'.

A well-known example of a derivation that violates the Substitution Principle is an attempt to derive `Circle` from `Ellipse`. While our mathematical knowledge tells us that the `Circles` are a subset of the `Ellipses`, `Circle` can only be a kind of `Ellipse` if `Circle` supports every service that `Ellipse` provides. If, for example, `Ellipse` provides a `resize(int x, int y)` method which asymmetrically resizes the `Ellipse`, `Circle` has to provide it too. The problem is that `Circle` cannot support `resize()` if `x` and `y` have different values. Essentially, the 'kind-of' relationship has to do with *services*, not just subsets. And finally...

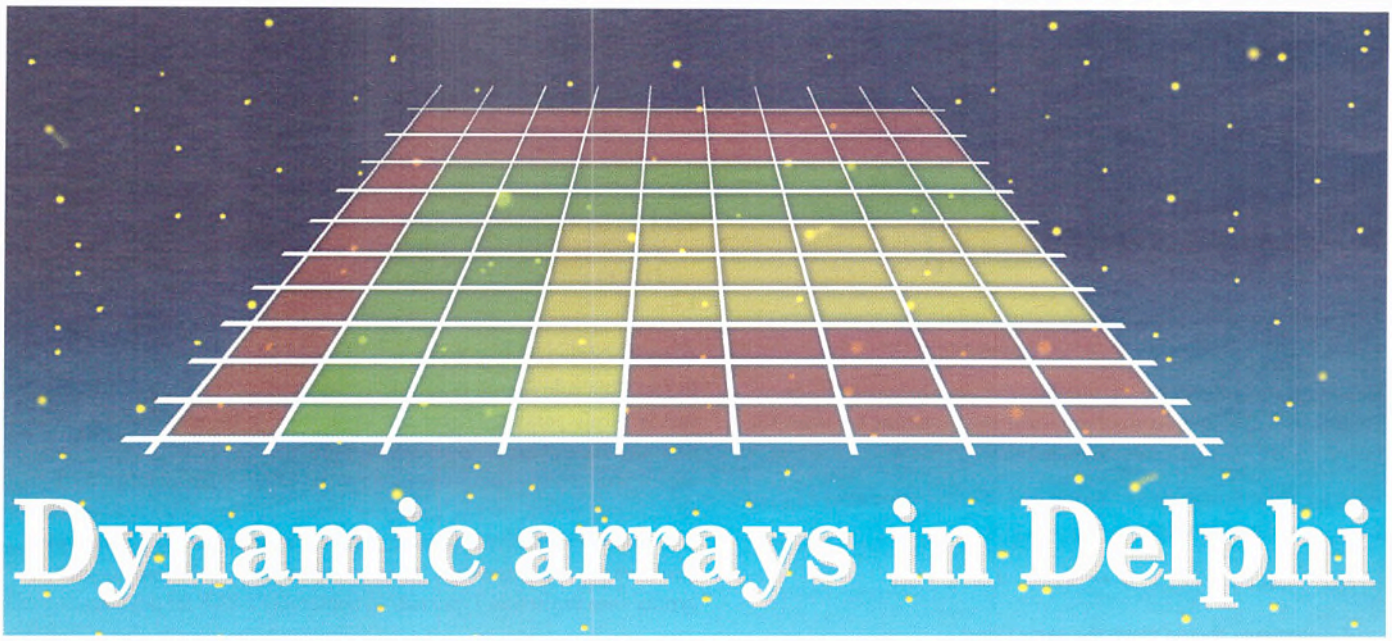
Remember there are exceptions to every rule

Including this one - and don't get stuck in a recursive loop trying to figure *that* one out! ■

Marshall Cline and Greg Lomow, C++ FAQs.
Addison-Wesley, 1994. ISBN 0-201-58958-3.

Robert Martin, Designing Object-Oriented Applications using the Booch Method.
Prentice-Hall, 1995. ISBN 0-13-203837-4.

David Spuler, C++ and C Debugging, Testing and Reliability.
Prentice-Hall, 1994. ISBN 0-13-308172-9.



Dynamic arrays in Delphi

Are you feeling cramped by standard Pascal arrays? **Warren Kovach** shows you how to create and use large, fast and resizable two-dimensional arrays with Delphi.

Memory limitations are the curse of the PC programmer's life. If your 16-bit programs need to work with large amounts of data you will often find your head hitting the ceiling of the 64 KB limit on data structures. Various methods have been developed to allow programmers to use all the memory in modern multi-megabyte computers. These vary in their complexity, flexibility, ease of use and speed.

For my statistical programming I need fast, flexible two-dimensional arrays for holding the data matrices. There are a number of approaches that can be taken, some more suitable than others. This article will summarise a variety of these methods, pointing out their strong and weak points. I'll then focus on one particular method that suits my needs.

I've developed these methods using Delphi. This is mainly because it is such a nice programming environment for Windows, but also so that I could take advantage of its 'properties' feature. Most of the techniques can also be employed in Borland Pascal, using either Windows or DOS protected mode. The object-oriented code will need to be modified to follow the earlier Borland Pascal object syntax rather than the new Object Pascal used in Delphi. This mainly involves using the `object` declaration rather than `class`, calling the constructor `Init` rather than `Create` and the destructor `Done` instead of `Destroy`.

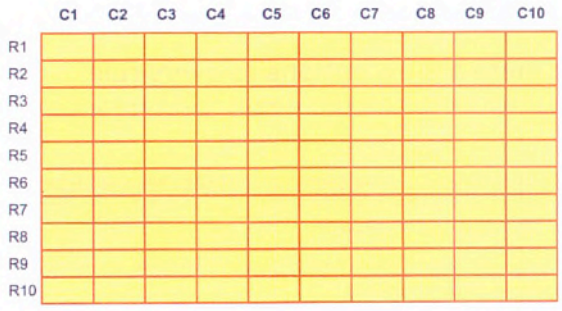


Figure 1 - A 2D standard Pascal array, with the 10 rows and 10 columns labelled

Design goals

In developing a dynamic array structure there were four priorities:

1) Speed - data access must be as fast as possible. Many scientific calculations, such as matrix inversions, can take a very long time. Analyses of large data sets can easily take hours.
2) Size - the arrays should have no predefined limits to the number of rows and columns (except the maximum value of `Longint`); some users may have tens of thousands of samples to analyse.
3) Resizing - the size of the array must be defined at run-time, not compile-time, and it should be easy to resize the array to allow the user to add new data.
4) Ease of use - the syntax for storing and retrieving the data should be as simple as possible, preferably like standard Pascal arrays.

Possible approaches

There are two basic ways that you can design a two-dimensional data structure. First, you can set up a two-level hierarchy. This consists of one data structure that either contains or points to a series of a second type of data structure. Alternatively, you can have just one big block of data, then access that as if it were two-dimensional. These two types of structure are shown in Figures 2 and 3.

Standard Pascal two-dimensional arrays are actually intermediate between these two models. A 2D array can be declared as `array [1..n] of array [1..n] of Double`, which fits the two-level model. More usually, an array is declared as `array [1..n,1..n] of Double`. In either case, the data conceptually form a 2D grid (see Figure 1). This is actually stored as a contiguous sequence of values in linear memory. The compiler then generates code that converts the row

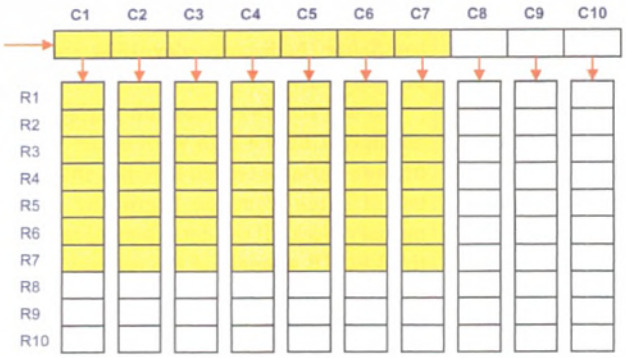


Figure 2 - A 2D array emulated by a 1D array of pointers to a series of 1D arrays. The arrows indicate pointers, while the shaded areas show the parts of the arrays actually allocated with GetMem.

and column specifications to an offset into this sequence.

This form of array may be the fastest and simplest, but it loses out on two of the design requirements. You must declare the size of the array at compile-time, not run-time. Also, in most (if not all) PC implementations of Pascal, the array is limited to 64 KB of data. When the array is composed of eight-byte `Doubles` this gives a maximum of 90 rows and 90 columns. In addition, the array must share the 64 KB data segment with other variables.

The problem with the data segment size can be surmounted by declaring a pointer to the array and using the Pascal procedure `New` to place it on the heap. In addition, we can increase the maximum size of the array by making it a one-dimensional array of pointers to a series of 1D arrays (Figure 2). The declaration for this is in Listing 1.

This will give us an array of up to 8190 rows and 16380 columns. We would need to allocate the `PDoubleArray2d` pointer and each `PDoubleArray` using `New` (and de-allocate them with `Dispose`), but we are still limited to declaring the size at compile-time.

Dynamic trio

You may find that 8190 rows and 16380 columns is adequate for your needs. However, the entire data structure would consume about 1024 MB of RAM (somewhat more than the usual requirements for a Windows program). There is a trick that allows you to declare arrays larger than necessary, and then just allocate the amount of memory needed. This involves using `GetMem` rather than `New`. The declaration would be the same as that in Listing 1, but we would allocate the memory for the pointers with:

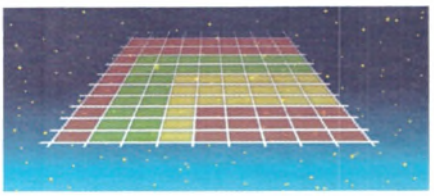
```
GetMem(DoubleArray2d,
NumCols*SizeOf(Pointer));
for i := 1 to NumRows do
    GetMem(DoubleArray2d^[i],
           NumRows*SizeOf(Double));
```

A particular element can then be accessed with `DoubleArray2d^[i]^[j]`. Of course, you must remember to de-allocate the memory with `FreeMem`. If you try to use any parts of the array outside what was actually allocated, you will get a general protection fault. This partially-allocated structure is shown shaded in Figure 2 .

You may prefer a more object-oriented approach. You could develop a two-level hierarchy of data as above, but have each 1D array wrapped up in an object with methods that may, for instance, return the sum of each array. You could also use the `TList` object in Delphi instead of the `TDoubleArray2d` structure used above. This would allow you to use the `TList` methods such as `Add`, `Insert`, and `Exchange` to manipulate the data. You would still be limited to 16380 columns, though, `TList` works in the same way as `TDoubleArray2d:` an array of pointers is declared and memory allocated with `GetMem`.

```
const
  MaxDoubleArray = (65520 div SizeOf(Double));   { = 8190 }
  MaxPointerArray = (65520 div SizeOf(Pointer)); { = 16380 }
type
  TDoubleArray = array [1..MaxDoubleArray] of Double;
  PDoubleArray = ^TDoubleArray;
  TDoubleArray2d = array [1..MaxPointerArray] of PDoubleArray;
  PDoubleArray2d = ^TDoubleArray2d;
var
  DoubleArray2d : PDoubleArray2d;
```

Listing 1- Declarations for a 2D array composed of an array of pointers (like that in Figure 2)



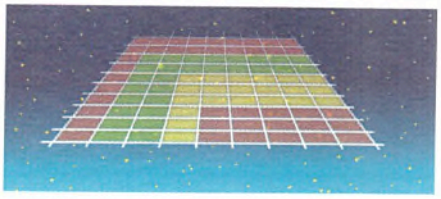
The best way to avoid all compile-time constraints on size is to grab a large chunk of memory and manipulate it yourself. Each element in the array can be accessed by taking the row and column of the element and calculating the offset into the linear block of memory. If the number of columns is `NumCols` then the offset can be calculated as `Offset := (Pred(Row) * NumCols) + Column`. You would then multiply `Offset` by the size of the data element in bytes to determine at which byte the data value starts. This is done with `Pred(Offset) * SizeOf(Double)`. This assumes that the first element in the array has an index of 1, not 0; the call to `Pred` converts the offset to the fundamentally zero-based linear memory. Figure 3 shows such a structure.

The simplest method of allocating a large block of memory is to use Delphi's built-in but poorly documented stream mechanism. By using `TMemoryStream` you can set aside as large a block of memory as you would like (limited, of course, by the amount of available memory). You can access a particular array element by calculating the offset as above and using the `Seek` method to set the stream pointer to the right place. You then use the `Read` and `Write` methods to get and set the value. Listing 2 shows a code fragment of a simple `TMemoryStream` descendant that can be used as a 2D array.

This method is simple to use and flexible. It also has the added advantage of allowing you easily to redirect the stream to a file using the `SaveToFile` method. The array can then be loaded later with `LoadFromFile`. In my quest for speed, though, I found that the overhead of the streaming mechanism reduced the performance of the arrays. I wanted an even faster method.

Speed it up

The speed of data access can be improved by bypassing the streaming mechanism and working directly with memory through the Windows API. The Pascal procedure `GetMem` is limited to allocating memory blocks of 64 KB or less, so we need to use the Win-



dows function `GlobalAlloc`. This lets you allocate memory blocks of any size, up to the amount of available memory.

One difficulty with this approach is that we must do some extra manipulation of pointers when using blocks of data larger than a single 64 KB segment. When a regular Pascal pointer reaches the segment boundary, it will wrap back to the start of that segment rather than going on to the next segment. For instance, let's say that we have a block of memory that starts at the address $1000:$0000. If we point to the end of the first 64 KB segment ($1000:$FFFF) and increment the pointer, it will point back to the beginning of the block ($1000:$0000). To access the next 64 KB segment, the first part of the address (called the *selector*) must also be incremented.

If you are programming in C you can simply use huge pointers, which automatically take care of the extra calculations to increment the selector and jump over segment boundaries. Unfortunately, Pascal does not have an equivalent, so we must do this ourselves. We are aided in this task by the `SelectorInc` variable in Borland's runtime library. This contains the value that must be added to a selector to obtain that of the next segment. The use of this variable is not described in the Delphi documentation, but there is a good explanation in Borland Pascal 7's *Language Guide*, as well as in Brian Long's book *The Borland Pascal Problem Solver* (Addison-Wesley, 1993, ISBN 0-201-59383-1).

To use the `SelectorInc` variable we multiply it by the high word of the `ByteOffset` variable (which specifies the number of bytes from the beginning of the memory block). If the data element is in the third 64 KB segment of the memory block, the high word of `ByteOffset` will be 2. We then multiply this by `SelectorInc` and add it to the selector of the pointer to the beginning of the memory block. This gives us the selector of the desired segment. The function in Listing 3 shows how to do this.

`PtrRec` and `LongRec` are useful records declared in the `SysUtils` unit of Delphi (and the `Objects` unit of BP7) which let you work with the two words that make up a 32-bit pointer or a `Longint`. In this way the selector and offset of a pointer or the high and low words of a `Longint` can be dealt with directly. These can then be manipulated appropriately and converted back to a pointer with the `Ptr` function. In the current implementation of Pascal, these two records can be used interchangeably. However, in the `GetPtr` function both are used to distinguish the pointers from the byte offsets.

This function can be modified slightly to protect against the possibility that the sum of the offsets of `P` and `ByteOffset` is greater than 64 KB; this is done with the `MasterOffset` local variable in the full source code in Listing 4. If the sum is larger than a single segment, the selector of `MasterOffset` will be greater than 0; this is then added to the selector of `ByteOffset`.

```
type
  TMemDoubleArray2d = class(TMemoryStream)
  private
    FNumRows: Longint;
    FNumCols: Longint;
  public
    constructor Create(rows,cols: Longint);
    procedure SetVal(row,col: Longint; const val: Double);
    function GetVal(row,col: Longint): Double;
end;

constructor TMemDoubleArray2d.Create(rows,cols: Longint);
begin
  inherited Create;
  SetSize(rows * cols * SizeOf(Double));
  FNumRows := rows;
  FNumCols := cols;
end;

procedure TMemDoubleArray2d.SetVal(row,col: Longint; const val: Double);
begin
  Seek(((Pred(row) * FNumCols) + col) * SizeOf(Double), 0);
  Write(val, SizeOf(Double));
end;

function TMemDoubleArray2d.GetVal(row,col: Longint): Double;
begin
  Seek(((Pred(row) * FNumCols) + col) * SizeOf(Double), 0);
  Read(Result, SizeOf(Double));
end;
```

Listing 2 - Code fragment showing the use of a TMemoryStream descendant as a two-dimensional array

```
function GetPtr(P: Pointer; ByteOffset: Longint): Pointer;
begin
  GetPtr := Ptr(PtrRec(P).Seg + LongRec(ByteOffset).Hi
                * SelectorInc,
                PtrRec(P).Ofs + LongRec(ByteOffset).Lo);
end;
```

Listing 3 - Function to return a pointer ByteOffset bytes from the beginning of the memory block

Problem cracked

With this method of accessing large blocks of memory we are now well placed to develop an object that will encapsulate this memory so that it can be used as a two-dimensional array. The class, `THugeDoubleArray2d`, is a descendant of `TPersistent`. Using this base class opens the possibility of adding methods for reading and writing the object to a stream (although this is not done here). The full source code for a unit defining this class is shown in Listing 4.

All of the data fields are declared in the `private` section; they are then accessed through methods or properties. The object must store the number of rows and columns (`FNumRows` and `FNumCols`), the total number of elements in the array (`FNumElem`), the size of the data element (`FElementSize`) and a pointer to the actual block of data (`FDataPtr`). Since the data are `Doubles` we declare the pointer as a `PDouble` (`= ^Double`). This lets us use the floating-point value directly by simply dereferencing the pointer. If we wish to make this an array of

Visual Basic Visual FoxPro Visual Objects
Delphi C/C++ Clipper PowerBuilder Lotus Notes

From desktop to enterprise, for the latest versions of the best developers' products and add-ons, call us today

3D Graphics Tools	£115.00
3D Widgets	£85.00
†Aware/VBX	£110.00
ButtonMaker	£70.00
Calendar Widgets	£90.00
ClassAssist	£169.00
†CodeBasic	£129.00
CodePrint Pro	£79.00
Common Dialog Toolkit OCX	£35.00
Crystal Reports Pro	£295.00
Crystal upgrade from VB Pro	£149.00
†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 Cmn Dlg Toolkit	£30.00
†Designer Widgets	£90.00
Distinct TCP/IP SDK Vis.Ed.	£179.00
Enquiry VBX/OCX	£230.00/299.00
†First Impression/OCX	£125.00/195.00
Formula One/OCX	£125.00/195.00
FXTools/VB Professional	£255.00
Gantt VBX	£189.00
†Grid VBX	£70.00
†Graphics Server SDK OCX	£199.00
†HighEdit SDK	£195.00
InfoSleuth	£115.00
†ImageKnife 2.0 Pro	£275.00
†ImageStream	£125.00
Integra/VDB for VB Desktop	£199.00
†LeadTools 5 OCX Prof.	£365.00
†List & Label (VB)	£275.00
M.4 VB	£155.00
†MediaKnife 1.1	£275.00
†MicroHelp Comms Library	£109.00
MicroHelp Compression Plus	£170.00
MicroHelp Fax Plus	£170.00
†MicroHelp Muscle	£119.00
MicroHelp Network	£79.00
MicroHelp OLE Tools	£140.00
MicroHelp Spellpro	£119.00
MicroHelp Uninstaller	£49.00
MicroHelp VBTools 4	£105.00
†MicroHelp VBXRef	£79.00
†MicroHelp Viewer	£79.00
Microsoft VB 4 Pro V upgrade	£115.00
Microsoft VB 4 Pro C upgrade	£209.00
Microsoft VB4 Ent V upgrade	£379.00
Microsoft VB 4 Professional	£349.00
Microsoft VB 4 Enterprise	£689.00
My Data Control	£115.00
NetPak Pro VBX/OCX	£129.00/155.00
Open Mail System (VBX)	£189.00
†PDQComm VBX OCX	£95.00/155.00
PowerPak Pro VBX/OCX	£625/1075
†QuickPak Pro VBX/OCX	£140.00/195.00
Schedule/VBX	£189.00
Sp_Assist from Sheridan	£450.00
†Spread VBX	£165.00
†SpyWorks VBX/OCX	£115.00
StorageTools OCX	£115.00
SuccessWare Database Engine	£195.00
†Tab Pro	£70.00
†ToolThings	£125.00
†TrueGrid Pro *special*	£99.00
TrueGrid Pro incl OCX	£149.00
TruedBgrid OCX	£159.00
VBA Companion from Apex	£65.00
VBAssist	£135.00
VB-Cert	£99.00
†VBCompress	£95.00
†VBDB Code/Form Generator	£149.00
VBDD Datadict.	£129.00
†VB Language Manager Pro	£195.00
VB ProjectWorks Pro 3 User	£225.00
VBtrv	£185.00
Version Stamper VBX/OCX	£120.00
Versions/VB 1.1	£75.00
Videoplay OCX	£59.00
Visual Developers Suite Deal	£205.00
†Visual Speller	£125.00
†VisualWriter Pro	£155.00
†VSFlex OCX	£89.00
†VSVBX OCX	£40.00
†VSView OCX	£89.00
Xref for VB VBX/OCX	£79.00/95.00

NOVEMBER NEWS

OCX Versions. Now available: Graphics Server, First Impression, Formula One, Calendar Widgets, Designer Widgets, LeadTools 5, VSFlex, VSVBX, VSView, Enquiry, NetPak Pro, PDQComm, QuickPak Pro, PowerPak Pro, Xref, SpyWorks, Storage Tools, TrueGrid, Version Stamper, Videoplay.. More 32 bit OCXs are coming on stream every week; phone us for the latest or check out the OCX grid on our bulletin board.

Visual Basic 4. Much improved stock levels for both Professional and Enterprise versions. Call us now if you haven't upgraded yet. Prices are: VB4 Pro version upgrade, £115, VB4 Pro competitive upgrade £209, VB4 Enterprise version upgrade, £379.00.

Delphi. *NEW.* Wise 4.0 - 16/32 bit version of this excellent installation system. *NEW.* ABC (Advanced Business Components) Bundle of 30 visual controls to add to your VCL for just £135.00. *NEW.* Titan - replaces BDE and bypasses ODBC layer for superfast BTrieve dataflow.

Haute Couture: Free teeshirt with every purchase of a Crescent product (Enquiry, PDQComm, NetPak, QuickPak, PowerPak, VBDD, Xref). Very fetching. Free teeshirt also with every purchase of NovLib.

Upgrades. Check out the price lists for V (version upgrade) and C (competitive upgrade).

†ABC for Delphi Pro	£135.00
†Apollo from SuccessWare	£115.00
†Apollo Pro	£185.00
†Async Professional	£135.00
Component Create	£135.00
Conversion Assistant Standard	£65.00
Conversion Assistant Database	£119.00
Delphi Client/Server	£799.00
Delphi CS upgrade from DT	£649.00
Delphi Desktop Comp. Upg	£129.00
Delphi RAD Pack	£129.00
Delphi Desktop plus RAD	£249.00
Distinct TCP/IP SDK Std	£375.00
†Graphics Server SDK	£199.00
ImageKnife Pro Version 2	£275.00
†InfoPower VCL + Source	£298.00
†InfoPower VCL	£199.00
†LeadTools Pro 5.0 DLL	£725.00
OMS Open Mail System	£379.00
†Orpheus	£135.00
†Mobius Drawkit	£95.00
†Mobius Widgets	£95.00
Silverware Win Comms Kit	£219.00
†Titan runtime	£225.00
†Titan runtime inc. source	£375.00
TOLEAutomation Client	£39.00

Calendar Widgets OCX	£90.00
CommTools	£225.00
Cryptor Windows DLL 6 user	£299.00
Cryptor DOS PLB 1 User	£99.00
Foxfire Developer's Edition	£299.00
FoxFix for DOS and WIN	£149.00
Graphics Server SDK	£199.00
Mac Rubel's Power Dev. Lib	£125.00
MIX (Single User)	£95.00
NetLib Network Library	£189.00
Raidar debugger	£175.00
ReFox Decompiler	£295.00
†SilverFox Comms (Win/ DOS)	£249.00
StepUp Foundation Classes	£150.00
SymScript	£145.00
T-BASE Graphics	£399.00
The FEEL Editor	£85.00
Visual FoxPro Standard	£155.00
Visual FoxPro Professional	£389.00
VFP 3 Standard V/C Upgrade	£89.00
VFP 3 Pro V/C Upgrade	£229.00
Xilights Editor Enhancement	£99.00

Barcode Library for Windows	£249.00
Barcode Library for DOS	£389.00
BugTrak 2 (1 user)	£159.00
dBest Barcodes for Windows	£345.00
DemoQuick Express	£230.00
†DemoQuick Simulation Plus	£390.00
†Doc-To-Help Help Developer	£265.00
Quicture plus Smooth Scaling	£99.00
†ED for Windows v 3.5	£145.00
ED for DOS and Windows	£199.00
†InfoModeller DeskTop	£99.00
InfoModeller Server	£499.00
†InstallSHIELD 3 Windows 3.1	£319.00
InstallSHIELD 3 Win'95, NT	£495.00
LightLib Images standard	£225.00
LightLib Images professional	£359.00
MS Office 95 Pro (Incl. Access)	£449.00
MS Office 95 Pro V. Upgrade	£449.00
MS SourceSafe 95 (platform)	£349.00
MS SourceSafe 95 Upgrade	£79.00
Microsoft Test 95	£419.00
Microsoft Test 95 V. Upgrade	£159.00
MS Windows 95 Upgrade 3.1	£69.00
MS Windows 95 Upgrade DOS	£125.00
MultiEdit Pro + Evolve	£155.00
PC-Install for DOS+Windows	£179.00
†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 (single)	£195.00
Wise Installation System (Dev.)	£149.00
†ZipitUp Software Protection	£99.95

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	£195.00
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 5.0 32 bit DLL	£965.00
List & Label (C/C++)	£389.00
MS VC++ Subscription 4.0	£349.00
MS VC++ Subs 4.0 V. Upgrade	£179.00
VBtrv for C++	£275.00
Watcom C/C++ (CD)	£199.00
Zinc App Framework	£call

Advanced Developers Toolkit	£385.00
Component Pack	£189.00
Desktop 4	£489.00
Enterprise	£3195.00
FUNCky for PowerBuilder	£175.00
†PowerFrame App.Fr. Library	£305.00
PowerFrame Navigator Object	£99.00
PowerFrame Object Analyser	£99.00
PowerFrame TabFolder	£99.00
Powersoft Portfolio	£329.00

Adv. Xbase Server SDK	£69.00
Adv. Xbase Server 10 user	£1195.00
CA Visual Objects	£399.00
†Graphics Server SDK	£199.00
LightLib Business Pro	£299.00
SuccessWare Database Engine	£195.00
Silverware Win Comms Kit	£219.00

Adv. Xbase Server SDK	£69.00
Adv. Xbase Server 10 user	£1195.00
†Blinker Linker	£179.00
CA Clipper 5.3 Upgrade	£175.00
†Class(y)	£109.00
†Clip-4-Win	£195.00
†Comix RDD	£99.00
†dGE Graphics	£189.00
Factual II	£225.00
†Fast Text Search for Clipper	£70.00
†FAXual.lib	£79.00
Flexfile	£99.00
†FUNCky	£169.00
NetLib Network Library	£189.00
†NovLib Library	£189.00
†R&R Report Writer/DOS	£195.00
Scription PostScript Library	£129.00
SilverClip SPCS Comms	£249.00
†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

Borland InterBase Win NT	£880.00
Borland InterBase Novell NLM	£880.00
Borland InterBase 8 user dev.	£2509.00
Borland InterBase 16 user dev.	£4873.00
Borland Paradox Win	£349.00
Borland Visual dBASE	£255.00
Microsoft Access 95	£275.00
Microsoft Access 95 V. Upgrade	£89.00
MS SQL Server NT 6	£720.00
MS SQL Server NT 6 Upgrade	£429.00
MS SQL NT Client (1)	£120.00
MS SQL NT Client (1) Upgrade	£69.00
MS SQL Server 6 Client (20)	£1950.00
MS SQL 6 Workstation	£385.00
MS SQL 6 Wkst Upgrade	£199.00
Watcom SQL Developers' Ed.	£189.00
Watcom SQL Server 6 User	£529.00
Watcom SQL Server 16 User	£1085.00
Watcom SQL Server 32 User	£1965.00

†Maestro Pro for Visual Basic	£695.00
†Maestro Desktop for VB	£195.00
†Notes Netware Server	£345.00
†Notes Starter Pack	£749.00
†Notes VIP Designer	£695.00
†Transporter (5 server)	£2375.00

†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
0181 994 4842

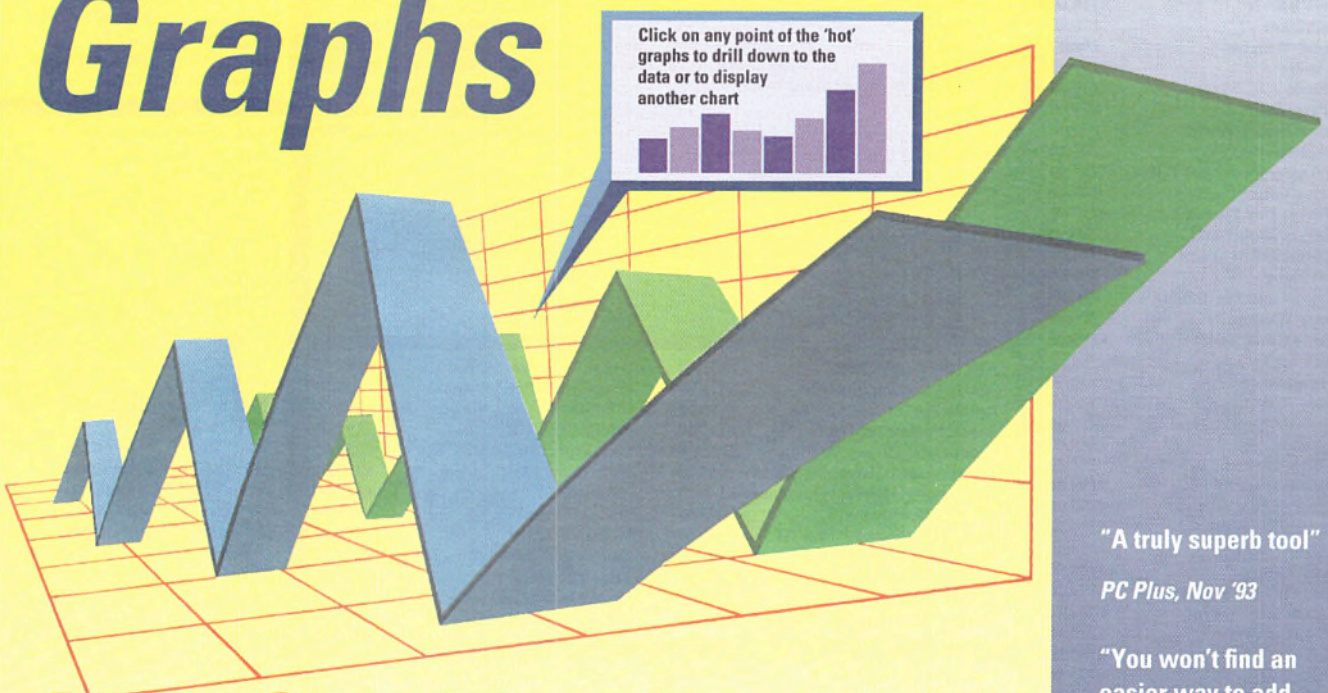
QBS Software Limited
10 Barley Mow Passage, London W4 4PH
Ph: +44 181 994 4842, Fax: +44 181 994 3441
BBS: +44 181 747 1979

*Prices, correct at time of going to press, are subject to change and do not include shipping and VAT.

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

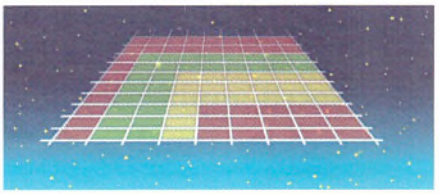
"You won't find an easier way to add charting to your Windows applications"
PC Magazine, Sept '93

BITS PER SECOND

14 Regent Hill
Brighton
England
BN1 3ED

Telephone
01273 727119

Fax
01273 731925



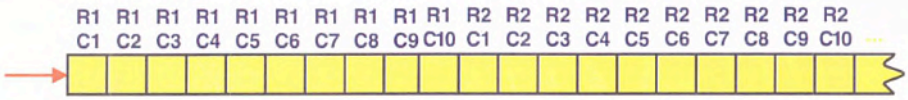


Figure 3 - A 1-D block of data, indexed as a 2-D array. The block can grow to the right as much as needed

some other data type (`Integer`, for instance) we would simply change the declaration of the pointer and the setting of `FElementSize`.

The constructor takes the desired number of rows and columns in the array and sets the various data fields. The required memory is then allocated with the Windows API function `GlobalAllocPtr`. This combines the two steps of allocating the memory with `GlobalAlloc` and locking it with `GlobalLock`. It returns a pointer to the memory block. We make sure we are Windows-friendly by allocating a moveable memory block with the `gmem_Moveable` flag. We also use the `gmem_ZeroInit` flag to initialise the block of memory to zeros.

If the pointer returned by `GlobalAllocPtr` is `nil` then the allocation or locking has failed. This is most likely to occur with inadequate memory, so we can call the Delphi procedure `OutOfMemoryError` to raise an exception. The memory is freed with `GlobalFreePtr` in the destructor.

The methods `GetVal` and `SetVal` use the techniques described above to set and retrieve the data values. The desired row and column are passed to either method and the position of the element in the block of data is calculated, taking into account the size of the data element. The address of a pointer to the element is then calculated, taking care to jump over segment boundaries. Dereferencing this pointer lets us either set or retrieve the floating-point value.

The calculations of `ByteOffset`, `MasterOffset`, and `ElemPtr` are the same in both methods. Stylistically, it would be better to place this code in a separate function rather than repeating it in each method. However, the overhead of calling a function each time reduces the performance of the arrays by 20%, according to my tests.

We can add other methods to simplify use of the data. For instance, if we regularly need to retrieve the floating-point value as a formatted string, we can add a method called `GetValAsString`. This uses the Delphi function `FloatToStrF`, which takes one of five possible formatting styles (`ffGeneral`, `ffExponent`, `ffFixed`, `ffNumber`, `ffCurrency`), the desired precision and number of decimal places and returns a string.

Resizing

Memory blocks that are allocated with `GlobalAllocPtr` can easily be made larger or smaller with `GlobalReAllocPtr`. We simply pass it the existing pointer to the block, its new size, and the flags. The `gmem_ZeroInit` flag will cause it to initialise just the new memory locations, so existing data will be preserved. If the function cannot allocate more memory a `nil` pointer will be returned. The original pointer will still be valid in this case, so we

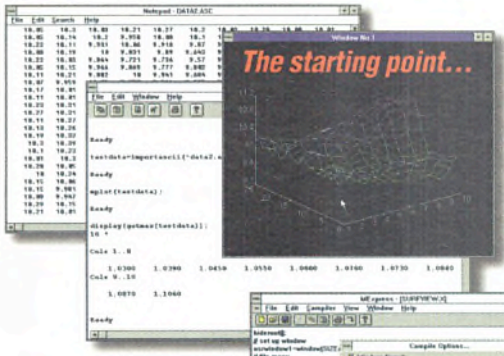
```
unit HugeArray;
interface
uses Classes, SysUtils, WinProcs, WinTypes;
type
  PDouble = ^Double;
  THugeDoubleArray2d = class(TPersistent)
    FNumRows,
   private
    FNumCols     : Longint;
    FDataPtr     : PDouble;
    FElementSize : Byte;
    FNumElem     : Longint;
   public
    constructor Create(Rows,Cols : Longint);
    destructor Destroy; override;
    function  GetVal(Row,Col : Longint) : Double;
    procedure SetVal(Row,Col : Longint;const val : Double);
    function  GetValAsString(Row,Col : Longint;
              Format : TFloatFormat;
              Precision,Digits : Integer) : string;
    function  GetPtr(ElemOffset:Longint) : PDouble;
    procedure IncPtr(var P : PDouble; IncAmount: Longint);
    function  ReSize(Rows,Cols : Longint) : Boolean;
    { this allows access as if it were a regular
      Pascal array }
    property  Data[Row,Col : Longint] : Double
            read GetVal write SetVal; default;
    { force these variables to be read-only to the
      outside world }
    property  NumRows : Longint read FNumRows;
    property  NumCols : Longint read FNumCols;
  end;

implementation

constructor THugeDoubleArray2d.Create(Rows,Cols : Longint);
begin
  inherited Create;
  FElementSize := SizeOf(Double);
  FNumElem := Rows*Cols;
  FDataPtr := GlobalAllocPtr(gmem_Moveable or gmem_Zeroinit,
                             FNumElem * FElementSize);
  if FDataPtr = nil then
    OutOfMemoryError; { raise an exception }
  FNumRows := Rows;
  FNumCols := Cols;
end;

destructor THugeDoubleArray2d.Destroy;
begin
  if FDataPtr <> nil then GlobalFreePtr(FDataPtr);
  inherited Destroy;
end;

function THugeDoubleArray2d.GetVal(Row,Col : Longint) : Double;
var
  Offset : Longint;
  ByteOffset, MasterOffset: Longint;
  ElemPtr : PDouble;
begin
  { calculate offset of 2-D coordinates into 1-D structure }
  Offset := (Pred(Row)*NumCols) + Col;
  { Use Pred to convert 1-based array coordinates to 0-based }
  ByteOffset := Pred(Offset) * FElementSize;
  MasterOffset := Longint(PtrRec(FDataPtr).Ofs)
                  + LongRec(ByteOffset).Lo;
  ElemPtr := Ptr(PtrRec(FDataPtr).Seg
             + (LongRec(ByteOffset).Hi
             + PtrRec(MasterOffset).Seg)
             *  SelectorInc,
                LongRec(MasterOffset).Lo);
  { return floating point value at that element }
  GetVal := ElemPtr^;
end;

procedure THugeDoubleArray2d.SetVal(Row,Col : Longint;
                                    const val : Double);
var
  Offset : Longint;
  ByteOffset, MasterOffset: Longint;
  ElemPtr : PDouble;
begin
  Offset := (Pred(Row)*NumCols) + Col;
  ByteOffset := Pred(Offset) * FElementSize;
  MasterOffset := Longint(PtrRec(FDataPtr).Ofs)
                  + LongRec(ByteOffset).Lo;
  ElemPtr := Ptr(PtrRec(FDataPtr).Seg
             + (LongRec(ByteOffset).Hi
             + LongRec(MasterOffset).Hi)
             *  SelectorInc,
                LongRec(MasterOffset).Hi);
  ElemPtr^ := val;
end;

function THugeDoubleArray2d.GetValAsString(Row,Col : Longint;
                    Format : TFloatFormat;
                    Precision, Digits : Integer) : string;
```

Listing 4 - Unit containing the THugeDoubleArray2d object

NEW from Adept Scientific!

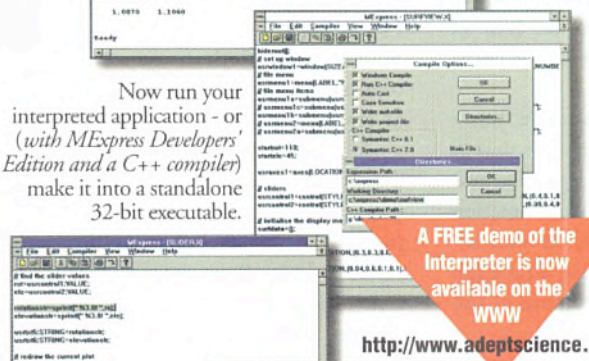
MExpress

At last — a powerful 4GL that lets you compile your routines as standalone executables!



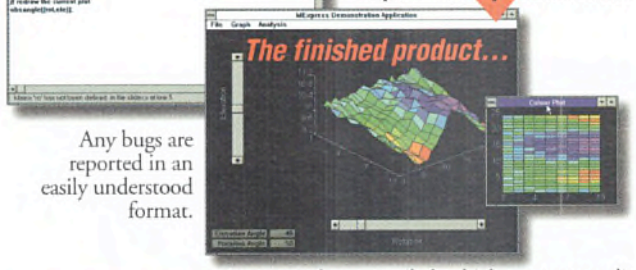
...import, analyse and display your data, then develop your script - and a professional interface.

Now run your interpreted application - or (*with MExpress Developers' Edition and a C++ compiler*) make it into a standalone 32-bit executable.



A FREE demo of the Interpreter is now available on the WWW

http://www.adeptscience.co.uk



Any bugs are reported in an easily understood format.

...interpreted or compiled, whichever you need!

New MExpress lets you develop powerful 4GL applications and compile them as standalone windows .exe programs, or run them in interactive interpreted mode! ***MExpress*** offers fast and flexible programming and seamless integration of maths and graphics.

Call Adept Scientific today for more information or fill out the coupon below and send to us.

Adept Scientific plc
6 Business Centre West, Avenue One, Letchworth, Hertfordshire, SG6 2HB, UK
Tel: 01462 480055 Fax: 01462 480213
Email: mexpress@adeptscience.co.uk WWW: http://www.adeptscience.co.uk

Need more Information?

Please fill out and fax back or post this coupon to Adept Scientific for your FREE **MExpress** information pack!

Name
Telephone No.
Position
Company
Department
Address
Town
County
Post Code

ADEPT SCIENTIFIC

(01462) **480213** *FastFax*

or return to:
Adept Scientific plc
6 Business Centre West, Avenue One, Letchworth, Hertfordshire, SG6 2HB
Tel: (01462) 480055

CIRCLE NO. 389

CAD VBX

Write graphics-based applications in a fraction of the time!

The CADControl is an exciting way to build any *graphic, mapping* or *GIS* application in Visual Basic or C++ without redeveloping all those tedious CAD and graphics capabilities. And it's **FAST!**

It has the complete functionality of the acclaimed ChoiceCAD for Windows 2D CAD package. Here are just some of its features:

- *FAST* zooms, pans, redraws.
- Full *CAD Engine* with double precision coordinate database.
- *Drawing commands:* points, lines, arcs, parallel lines etc.
- *Editing commands:* rotate, move, fillet, trim, undo etc.
- Powerful *text handling* and use of TrueType fonts.
- Definition and insertion of *symbols and objects.*
- AutoCAD® *DWG and DXF* file read and write.
- And hundreds of other powerful CAD features like *65,000 lockable layers.*
- Insertion of *unique entity IDs* for connection to an attribute database.
- *Query* of drawing details like "What's the nearest entity".

Just £399.00 (excl.)
(£475.88 incl. P&P and VAT)

SOFTcover *international* LTD

320 Old Brompton Road • London SW5 9JH
Tel: (0171) 259 2100 • Fax: (0171) 373 6368

CIRCLE NO. 404

Protecting your ***Investment***

'Imagine what two rabbits can do'

YOU SELL JUST ONE, AND ... IT MULTIPLIES.

Your success is our success

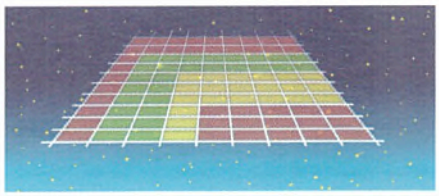
Protect your software for as low as £6.00
CALL BL COMPUTER SECURITY TODAY FOR YOUR *FREE* 30 DAY DEVELOPER PACK.

We support Dos, Windows, Win95, NT, Sco Unix, OS2 and Novell.

Call us on (44) 0181-343 0734
By Fax on (44) 0181-346 2672

Emil on: adeneli@cix.compulink.co.uk

CIRCLE NO. 403



assign the return value of the function to a temporary pointer, and then test it before assigning it to `FDataPtr`.

If we are just adding new rows we can simply extend the memory block and fill in the new data. However, if we are adding columns we must shift the blocks of existing data that represent each row, leaving spaces in between for the new columns. We do this in the `ReSize` method by calculating two pointers: one to the start of the existing row and one to the position of the row in the resized array. The new position can be calculated by multiplying the number of new columns by one less than the current row number.

Because this method will not be called as often as `GetVal` and `SetVal`, I've opted to place the pointer arithmetic in separate functions. The code in `GetPtr` is identical to that in the two data access methods. `IncPtr` will take a pointer and increment it by a given amount, then check to see if we've crossed a segment boundary. If so, we adjust the selector accordingly. Incrementing an existing pointer is faster than calculating it from scratch, so we use this when we are looping through a section of the array and moving data.

If we are making the arrays smaller we will also have to move the row data. In this case we will overwrite elements from the columns that are being deleted. This will need to be done prior to calling `GlobalReAllocPtr`, so that the data are not truncated before they are rearranged.

Property market

To top it off, this object employs a very useful feature of Delphi, the *array* property, to simplify access to the data. We can declare a property called `Data` that is indexed much like a standard Pascal array:

```
property Data[Row,Col: Longint] :
  Double read GetVal write SetVal;
default;
```

By using the `read` and `write` declarations for a property, the value for this property is retrieved and set using the existing `GetVal` and `SetVal` methods. The row and column indexes of `Data` must be of the same type and order as the parameters of the two methods. Other fields in the object, such as number of elements, can be made read-only by mapping them onto a property and not providing a write method.

If we use the `default` directive with an array property, Delphi allows us to access the property without specifying the property name `Data`. If `DataArray` is an instance of our object, we can access the data using the expression `DataArray[i,j]` rather than `DataArray.GetVal(i,j)`.

With this last enhancement we now have a dynamic array that can be used exactly like a standard Pascal array. If you have large libraries of Pascal code that manipulate standard arrays you can now simply drop in this object to greatly increase the maximum data size. All that needs to be done is to add calls to the `Create` and `Free` methods. ■

In a previous life Warren Kovach was a palaeobotanist. He now writes and markets statistical software. You can reach him at WarrenK@kovcomp.demon.co.uk.

```
begin
  GetValAsString :=
    FloatToStrF(GetVal(Row,Col),Format,Precision,Digits)
end;

function THugeDoubleArray2d.GetPtr(ElemOffset:Longint) : PDouble;
var
  ByteOffset, MasterOffset: Longint;
begin
  ByteOffset := Pred(ElemOffset) * FElementSize;
  MasterOffset := Longint(PtrRec(FDataPtr).Ofs)
                    + LongRec(ByteOffset).Lo;
  Result  := Ptr(PtrRec(FDataPtr).Seg
             + (LongRec(ByteOffset).Hi
             + LongRec(MasterOffset).Hi)
             *  SelectorInc,
                LongRec(MasterOffset).Lo);
end;

procedure THugeDoubleArray2d.IncPtr(var P : PDouble;
                                    IncAmount:Longint);
var
  SaveP: Pointer;
begin
  SaveP:= P;
  Longint(P):= Longint(P)
           + (IncAmount*FElementSize);
  if PtrRec(SaveP).Seg <> PtrRec(P).Seg then
    if Abs(IncAmount) = IncAmount then
      PtrRec(P).Seg := PtrRec(SaveP).Seg +
        ((PtrRec(P).Seg - PtrRec(SaveP).Seg) * SelectorInc)
    else
      PtrRec(P).Seg := PtrRec(SaveP).Seg -
        ((PtrRec(SaveP).Seg - PtrRec(P).Seg) * SelectorInc);
end;

function THugeDoubleArray2d.ReSize(Rows,Cols : Longint) : Boolean;
var
  NewPtr : PDouble;
  i,j,
  NewRows,NewCols,
  NewNumElem,Offset,
  ByteOffset,MasterOffset: Longint;
  SourcePtr,
  DestPtr    : PDouble;
begin
  NewNumElem := Rows*Cols;
  NewRows := Rows-NumRows;
  NewCols := Cols-NumCols;
  { if reducing the number of columns, move rows up to
    overwrite columns that will be deleted }
  if NewCols < 0 then
    for i := 2 to NumRows do begin
      { Point to start of row... }
      Offset := (Pred(i)*NumCols) + 1;
      SourcePtr := GetPtr(Offset);
      { ...and to place where data should be moved }
      Offset := Offset + (Pred(i)*NewCols); { NewCols will be negative }
      DestPtr := GetPtr(Offset);
      for j := 1 to NumCols do begin
        { copy data & set old element to 0 }
        DestPtr^ := SourcePtr^;
        SourcePtr^ := 0.0;
        if j <> NumCols then begin
          IncPtr(SourcePtr,1);
          IncPtr(DestPtr,1);
        end;
      end;
    end;
  NewPtr := GlobalReAllocPtr(FDataPtr, NewNumElem
         * FElementSize, gmem_Moveable or gmem_Zeroinit);
  { raise an exception if failed }
  if NewPtr = nil then OutOfMemoryError;

  FNumElem := NewNumElem;
  FDataPtr := NewPtr;

  { if adding new columns then we must now move blocks
    of data down to allow space at ends of each "row"
    for new columns}
  if NewCols > 0 then
    for i := NumRows downto 2 do begin
      Offset := (Pred(i)*NumCols) + NumCols;
      SourcePtr := GetPtr(Offset);
      Offset := Offset + (Pred(i)*NewCols);
      DestPtr := GetPtr(Offset);
      for j := NumCols downto 1 do begin
        DestPtr^ := SourcePtr^;
        SourcePtr^ := 0.0;
        if j <> 1 then begin
          IncPtr(SourcePtr,-1);
          IncPtr(DestPtr,-1);
        end;
      end;
    end;
  FNumRows := Rows;
  FNumCols := Cols;
end;
end.
```

Listing 4 - Unit containing the THugeDoubleArray2d object

take advantage of

Windows 95™ & Visual Basic 4™

with the new 32 bit OCX controls from

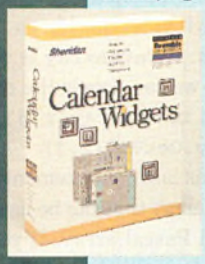
Contemporary Software

CONTEMPORARY SOFTWARE

SHERIDAN CALENDAR WIDGETS™

Reusable Components for Date & Time Management

Calendar Widgets™ is perfect for use in developing any Windows™ based application that needs to display, select and manage dates, times and related information.



- Visual Basic™ Custom Controls (VBX) and 16 and 32 bit OLE Controls (OCX)
- YearView control displays one year at a time to which you can add colour and pictures
- MonthView displays one to three months for quick and easy selection of dates.
- DayView is the perfect control when a daily time schedule is needed.
- Specify the input and format masks with DateCombo
- Controls can be bound to any database supported by the Visual Basic™ data control

SHERIDAN CALENDAR WIDGETS £99

SHERIDAN sp_ASSIST™

Enhanced Development Environment for Microsoft SQL Server® and compatible databases

This exciting new product is a multi-user development environment for creating SQL Server databases and coding client/server applications with SQL Server. Build on the strength of Visual Basic and SQL Server with sp_Assist.

- Code includes encapsulation of all SQL key words, executable statements and system stored procedures.
- Use drag and drop to quickly and easily Generate Code for SQL Server and Visual Basic.
- Use Data Wizards to create stored procedures, queries and tables.
- Database Administration features allow maintenance of access rights to SQL databases and SQL objects.
- Powerful SQL Server Interaction features mean that sp_Assist is the only tool you'll ever need for SQL Server to manage and code client server applications.

SHERIDAN SP_ASSIST £450

SHERIDAN DESIGNER WIDGETS™ 2.0

Create the look and feel of today's State-of-the-Art applications

With this new release of the best selling Designer Widgets, you can easily develop application interfaces similar to those of the most popular commercial applications.

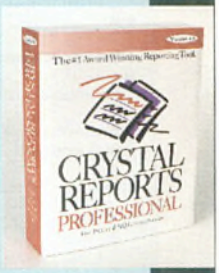
- Visual Basic™ Custom Controls (VBX) and 16 and 32 bit OLE Controls (OCX).
- With the Notebook Wizard and the new Notebook control you can easily build controls using a 'notebook' metaphor.
- New features of the Dockable Toolbar include automatic resizing and the ability to contain other controls including combo boxes.
- The FormFX control allows you to customise the look and behaviour of your form.
- Use the Index control to lay out your controls with the look, and ease of use, of index cards.

SHERIDAN DESIGNER WIDGETS 2.0 £99 UPGRADE ONLY £45

CRYSTAL REPORTS PROFESSIONAL 4.5

The number one Reporting Tool for the professional developer

Find out why Microsoft™, Borland™ and more than 80 other major software companies have chosen to include Crystal Reports in their programs. The new version 4.5 now includes:



- Visual Basic™ Custom Controls (VBX), 16 and 32 bit OLE Controls (OCX), Delphi VCL control and 32 bit DLLs
- Runs on Windows™ 3.1, Windows NT™ and Windows 95™
- 40 new (>80 total) print Engine Calls for maximum printing flexibility
- 2-10 times faster report processing
- Fully integrated and tailored graphing
- Report distribution expert with free runtime
- Special upgrade price for Visual Basic™ developers

CRYSTAL STANDARD 4.5 £149 CRYSTAL PROFESSIONAL 4.5 £299
CALL FOR UPGRADE PRICES

FARPOINT BUTTONMAKER

Design buttons that are unique to your application

ButtonMaker is the best tool for enhancing any application by providing unlimited options for a button's appearance. ButtonMaker also includes a customisable Balloon control that displays a help bubble when a mouse is positioned over the control.

- Visual Basic™ Custom Controls (VBX), 16 and 32 bit OLE Controls (OCX) and 32 bit DLLs.
- The ability to define cells allows you to customise individual elements of a button.
- Add multiple bit maps, text and hot spots to any button.
- Define multiple segments and animate your buttons.

FARPOINT BUTTONMAKER £75

Order today

CIRCLE NO. 390

for more information call us now on

Tel: 01727 811999

CONTEMPORARY Software

Contemporary Software, Abbey View, Everard Close,
St Albans, Hertfordshire AL1 2PS
Tel: (01727) 811999 Fax: (01727) 848991
E-Mail: cssales@contemporary.co.uk

CONTEMPORARY SOFTWARE A DIVISION OF INTERSOLV PLC

The Web is not limited to static HTML pages. **Peter Collinson** demonstrates step by step how to call programs from a Unix Web server via the Common Gateway Interface.

Using programs as Web pages

If you provide a Web server, then at some point you are going to want to display more than just static pages to the surfing customer. You will perhaps want them to access some other program on your system or maybe you will provide a form that the user can use to send you some information. What is needed is a way for the Web Server to execute a program when the user selects the appropriate Universal Resource Location (URL).

Luckily, the designers of the Web foresaw this need and designed a protocol called the *Common Gateway Interface* (CGI). This is used by the Web server daemon to interact with a program on your machine. It defines how the server calls the program and what responses the program should make to communicate back to the server. The good news is that most servers for UNIX (certainly all the servers I have seen) do follow the standard CGI rules, so CGI programs are portable from server to server. (See *Beyond a browse* in the February and March '95 issues of *EXE* for creating DOS and Windows CGI programs).

Of course, on a UNIX system, a program can easily be a script written in some arbitrary language. Many people write CGI programs in Perl because the language is good at cracking apart some of the common text formats. I am using the standard shell for demo purposes in this article, largely because it's more readable by mortals.

Kevin February

Output from programs

The CGI program is launched by the server and sends data back to it. The server then *executes* the text stream from the CGI program, perhaps sending the data from the program directly to the client, or maybe doing some of its own processing before responding to the client.

The server provides information to the CGI program using three sources: environment variables, command-line information and the standard input channel. The CGI

DEVELOPERS TRAINING

C++ FOUNDATION
No previous C experience? Good. This course is designed specifically for non-C programmers and leads on to our VC++ for WINDOWS course. *5 days* £1,200 + VAT

VISUAL C++ 2.0 PROGRAMMING
You're a C developer and you want to make effective use of C++ language extensions. This is the course for you.
5 days £1,200 + VAT

VISUAL C++ 2.0 FOR WINDOWS
A thorough, practical course covering MFC programming. Still using VC++ 1.5 and 16 bit WINDOWS? Treat yourself.
5 days £1,200 + VAT

MS WINDOWS: 32 BIT DEVELOPMENT – VC++
You're a C++ programmer and need to know about NT and Windows '95. Maybe you want Solution Developer accreditation. This is the course for you.
5 days £1,200 + VAT

VISUAL BASIC PROGRAMMING
A practical course for developers starting to build and support systems written using the Professional Edition.
5 days £1,200 + VAT

VISUAL BASIC 4.0 NEW FEATURES
OLE automation, reusable objects, OCX's, 32-bit support, programming under WINDOWS 95 and more.
2 days £560 + VAT

VISUAL BASIC DATABASE PROGRAMMING
You have a working knowledge of VB but need to know more about the JET engine and accessing other data sources. *2 days* £560 + VAT

ADVANCED VISUAL BASIC PROGRAMMING
DLL's, API calls, making OLE work, custom controls, managing WINDOWS resources, and more.
3 days £770 + VAT

WHY CHOOSE RICHFORDS?

- Expert course leaders – actively involved in development work
- Our own courseware – designed to satisfy real-life training needs
- Pre-course questionnaires – establish the right course for you
- Small classes – maximum six delegates
- Developer spec. machines – no sharing!
- Access to course leaders – before and after the course

Which other company is able to offer better value?

CALL 0171-922 8819

. . . for full course outlines, dates and availability.

RICHFORDS
South Bank Technopark 90 London Road London SE1 6LN
All trademarks acknowledged

CIRCLE NO. 386

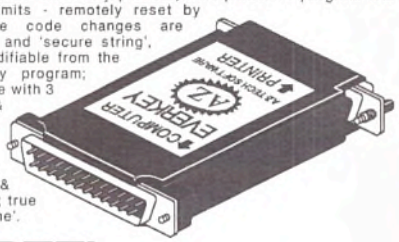
Software COPY PROTECTION and REGISTRATION/DISTRIBUTION CONTROL

EverLock *Software Copy Protection*

SOFTWARE-ONLY COPY PROTECTION System: Can be applied to virtually any program without code-change; transparent to application program; positive security against unauthorised use or theft; Date, Execution Count, Network con-current Limits & program Access/User-Data Flags - remotely reset by telephone; Dynamic File Compression; supports all DOS based programs including WINDOWS, WIN'95, 4GL & DOS Extender programs; Remote product support; Product serialisation and User Registration information; easy to use & support in-the-field; up-grade support of products; Site Licence and options for media restriction; uses standard diskette media, supports all Hard/Floppy disk formats; CD-ROM protection; all networks supported; mass-duplication; established in 1986 & used worldwide in over 100 countries.

EverKey *Hardware Copy Protection*

TOTAL Hardware Compatibility; Single-Wire-Zero-Load, no PC power requirement; patented design not available in ANY other Key product; transparent to program and End User; Date & Execution Limits - remotely reset by telephone; no program source code changes are necessary; Product Serialisation and 'secure string', up to 1024 bytes + 16 bytes modifiable from the program!; protects virtually any program; easy-to-use programming software with 3 User Levels; Dynamic encryption & File Compression; Developer defined Expire and Unauthorised Messages; support for most networks; optional .OBJ code (encrypted!) for customisation & linking with source code changes; true compatibility with ANY PC or 'clone'.



FREE!

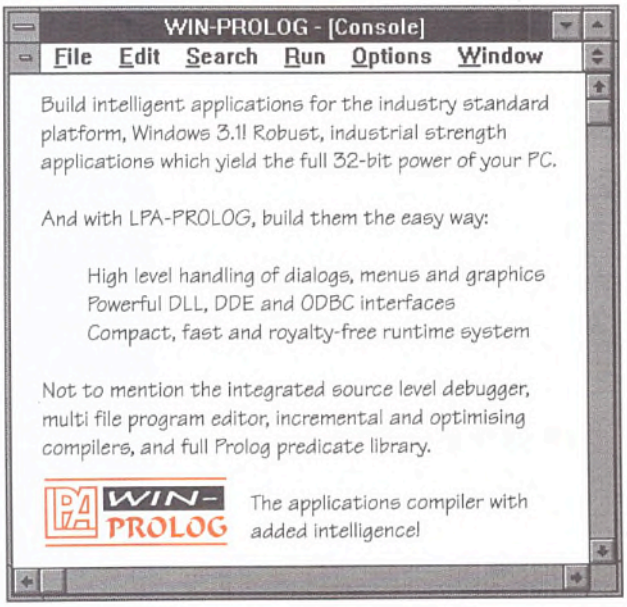
GLYN WILLIAMS & ASSOCIATES

gwa COMPUTER SECURITY

3, HIGH STREET, STUDLEY, WARWICKSHIRE. B80 7HN. UNITED KINGDOM
Tel: 01 527 85 3322 Fax: 01 527 85 4411
BBS: 01 527 85 7878 (8, 1, None Parity)
Internet: gwa@gwassoc.demon.co.uk

CIRCLE NO. 398

Add Intelligence to your Applications!



Logic Programming Associates Ltd
Phone (US Toll Free): 1-800-949-7567
Phone: +44 181 871 2016 - Fax: +44 181 874 0449
Email: lpa@cix.compulink.co.uk - Compuserve: 100135,134

CIRCLE NO. 391

program communicates back to the server by writing data to its own standard output channel. The server arranges to capture the program output and deal with it appropriately.

Let's look at the output from the CGI program first. The data sent by the CGI program to the server is in text form. The text starts with a small header: some text lines terminated with a blank line. Any line which is not recognised as a header line is sent back to the client. There are currently three recognised header lines: `Content-type`, `Location` and `Status`.

The `Content-type` header line is used to specify the MIME type of the document that the CGI program is generating. At this point, you need to understand that the Web makes use of the Multi Media Mail Extensions (MIME) to provide type information for the various classes of data that may be carried in the normal text byte stream. MIME allow Web clients to receive some text (and interpret it as text), some HTML code, a GIF image, an audio file, a QuickTime movie or whatever.

Here's a simple shell script that can be used to display a page that is today's date:

```
#!/bin/sh
echo 'Content-type: text/plain'
echo
/bin/date
```

Notice how the second `echo` command is used to ensure that there is a blank line after the header. The output from this script is simply three lines of text; the server will interpret the data as a plain text page. It will be formatted in a fixed-width font and sent onto the client.

Of course, using a fixed-width font is perhaps not too exciting. We can improve the look a little by generating HTML code:

```
#!/bin/sh
echo 'Content-type: text/html'
echo
echo '<HTML><HEAD>'
echo '<TITLE>The Date</TITLE>'
echo '</HEAD><BODY>'
echo '<H1>Date and Time</H1>'
echo 'The date is'; /bin/date
echo '</BODY></HTML>'
```

We made sure to place the body of the text in quotes because we don't want the shell to get too excited doing data redirection (remember, that's what the UNIX shell normally does with > and <). It could all be *one* quoted section, but I've avoided doing that to make it easier to understand.

Notice how we obey the rules for HTML document formatting in the text that is returned. After all, the text will be sent to the client as HTML for local formatting. The page contains a title and body section in the same way as a regular static page. I'll concede that the text that the script displays is still not that exciting, but does give you a glimpse of what might be done.

People want to write in shell or Perl because it's convenient

The use of other MIME types is certainly possible from the CGI program; you can send audio or graphics if you wish to. Rather than generating a document for the user to look at, you may be using a CGI program to provide the user with navigation instructions. You may wish to point the user at another page in your server, or at another URL. The `Location:` header can be used to do this. A simple shell script might be:

```
#!/bin/sh
echo 'Location: file.html'
echo
```

This will look for a file called `file.html` in the same directory as the URL of the script being executed. This can be a little counter-intuitive. Of course, you can use a full URL as the location if you wish.

Finally, you can return a *standard* error message from the CGI program by using the `Status:` directive. An example might be:

```
#!/bin/sh
echo 'Status: 403 Forbidden'
echo
```

This passes a numeric error code back to the client, and a text message. There is a set of standard error messages located at the URL, see *http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTRESP.html*.

Arguments to CGI programs

CGI programs that simply output data are rare. You mostly want to write a CGI program to provide some access to something like a database lookup system or to decode data from interactive forms. Let's start by looking at the first case, where the CGI program provides a *gateway* to some other system. Incidentally, this is somewhat chicken-and-egg so bear with me.

The gateway program will have a URL by which it is invoked. We need to provide the gateway with a search string, and this is done by adding a question mark and some text after the URL. If the text contains

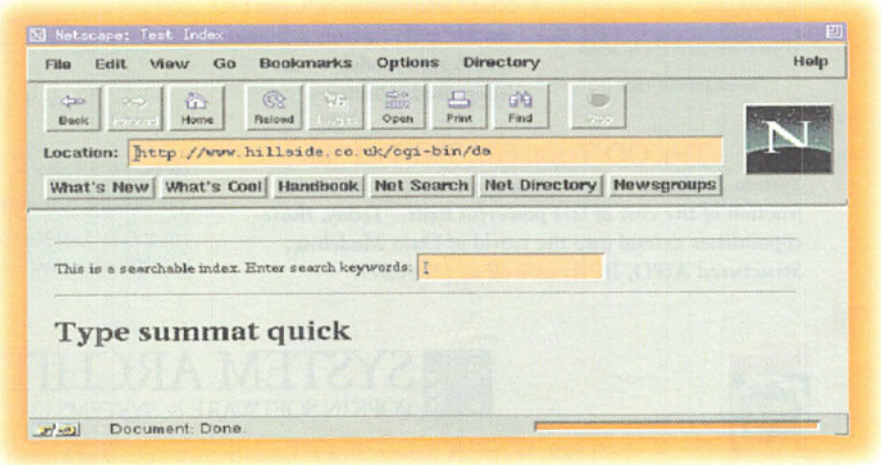
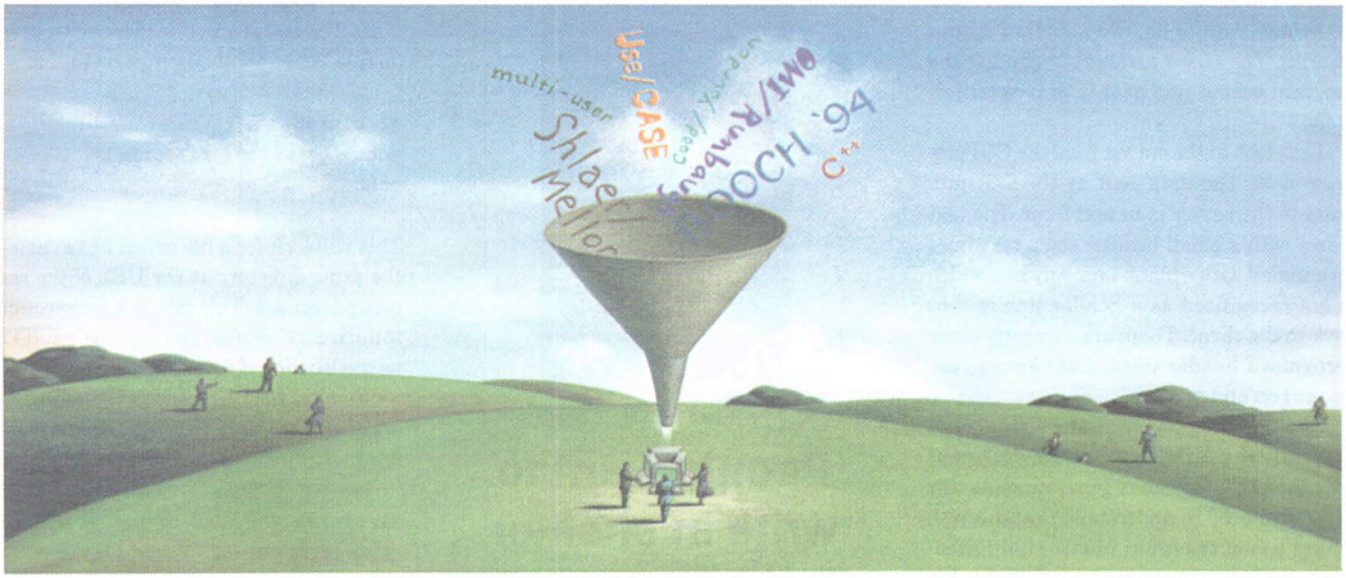


Figure 1 - A CGI program called with no argument

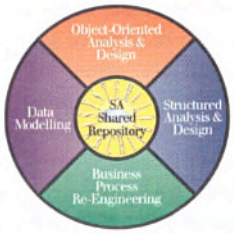


The Multi-User Object-Oriented Tool The Whole World Is Talking About.

System Architect. Discover What An Affordable OO Application Development Tool Can Really Do.

All The Power You Need

If you want to enter the OO world but aren't sure you can afford the learning curve, System Architect is the solution. It's the only affordable client/server application development tool with a multi-user repository for the leading OO methodologies.



SA gives you all the benefits of the best repository-based data modeling tool, plus the functionality of a complete object-oriented application development tool.

Some of System Architect's OOA&D features include:

- *Generate and reserve C++ header and .CPP files.*
- *Generate schema from class definitions with the SA Schema Generator for Oracle, SQL Server, SYBASE, Informix and over 14 other SQL and non-SQL databases.*
- *Create ER diagrams from class models automatically.*
- *Share information between OO models and structured analysis and design models in an integrated repository.*

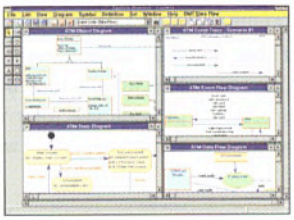
System Architect Supports:

- OMT/Rumbaugh
- Coad/Yourdon
- Jacobson Use-Case
- Booch '94
- Shlaer/Mellor*

(*Call for details)

50,000 Users Can't Be Wrong

That's the number of satisfied SA users worldwide and it's growing every day.



◀ Comprehensive support for OMT.

Comprehensive support for Booch '94. ▶

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

"...(the) folks at Popkin seem willing to do what it takes to provide the best support possible for their (OO) methodologies."

- T.L. (Frank) Pappas, From an article in IEEE Computer Magazine entitled "Object-Oriented Technology: PC Based OOA&D Tools, Part II", July 1994

The OO Tool For The Future

System Architect offers comprehensive capabilities at a fraction of the cost of less powerful tools. Today, those capabilities extend into the world of Data Modeling, Structured A&D, BPR, as well as OOA&D.

Software Development Productivity Award 1990

SYSTEM ARCHITECT™
POPKIN SOFTWARE & SYSTEMS LTD

Database Programming & Design 1995

118-120 Warwick Street Royal Leamington Spa Warwickshire England CV32 4QY

Tel: (01926) 450858 • Fax: (01926) 311833

Australia (1-800) 658660 • Benelux 31-3406-65530 • Brazil 55-11-535-5200 • Chile 56-2-695 3330 • Columbia 57-1-218-8877 • Denmark 45-45999-9300
Germany 49-6151-866620 • Italy 39-49-8700366 • Japan 81-462-8700366 • Korea 212-751-1109 • Malaysia 60-3-757-1806 • New York 1 212 571 3434
New Zealand 61-02-346499 • Peru 5114-417828 • Spain 34-3-415-7800 • Sweden 46-8-626-8100 • Switzerland 41-61-6922-666

Microsoft Windows

PowerSoft Code Partner

CIRCLE NO. 392

spaces, then they are translated to plus characters. For example:

```
http://www.eg.co.uk/bin/x?look+up
```

Here `bin` is assumed to be a directory holding the CGI programs, `x` is the program being executed with a lookup string of `look up`. When the CGI program is called, it is supplied with two arguments `look` and `up`. Also, the environment variable `QUERY_STRING` will be set to `look+up`.

Well, you can type in the URL, the question mark and the argument but this is not too user friendly. We normally get the client to ask the user for a search string. This is done by calling the CGI program twice. On the first pass, the program presents an index and asks for a string. On the second pass, the program now has search arguments and performs the lookup.

The CGI program can tell what it should do by looking at the number of arguments with which it is called. If the program is called with no arguments, it will output an HTML index page that contains the tag `<ISINDEX>`. In turn, this provokes the browser to provide an input box with a phrase like: *'This is a searchable index. Enter search keywords: <type-input-here>'*. You can see the index page from the example below in Figure 1.

The user supplies some text and hits return. The text is appended to the URL using the question mark notation and the program is called again for the second pass, this time with arguments. Here's all that in a script:

```
#!/bin/sh
echo 'Content-type: text/html'
echo
if [ $# -le 0 ]
then

echo '<HEAD>
<ISINDEX>
<TITLE>Test Index</TITLE>
</HEAD><BODY>
<H1>Type summat quick</H1>
</BODY>'

else

echo '<HEAD>
<TITLE>Test output</TITLE>
</TITLE>
</HEAD><BODY>
<H1>You typed</H1>'
echo $@
echo </BODY>

fi
```

The TCP/IP stacks on PCs often cannot cope with the `identd` request

We have seen that the query is coded by replacing any spaces by plus characters. There is some additional escape coding to allow any character to be transmitted using the ASCII set. A character may be encoded as a percent character followed by two hexadecimal values.

Passing the arguments in the URL using the method described above is referred to as the `GET` method in all the WWW literature. You can create a form that passes data by this method too. However, it's more usual for forms to use the `POST` method because the amount of information that is passed from the form into the CGI program is greater. The `POST` method means that the CGI program will use its standard input channel to read data originally entered by the user on the form. The size of the data sent to the CGI program by the server is supplied in an environment variable called `CONTENT_LENGTH`.

Forms behave in a different way from the gateway example described above. A form is specified by an HTML page that describes the page image, including the fields that the user can change, perhaps by pressing buttons, perhaps by entering text. The CGI program is activated when the user presses an enter button. The data from the form is sent into the CGI program using the character encoding scheme described above.

Of course, the form can contain several input areas and so we need a naming mechanism to identify where data originated on the form. This is simple: every input area is given a name by the author of the page, and the data from that area is sent as a `NAME=contents` pair to the CGI program.

All the data from the form is sent in one chunk with each `NAME=contents` pair separated from its successors by a question mark. No UNIX-friendly newline characters here. This does mean that the `GET` and `POST` forms use exactly the same data encoding method, it's just that the data is passed into the CGI program using a different mechanism. The standard Perl CGI library contains routines for pulling this chunk of data apart into an associative array.

Environment variables

The CGI program is provided with information that the server places in the environment of the child process that is running the program. Environment strings are of the form `NAME=value` and can easily be obtained from programs or programming languages. Here's the output from a test program run on my machine:

```
SERVER_SOFTWARE = NCSA/1.3
SERVER_NAME = www.hillside.co.uk
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
```

This is all pretty self-explanatory stuff that just supplies basic information that the server knows. Beware that the environment variables shown below are those supported by my server, NCSA's `httpd`. Things may be different for other servers, check your documentation.

```
REQUEST_METHOD = GET
```

As we have seen, the CGI program can be invoked using two 'methods': `GET` and `POST`. This environment variable tells you which is being used at the moment. CGI programs should expect to be invoked using either method and should fail cleanly if one method is not supported. It's quite usual to exit from programs that process input from forms when a test on `REQUEST_METHOD` shows that it's not set to `POST`. It's needed in case someone decides to invoke the CGI program directly by specifying the URL.

```
HTTP_ACCEPT = */*, image/gif,
              image/x-xbitmap, image/jpeg
HTTP_USER_AGENT = Mozilla/1.1N
              (X11; I; BSD/OS 2.0 i386)
```

These lines are wrapped for printing purposes. They were originated by the client and sent to the server in the MIME header. Lines like this are placed in the environment using their original MIME keywords preceded by

OXFORD COMPUTER CONSULTANTS

the new choice for your software needs

Help Browser 2

–Adds two new buttons to the Winhelp Button Bar Browse and Find Text . Helps one navigate easily through the layers of a help file

£49*

WinHelp Office

RoboHelp,
WinHelp Video Kit,
WinHelp Tool Kit,
WinHelp Hyper Viewer,
Mastering WinHelp,
Moving to WinHelp

£595*

Quarterdeck

Cleansweep for Windows	£	39
EMM 386 V7.5 3.5"		150
Desqview X V2 3.5"		175
Web Navigator		59

Asymetrix

3-D FX	£	95
Toolbook V3		175
Multimedia Toolbook CD-ROM		559

Borland

Delphi	£	175
Delphi Client Server		699
Database Engine		220

Please ring for other products!

£265*

ForeHelp 2.0 – Award winning WYSIWYG Help Authoring System for Windows that lets you work in an actual hypertext environment

£295*

RoboHelp – A powerful Help Authoring tool for Windows and Windows NT. RoboHelp guides you through the entire process of creating Help systems and Electronic Hypertext documents.

***All prices Include post and packaging**

OXFORD COMPUTER CONSULTANTS
Windows Consultancy Specialists

Barclays Venture Centre,
University of Warwick
Science Park,
Coventry, CV4 7EZ
Tel: 01203 690934
Fax: 01203 410156

All prices, specifications & offers subject to change without notice

AMERICAN EXPRESS VISA MasterCard

Contact Matt Boyle direct on 01203 690934

HTTP_. The HTTP_ACCEPT variable gives the MIME types that the client will accept. The User Agent here is Netscape 1.1N using X11 running on my BSD/OS system.

```
PATH_INFO =
PATH_TRANSLATED =
```

A CGI program does not have to be the last name in the URL, it's possible to have a URL like:

```
http://machine/dir/CGIPROG/a/b/c.
```

When CGIPROG is executed, the /a/b/c in the example above will be placed in the environment in the variable PATH_INFO. The server always manipulates pathnames of files for security reasons, and will set the PATH_TRANSLATED variable to the full pathname relative to the place on the system where your documents are stored. In my case, the variable will be set to /var/www/docs/a/b/c. This allows you to write scripts which intercept URLs and redirect them intelligently to some other place in the tree.

```
SCRIPT_NAME = /cgi-bin/test-cgi
```

The SCRIPT_NAME variable contains the pathname of the program being executed, allowing a recursive call if needed.

```
REMOTE_ADDR = 192.88.50.3
REMOTE_HOST = wooded.hillside.co.uk
REMOTE_IDENT =
```

This gives you details of the host that is calling you. You should always get a REMOTE_ADDR; after all, the server needs that information to send data back to the remote client. You may get a REMOTE_HOST, but this depends on a successful reverse lookup in the Domain Name Service (see the DNS article in *EXE* August '95). There now seem to be boundless reasons why this doesn't work. A great many contacts made to my server now come in without the name server being able to make a successful reverse lookup of their IP addresses.

The REMOTE_IDENT variable will be set if you have enabled support for RFC932 Identification in your server and the remote host is running a version of identd to serve the identification information to you. I turned on the RFC932 code in my server for about a month and found that this created more trouble than it was worth. First, very few sites that are able to run identd actually do, for whatever reason: perhaps security or perhaps anonymity. Second, the TCP/IP stacks on PCs often cannot cope with the identd request and Web access to your site may time out. As a result, it is basically not possible to find out who is calling your service. Actually, I ask politely, and some people respond by sending me email via my Visitor's Book page.

Security is a compromise between safety and convenience

Security

There was once a time when people were *blasé* about security. Things have changed. First, you will find that the server will not execute a CGI program from anywhere in the tree, it has to be placed in a known script directory. Second, if you are putting a program into a position where anyone can type any text into it, then you need to be aware that someone will attempt to use that to break into your machine.

As an example of this, take the 'Type summat quick' example above and replace the line that reads:

```
echo @$
```

by a simple command:

```
cat $@
```

Ok. You now have a script that will display a file on the client's screen. There may be a very good reason for this script to exist, maybe you are displaying data from files for the user. It would be usual to change into some directory and make the index page display the ls output for the directory. Then the user can select a file using a single filename key typed into the selection box. This is a perfectly innocuous simple application. Unwittingly, you have also provided anyone with the ability to obtain any file on your machine. Rather than typing the single file name that you were expecting, our hacker simply types /etc/passwd into your request box and is rewarded by a screen containing the contents of your password file. Hopefully, when your WWW server runs, it doesn't run as super-user, so the hacker can only get to publicly accessible files on your machine. However, you probably don't want the hacker to get to these files: you generally want to minimise access by strangers to files on your machine.

Well, one reaction to this might be — don't write CGI scripts in interpreted languages that are prone to attack in various different ways. The trouble with this approach is that people *want* to write in shell or Perl because it's convenient. As I frequently say, security is a compromise between safety and convenience.

What we actually need to do is to ensure that the data that the user types is validated in some way. For example, you can easily test for a full pathname in the script above and refuse to deliver files that start with a slash. Actually, it's better to prohibit filenames with a slash anywhere so that ../../etc/passwd is also prohibited. All this will disappoint the hacker. Shame.

It's standard practice in Perl to validate all input to ensure that it contains only alphanumeric characters, underscore, minus, plus, space, tab, slash, backslash, the 'at' sign and the percent character. This eliminates a great number of characters that might be useful for someone attempting to subvert the actual script by supplying contents to variables that are parts of the Perl language.

Further reading

I've again used the excellent *Managing Internet Information Services* by Cricket Liu, Jerry Peek, Russ Jones and Adrian Nye. It is published by O'Reilly and distributed in the UK by International Thomson Publishing. Its ISBN is 1-56592-051-1.

There is also a certain amount of information on the Web: the best entry point to this is *http://hoohoo.ncsa.uiuc.edu/cgi/*. There are a bunch of gateways to be found on *http://www.w3.org/hypertext/WWW/Gateways.html*. The CGI archive can be found on *ftp://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa_httpd/cgi/*. You will also find there the Perl library for unpacking CGI arguments - it's called *cgi-lib.pl.Z*. ■

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as pc@hillside.co.uk, on WWW at http://www.hillside.co.uk (see the brand new 'Walking tour of Canterbury') or by phone on 01227 761824.

Have you ever tried to stop a program after its completion? **Francis Glassborow** looks at the issues involved.

Tidying up

First, a few words on compiler versions. Twice in the last few days I have had phone calls from students wanting to know where they could buy a copy of Borland C++ 3.0 because that was the package being used on their course. What could I say? Their chance of finding a kosher second-hand copy is remote and Borland has long discontinued stocking this version. If Universities and other academic institutions insist on using out-of-date software they should arrange for special distribution facilities with the original vendors of the software. Anything else encourages students to use unlicensed copies out of pure desperation.

On a more positive note, I was delighted to receive pre-release information about the next version of Microsoft's Visual C++ (see review in EXE October '95). Continuing with its confusing numbering scheme, this is VC++ 4.0. I have frequently criticised Microsoft for marketing C++ compilers that do not support current C++ language features, so it gives me great pleasure to note that this release will include support for both RTTI and namespaces. It will also include a version of the Standard Template Library. As the only generally available C++ compiler supporting namespaces till now has been Metaware High C++, this puts Microsoft near the front in language features support. I do not yet know if they have implemented namespaces correctly — High C++ doesn't. As it was only this summer that many members of X3J16 discovered that they had misunderstood some of the details when they voted in namespaces eighteen months previously, I won't be throwing any stones if the Microsoft version is not exactly right. At least we will have something to use while exploring the ramifications of namespaces. In this context I am disappointed by Microsoft's determination to restrict new language support to its 32-bit compilers. Supporting new language features creates problems for writers of lexical analysers, parsers and the like. Supporting 16-bit versions is only an issue for code generators. Microsoft's distinction between 16-bit and 32-bit development tools seems to me to be part of a larger commercial objective rather than a serious technical problem.

Something for thought

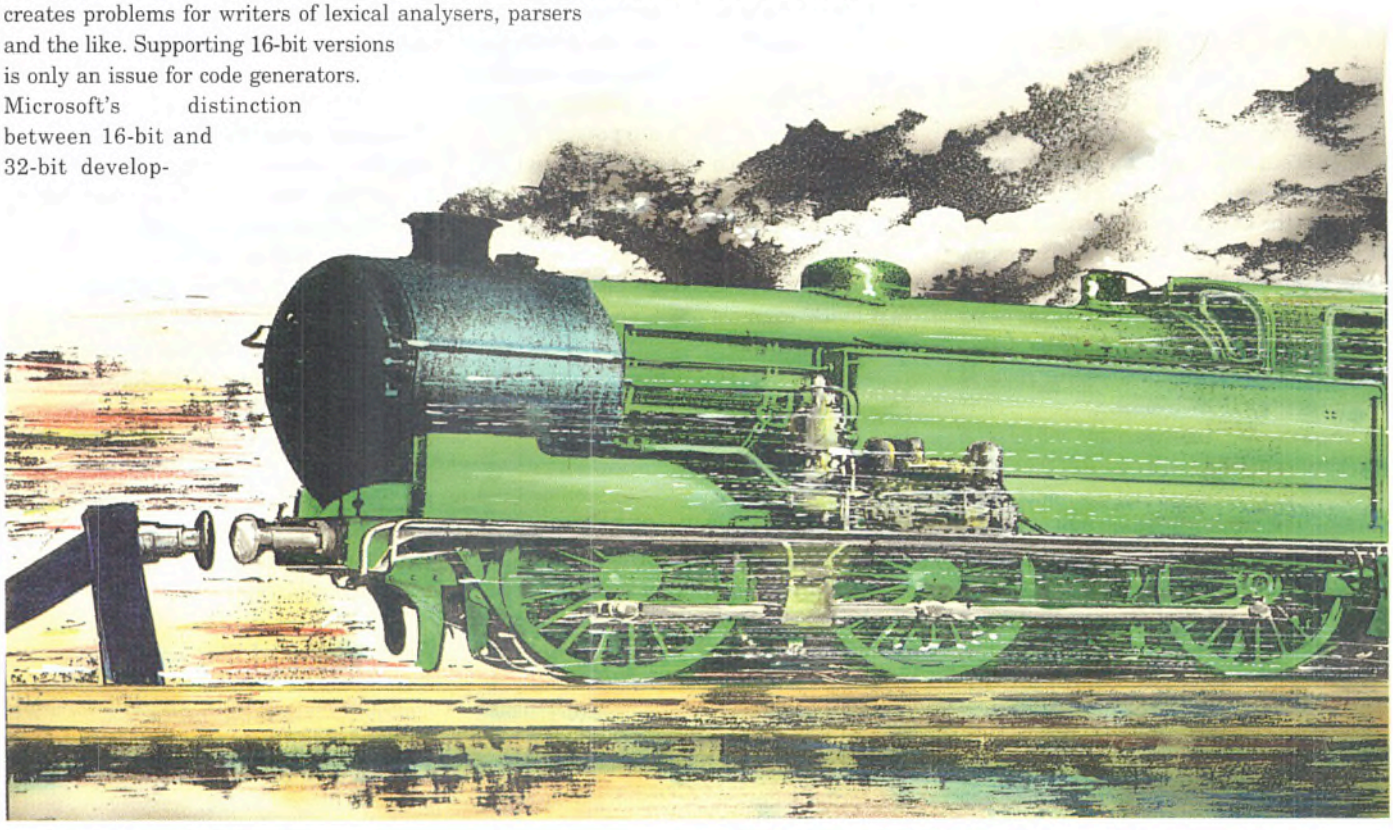
Rather than place some problem code at the beginning of my column, and then give an answer at the end, this time I am giving you a month to think about it. So here is a problem that I will explore next time. *What should be the return type of a function which must return success/failure information that cannot easily be abused or used out of context?* There is an excellent answer in C++, but it is not the one that may first come to mind. As a further clue, my solution works in C as well as C++ though not quite as effectively.

An object for instrumenting code

A few days ago a delegate on a C++ training course I was presenting asked me how he could inspect output generated after exit from `main()`. He was working on a windowing platform that automatically closed the program's window at the end of a program run. Quite reasonably he wanted to track calls of destructors that he had instrumented so that they sent a message on execution. What we needed was a quick fix so that the program would pause after completion — at the end of `main()` was too early.

First note that many debuggers cannot cope with code that has to be executed before entry to, or after exit from, `main()`. This means that setting a breakpoint in a destructor may not work.

Those familiar with 'registering' functions to be called on exit might try to write a function that pauses for input and then register it with `atexit()`. Something like this
(see overleaf):



```
void wait(){
        cout<<"Waiting. Press ENTER to continue";
        cin.ignore(1000,'\n');
}
int main(){
        atexit(wait);
        // rest of code
}
```

This almost works, but not quite. Functions registered by `atexit()` are executed in reverse order of registration *before* any static objects are destroyed. Remember that we want to pause the program just before it returns to the operating system — `atexit()` registered functions are executed too early.

What we need is a function that is called at the very last instant. The last functions executed by your program are the destructors of static (compile-time allocated) objects. There are actually some quite nasty technical problems lurking down here in the guts of your program, but as long as we stick to a single translation unit (.CPP file for those that are file-oriented), global objects are destroyed in reverse order of construction. As we use `iostream` objects they must be constructed first — before we intervene — and so be destroyed last — after we have finished. Try the following:

```
#include <iostream.h>
struct Pause {
    ~Pause(){cout<<"Waiting. Press ENTER to continue";
                cin.ignore(1000,'\n');
    }
} p;
// any other preprocessor directives
// and global declarations/definitions
int main(){
        //your code
}
```

This is just what we need for the simple problem that we started with. I think it is an excellent example of a minimalist object that simply encapsulates some behaviour. However, you might be in for something of a surprise if you assumed that `Pause` objects are of zero size. In order for all objects to have unique addresses, even dataless objects will have a size greater than zero. Currently the Standards Committees are considering the possibility of relaxing this rule in the case of sub-objects. The argument goes as follows:

Within any class definition, any member that is a dataless type will only provide behaviour. All such sub-objects of the same type will provide identical behaviour. If several dataless sub-objects share an address, the required behaviour will always be determinable in context. Allowing dataless sub-objects to share an address within an object can be important for memory efficiency. Note that we need not consider the complication of polymorphic dataless objects, because these do not exist (they must contain data to resolve their dynamic type — either a virtual function table pointer or something equivalent).

On even dates I am convinced that we should allow zero-sized sub-objects; the rest of the time I think not. A killer example showing that allowing them would produce ambiguous code would be helpful because without it their benefits are sufficiently seductive to attract efforts to mandate them into the language.

Back to class `Pause`: having written such a class it becomes a useful tool for investigating various aspects of our code. We can insert a pause anywhere in our code with:

```
{ Pause p;}
```

Many people believe that calling `exit(0)` from `main()` is equivalent to leaving `main()` with `return 0`. The ISO C Standard explicitly requires this to be true. Indeed, in Standard C, I think that any call of `exit()` is equivalent to returning from `main()`. This is not true in C++. The C++ `main()` is very different from the C version:

1) You cannot call `main()` from within a C++ program.
2) You cannot take the address of `main()` in a C++ program.
3) Falling out of `main()` in a C++ program is equivalent to executing `return 0` at that point (unlike falling out of any other function which is deemed equivalent to executing '`return`' — undefined behaviour unless the function's return type is `void`).
4) Executing either an implicit `return 0` or an explicit `return val` (where `val` is convertible to an `int` value) is equivalent to calling destructors for all `auto` objects in `main()` and then calling `exit()` — passing the return value to it.

Calling `exit()` anywhere in a program results in the `atexit()` registered functions being called before the destructors of globals being called. The destructors of `auto` (stack) objects are not called. Does this distinction matter? Yes. C++ objects can be holding dynamically allocated resources which are only released when their destructors are called. Try the following two variations:

Variation 1:

```
int main(){
    Pause p;
    return 0;
}
```

Variation 2

```
int main(){
    Pause p;
    exit(0);
}
```

One aspect of dataless objects that you might like to consider is whether it makes sense to allow them to be copied by either assignment or construction. Another question is that of the validity of a reference to such an object. And while we are asking these awkward questions, what about using one as a base class? Such questions are all part of class design. How should your class support your answers?

That is about all I have space for this time. Next time I will look at the problem I gave earlier. I will also suggest some further exploration that you can do with `Pause`. In particular I will consider the problem of order of initialisation in multi-file applications. This often surprises even experienced C++ programmers who have built a model of expectation without considering the deep problems that can exist.

Another issue that needs consideration is how to get out of a program cleanly without having to return all the way to `main()` via normal `return` statements. If I have the space, I might also suggest how C programmers can handle dynamic resources when using `exit()` deep in a program. Such methods are particularly important when writing applications for event-driven environments.

As always, I am interested in feedback on the above. If you have anything constructive to say, or you have any questions you would like answered in some future column, my email address is francis@robinton.demon.co.uk. In the current context, I would be interested in other classes that provide instrumentation and debugging support. ■

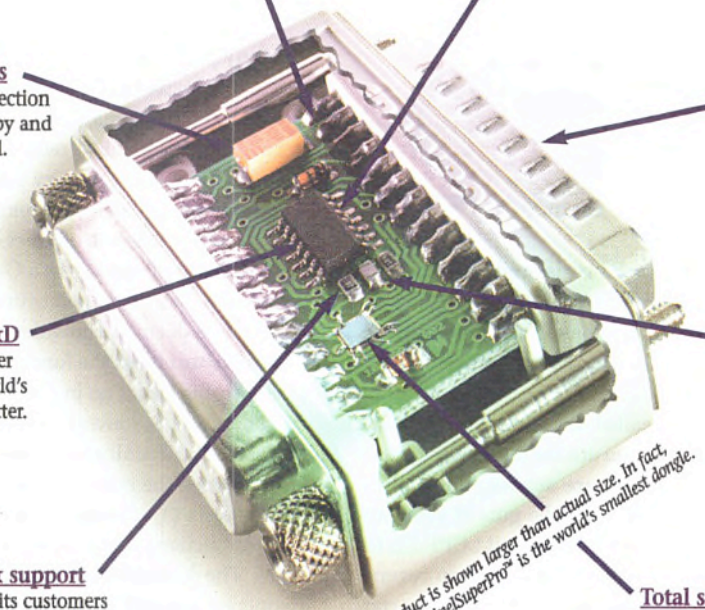
Reliability you can depend on
In 1994, Sentinel improved its industry leading reliability to over 99.985% – *far more reliable* than any other software protection product.

The industry's highest quality ISO 9002 Certified
Rainbow is the world's *only* software protection supplier with ISO 9002 certified quality standards.

yes NetWare Tested and Approved

Manage network licenses
NetSentinel™ is the *only* protection to undergo rigorous testing by and receive approval from Novell.

Truly transparent protection
Designed to go unnoticed by your customers, Sentinel does not interfere with hardware, peripherals or other software programs.



A substantial investment in R&D
In 1994 alone, Rainbow invested over $4,500,000 in R&D to make the world's leading software protection even better.

Compatible with your software
Our partnerships with Apple, Microsoft and IBM mean Sentinel protects software for any hardware or operating system.

Product is shown larger than actual size. In fact, the SentinelSuperPro™ is the world's smallest dongle.

Global service & support
Rainbow supports its customers with offices and distributors in more than 40 countries.

Total security & flexibility
Sentinel keys are available with proprietary ASIC technology, multiple EEPROM cells or even a microcontroller – giving you the world's best software protection.

Why this dongle protects more software than all others combined!

Over 6,500,000 Sentinel® keys protect software worldwide. In fact, 55% of all protected software has a Sentinel key, from Rainbow Technologies.

Today, software piracy is at an all-time high. If you're selling software without protection, you're losing sales and revenue.

Start protecting your software investment. Stop software piracy with Sentinel, then watch your sales and profits increase.

Discover the Sentinel difference
Sentinel is easy to implement, transparent to end-users, and backed by the world leader. When you need on-time delivery with local support, you need Sentinel.

Only Sentinel delivers leading-edge technology, ISO certified quality and over 99.985 % reliability.

Protect your software today
Order a complimentary 28-day *Sentinel Developer's Kit*. It comes complete with technical documentation, software drivers, utilities, and a Sentinel key.

*Call today and ask about our 20% **New Customer Discount.***

01932 570066

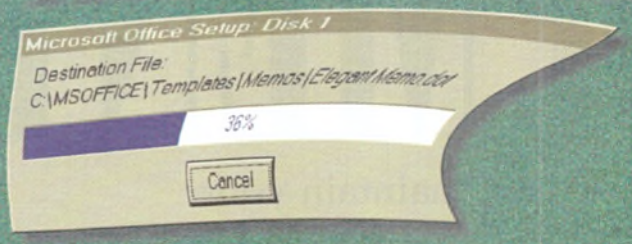
SENTINEL
Software Protection

RAINBOW TECHNOLOGIES

UNITED KINGDOM: 4 The Forum, Hanworth Lane, Chertsey, Surrey KT169JX ■ Fax: 01932 570743
USA/ASIA/LATIN AMERICA: 714/450-7300 ■ **FRANCE:** (33) 1 41 43 2900 ■ **GERMANY:** (49) 89 32 17 98 0
VISIT THE RAINBOW TECHNOLOGIES HOME PAGE AT: http://www.RNBO.COM

ARGENTINA: Agri-Aid, S.A. 54 1 8030536
AUSTRALIA: LOADPLAN 61 3 690 0455
BELGIUM/LUXEMBURG: E2S 32 92 21 11 17
BRAZIL: MIPS Sistemas Ltda. 55 11 574 8686
BULGARIA: KSIMETRO 35 9279 1478
CHILE: ChileSoft Ltda. 56 2 2327617
CHINA (Eastern): Shanghai Pudong Software Park Development Company 86 21 4371500
CHINA (Northern): CS&S 86 10 8316524
COLOMBIA: Construdata 57 1 610 7500
CZECH REPUBLIC: ASKON Int'l 42 2 3103 652
GREECE: Byte Computer S.A. 301 924 17 28
HONG KONG: Computers & Peripherals 852 2515 0018
HUNGARY: Polyware Kft 36 76 481 236
INDONESIA P.T. Promptrade InfoScan 62 21 375 166
IRAN: GAM Electronics 98 21 22 22374
ITALY: BFI IBEXSA SPA 39 23 31 00535
ITALY: Siosistemi 39 30 24 21074
JAPAN: Giken Shoji Co., Ltd. 81 52 972 6544
JORDAN: CDG Engineering 96 26 863 861
KOREA: Genesis Technologies 82 2 578 3528
LEBANON: National Group Consultants 961 1 494317
MALAYSIA: Eastern Sys. Design (M) Sdn Bhd 60 3 241 1188
MEXICO: Impex Comp., S.A. de C. V. 52 66 210 291
MIDDLE EAST: Hoche Int'l 44 81 459 8822
MOROCCO: Futur & Soft 212 2 40 03 97
NETHERLANDS: IntroCom 31 74 430 105
PHILIPPINES Mannasoft Tech. Corp. 63 2 813 4162
POLAND: HITEX Sp. z o.o. 48 22 41 97 51
PORTUGAL: COMELTA 351 1 941 65 07
SCANDINAVIA: Perico A/S 47 2249 1500
SINGAPORE: Systems Design PTE LTD 65 747 2266
SPAIN: MECCO 34 3 422 7700
SWITZERLAND: IBV AG 41 1741 2140
SWITZERLAND: Safe Compaid S.A. 41 2421 5386
TAIWAN: Evershine Tech. 886 2 8208925
THAILAND: BCS Int'l 66 2 319 4451
TUNISIA: ASCI 216 1 781 751
TURKEY: BIMEKS, Ltd. 90 216 348 3508
VENEZUELA: HRT-M Osers 58 2 261 4282

Installer for Windows 95



Dave Jewell discusses some of the issues that need to be addressed when creating a software installation program for Windows 95.

AS I'VE POINTED OUT IN the past, once you get '*under the hood*' of Windows 95, you'll often find many parts of the system that haven't changed much since Windows 3.1 - or even earlier! However, this doesn't necessarily mean that your existing Windows 3.1 installation program is going to cut the mustard as far as Windows 95 is concerned. If you run a Windows 3.1 installation under Windows 95, it will almost certainly do the job, but things won't necessarily be set up optimally from the user's perspective. Under Windows 95, users have a different expectation of exactly what an installation program should do.

For example, a Windows 3.1 program will simply open a DDE conversation with Program Manager and create a new Program Manager group, adding any necessary icons (for the program itself and any associated files such as sample data files, help files, or subsidiary utilities) to the new group. Under Windows 95, the DDE requests are fielded by the Windows 95 shell and interpreted as requests to add items to the Start menu. Unfortunately, this will have the effect of adding *all* these icons to the Start menu, as a large sub-menu. This approach is now considered bad practice since the Start menu can rapidly become unmanageable (see Figure 1). The preferred approach is to allow the user to choose exactly which icons he wishes to appear in the Start menu. Similarly, a well-behaved installation program might offer to add the application icon to the desktop.

Registry installation issues

More important than any of this, though, is the correct use of the registry. The registry is central to the correct operation of Windows 95, just as it is with Windows NT. Under Windows 3.1, everything was stored as a mish-mash of configuration information in SYSTEM.INI, WIN.INI and possibly another private .INI file which was specific to your own application. Under Windows 95, this is definitely *not* the way to do things!

This fact alone implies that a Windows 95 installation program must itself be a 32-bit application. For full access to the 32-bit registry API, you obviously need to be running in 32-bit mode yourself, unless you're prepared to perform all sorts of thunking atrocities!

Figure 2 shows the Windows 95 registry editor in operation. Here we can see the way in which application paths are set up. This is very important under Windows 95. Under Windows 3.1, the full and complete pathname of an application's executable file was stored in the Program Manager group files. Under Windows 95, this information is retrieved from the registry. Every application must register a *Default* value which specifies the path to the executable as you can see in the illustration. If you run an application from the start menu by typing it's name (e.g. RegEdit) in the Run dialog box, Windows will first look for the executable in the current path. If not found, it then looks up the full pathname of the executable in the registry and tries again. In addition, some applications will have an associated *Path* value. This is useful if your application has important data files or utilities that it expects to find by consulting the PATH environment variable. If a *Path* value is present in the registry, Windows amends the PATH variable to include the specified path before the program is executed.

While the `HKEY_LOCAL_MACHINE` registry tree is used to store information relating to the program itself, the `HKEY_CURRENT_USER` tree stores information that pertains to the current user's use of the program - i.e. user-specific preference information. This is the sort of stuff that would previously have been stored in the WIN.INI file. If you look at Figure 3, you can see part of the subtree used by the Windows 95 Explorer. Notice that because we're using the registry, we can create an arbitrarily complex subtree - the Explorer maintains a number of keys, (`DeskView`, `Printers`, `RecentDocs`, etc), each of which contains one or more data values such as `Fonts`, `NetHood` and `Programs`.

Figure 1 - Adding icons for absolutely everything to the Start menu is considered bad practice. It's better to add just the primary application icon and prompt the user for any other icons he/she wants to add.

CROSS-PLATFORM DEVELOPMENT

definition: the ability to easily maintain your source code on DOS/Windows, OS/2, Windows NT (Alpha, MIPS, Intel) and **now Windows 95**

SOFTWARE Development PRODUCTIVITY AWARD 1994

SOFTWARE Development JOLT COLA PRODUCT EXCELLENCE AWARD 1993

The award winning MKS Toolkit is the #1 choice of more than 250,000 users for powerful cross-platform development tools for the PC.

"Whether you're a programmer, administrator or power user, the MKS Toolkit turns Windows NT into a world-class operating system with a world-class set of commands and utilities. I wouldn't run without it."
—Open Computing Magazine

The power and flexibility of MKS Toolkit comes not only from its sheer number of commands, but from its ability to combine these commands to perform complex programming tasks. Here's just a sample of what you can do with MKS Toolkit:

- **tar, cpio, pax, mt, and dd enable file transfer** between operating systems and backup of your data to any ASPI compatible SCSI tape drive
- **the KornShell desktop tool** performs many time saving functions such as command aliasing, command recall and control structures
- **the vi editor** supports both a DOS command line interface and a Windows graphical user interface
- **the awk programming language** enables you to write powerful scripts and design prototypes
- **the make utility** allows you to perform precise, uniform, repeatable builds on any platform

The best toolset for Windows NT and Windows 95

Now available for Windows NT 3.5 and Windows 95, MKS Toolkit offers native 32-bit utilities such as Vi for Windows; tape utilities for data transfer allowing data exchange between UNIX and Windows 95, and 32-bit Scheduler for Windows (like a Windows cron). Other new features include *start, wcopy, wpaste, msgbox and filebox, vdiff, and viw* for Win32.

Once you try it, you'll see why MKS Toolkit – the developer's toolkit – is the **ultimate choice for cross-platform development.**

Call NOW to upgrade to the Win32 version of MKS Toolkit - from any previous version - for only £89 !
(plus shipping and VAT)

"MKS Toolkit is the most important software I own."
—Jake Ford, Software Mechanics
Wilsonville, OR

MKS TOOLKIT

MKS
MORTICE KERN SYSTEMS INC.

Tel: +1 519 884 2861
E-mail: sales@mks.com
CompuServe: 73260,1043
BBS: 519-884-2861
WWW: http://www.mks.com

To order call:
System Science
3-5 Cynthia Street, London N1 9JF
Tel: 0171 833 1022

To Order Call:
Grey Matter
Prigg Meadow, Ashburton, Devon TQ13 7DF
Tel: 01364 654 100

CIRCLE NO. 395

Code 1:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WombatWord
DisplayName=Wombat for Windows 3.7
UninstallString=C:\Progra~1\Wombat\REMOVE.EXE
```

Code 2:

```
[rename]
C:\WINDOWS\SYSTEM\NETAPI32.DLL=C:\WINDOWS\SYSTEM\NETAPI32.TMP
C:\WINDOWS\SYSTEM\VER.DLL=C:\WINDOWS\SYSTEM\VER.TMP
```

Code 3:

```
[rename]
C:\WINDOWS\SYSTEM\NETAPI32.DLL=C:\WINDOWS\SYSTEM\NETAPI32.TMP
C:\WINDOWS\SYSTEM\VER.DLL=C:\WINDOWS\SYSTEM\VER.TMP
```

Code 4:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs
        C:\Program Files\Common Files\System\vbrun300.DLL=3
```

Uninstalling your software

As you'll appreciate, Windows 95 provides a standard mechanism for installing and removing applications. To add an application, the user selects the Add/Remove Programs option from the Control Panel (Figure 4). After input from the user, this will search the designated drive for a program called SETUP.EXE. Similarly, the user can - in principle - remove any application that's been installed provided you supply the necessary information for this task to be accomplished. Again, this involves using the registry to store location and name of the uninstall program associated with a given product. If this information isn't present, then the product won't be included in the list of removable applications displayed by the Add/Remove Programs dialog. A typical entry in the registry looks like *Code 1* left:

The `DisplayName` value is the string that appears in the Control Panel's list of programs to remove, while the `UninstallString` specifies a complete path to the uninstaller. This may be followed by one or more command-line arguments.

It should go without saying that the uninstall program ought to ask for confirmation before jumping on the application directory from a great height! No confirmation is performed internally by the Control Panel itself - once the named application has been double-clicked, the uninstaller is immediately executed. As an example of how not to write an uninstaller, try removing the System Administrator documentation that comes as part of the Windows 95 Resource Kit. Yup - it's gone!

Microsoft recommend that you use the uninstall program in the *InstallShield* SDK Edition (SE) Toolkit as a starting point for your own application uninstaller. Do bear in mind, though, that NT doesn't currently support the Add/Remove Programs option, so you will have to add the uninstaller's icon to the relevant Program Manager group.

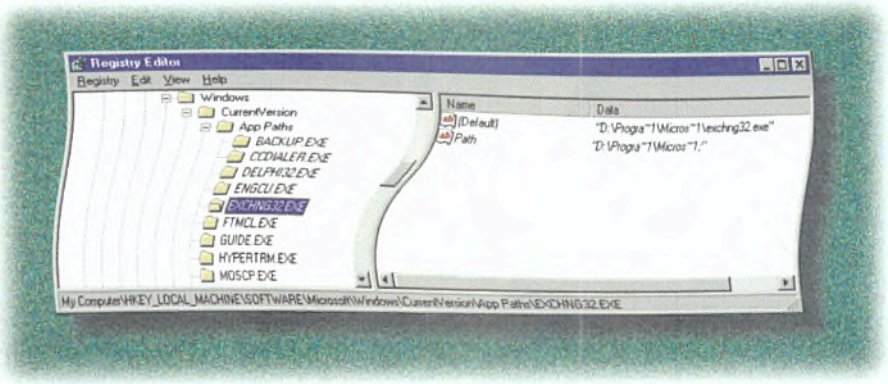


Figure 2 - All applicatins need to have a Default *key set up in the registry. This specifies the default location of the main executable. Some applications will also need to set up a* Path *value.*

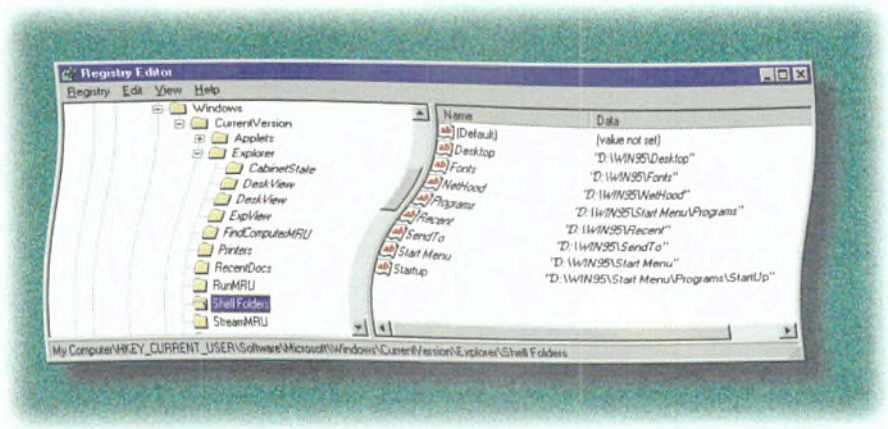


Figure 3 - The HKEY_CURRENT_USER tree is used to store user-specific configuration information. You can create your own hierarchical subtrees of information, making for a far tidier setup than you'd get with old-style .INI files.

CD-ROM based installations

A nice feature of Windows 95 is the ability to automatically run a designated program when a new CD is inserted into the CD-ROM drive. Unsurprisingly, this is called the *auto-run* facility. It's also applicable to floppy disk drives, although not all PC's have the hardware necessary to notify Windows 95 when a new floppy disk has been inserted - this is called the disk change line. You've probably already noticed that auto-run comes into operation if you load an audio CD - Windows checks to see if it's an audio CD; and if so automatically starts the CD Player program.

Using auto-run is very simple. All you have to do is include a file called AUTORUN.INF in the root directory of your CD. AUTORUN.INF is a simple text file which can contain just two lines, as below:

```
[AutoRun]
OPEN=install.exe
```

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. 396

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

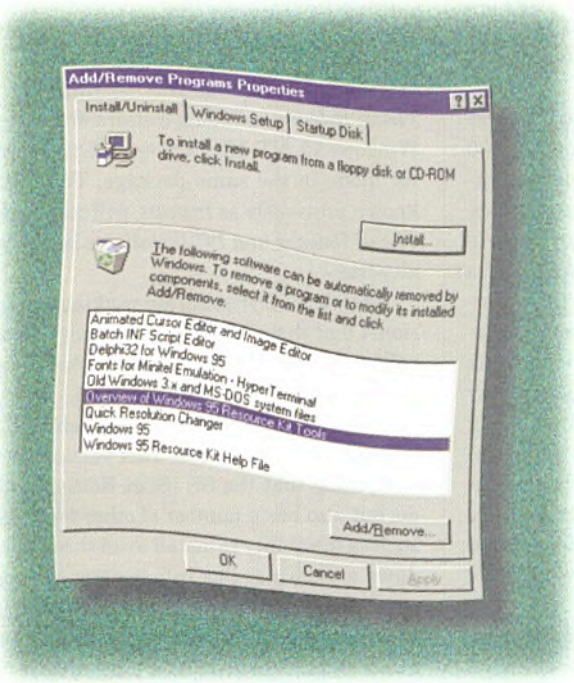


Figure 4 - Windows 95 has a built-in Add/Remove Programs option, but bear in mind that it's not available under Windows NT, so you'll need to add the icon of the 'uninstaller' program to the Program Manager group.

This will run the INSTALL.EXE program on the CD-ROM. For obvious reasons, it would be considered very bad practice for the install program to just go ahead and install the application software onto the user's hard disk without first asking! Because of the large storage capacity of a CD-ROM, it isn't always a good idea to auto-run the install program. A better choice might be to run some other application which gives the user the choice of installing the software onto a hard disk or of running it from the CD.

If you want to get really cute here, you could provide a check-box asking the user if he wants to be asked this question the next time round. By storing this information in the registry, you'll subsequently be able to auto-run (from the CD-ROM) each time the user inserts the CD. However, if you do this, you'll also need to provide an obvious mechanism for doing a full hard disk installation 'on demand' should the user ever want it. It would also be a good idea to install a short cut onto the start menu or desktop so that the user can run the program at any time, rather than just when the disk is inserted. Obviously, your user would get rather irritated having to eject the disk and then re-insert it in order to rerun the program! The shortcut should check that the required CD-ROM is present and prompt for it if it is not.

It's becoming quite popular to supply CDs which won't install a fully operational copy of an application until some sort of decryption key is entered. Typically, this key is obtained by phoning up the supplier who will be pleased to supply a key in return for intimate details of your credit card! Again, this sort of functionality could easily be catered for by supplying an auto-run application.

Do bear in mind, though, that auto-run cannot be assumed always to be available. It's possible to turn it off by holding down the shift key while inserting a CD, via an entry in SYSTEM.INI or by means of the Device Manager through the route shown in ***Code 2*** on the previous page. This being the case, you should adequately document which program the user should execute in the event of auto-run being disabled.

DLL file replacement

Installing a new application will often involve the replacement of one or more shared DLLs. For example, your application might come with a redistributable version of the common dialogs DLL. If your installer software determines that the version of the DLL currently on the hard disk is older than the version on the distribution floppies, then a replacement is called for. However, this poses something of a problem: what if the DLL we want to replace is already in use? Under Windows 3.1, if you try to replace a DLL which iss currently being used by one or more applications, then one of two things might happen: if you've got SHARE loaded, then you'll be denied access to the file in-use. If SHARE isn't loaded, then it's a simple case of 'light the blue touch-paper and retire'... This is because the next time the system tries to reload a discarded code segment from the DLL, it will end up loading garbage since the new DLL will have a different segment table layout to the previous one. Fun for all the family...

Windows 95 gets around this - though not in a particularly elegant manner. Having determined that a specific DLL needs to be replaced, the install program is responsible for updating a special configuration file called WININIT.INI. This file is processed by a small, real-mode application called WININIT.EXE next time that Windows is restarted. This application is responsible for processing the various entries in the `[rename]` section of the file. For example, a typical WININIT.INI file might contain lines as in ***Code 3*** on the previous page.

As you can see, the final name is given on the left-hand side of the equals sign, while the temporary name is given on the right. It's up to you what temporary name you assign to the new version of a DLL - just make sure it doesn't correspond to an existing file. Also, be sure to use standard '8.3' filenames rather than long filenames. This is because long filename support isn't available at the time that WININIT.EXE is executed.

There's another thorny issue relating to the sharing of DLLs. Microsoft recommend that when installing an application which makes use of a shared DLL, your install program should increment the usage count for the DLL, as stored in the registry, and shown in ***Code 4*** on the previous page.

In the example above (taken from Microsoft's documentation), the current usage count of the VBRUN300.DLL is 3. If you install a Visual Basic application that makes use of this DLL, then you're supposed to update the usage count to 4. Similarly, when the application is removed, the usage count is decremented and - if the count reaches zero - the user is given the option of removing the DLL. Personally, this approach strikes me as potentially disastrous. Even if *your* install program abides by the rules and religiously increments the usage count, what's to say that other people are going to bother? If this usage count gets "out of sync", the DLL may inadvertently be deleted and your - innocent - software might be what gets the blame. I'd steer well clear of this mechanism if I were you...

Rolling your own

From the above, you may have gathered that writing an installer for Windows 95 is a potentially more complex job than is the case with Windows 3.1 - and you'd be right. We haven't even touched on issues such as long filename support, data compression strategies and so forth. If you want your installer to work with Windows 95 and NT, then it's got to be smart enough to figure out the difference in capabilities between the two systems. If you want it to work under Windows 3.1 as well, then your best bet would be to start off by running a small 16-bit *stub* which determines whether to run a 16-bit or a 32-bit *afterburner*.

If you're going to roll your own, then at a minimum, you'll need to bone up on the registry, data compression, file copying and verifying, and shell links (which is based around an OLE implementation). If you're selling a large, complex application, then this spadework might be worthwhile, but if you're creating a small shareware, you may find that your program is dwarfed by the installer!

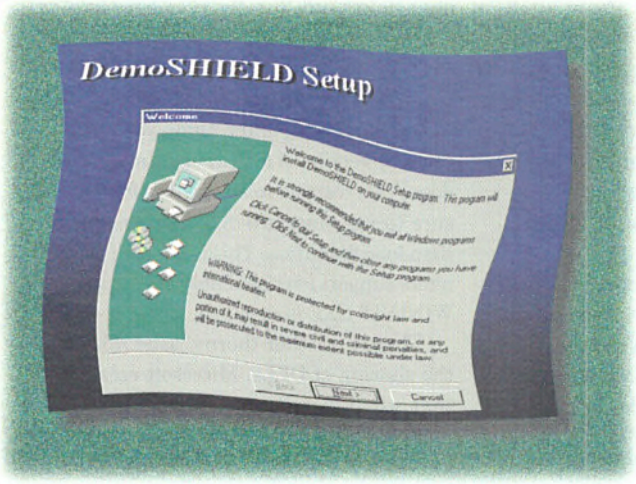


Figure 5 - InstallShield provides a consistent, Windows-95 style interface and uses a 'Wizard' to accumulate information from the user before commencing the install process.

Certainly, it makes a lot more sense to examine some of the commercial packages that now exist. At the present time, the only shipping Windows 95 installer is *InstallShield* from Sterling Technologies Inc. A special SDK Edition of *InstallShield* is distributed by Microsoft as part of the Win32 SDK, so full marks to Sterling for that particular bit of marketing strategy! As a consequence, you'll probably find that you've already got a copy of *InstallShield* if you subscribe to the MSDN Level 2 programme.

InstallShield is a large application (approx. 400 KB) and heavily script-based. It has a script compiler and can create encrypted scripts - this is very useful if you're worried about the added support overhead that results when users inevitably start tinkering with install scripts. The package includes an uninstaller program as described earlier, so you don't have to write your own.

Several other companies are working on Windows 95 compatible installers, although they're not yet available at the time of writing. These include:

PC Install from 20-20 Software. This is a simple, no-frills installer that gets the job done. It's aimed at those who do not need every possible bell and whistle that's going. Price will be £185 for a version that supports Windows 95 and Windows NT, £285 for additional support of Windows 3.1, and £345 for support of all three Windows environments plus MS-DOS.

Helpful Programs will be shipping a *Visual Release*, including install software for Windows 95, NT, MS-DOS and OS/2 character mode in the same package. This tool, known previously as *Instalit*, will cost £180. *Visual Release* and *PC Install* are both royalty-free.

Knowledge Dynamics are working on version 4.0 of their *WINSTALL* package. This will be priced at £325 for a version which supports Windows NT and Windows 95.

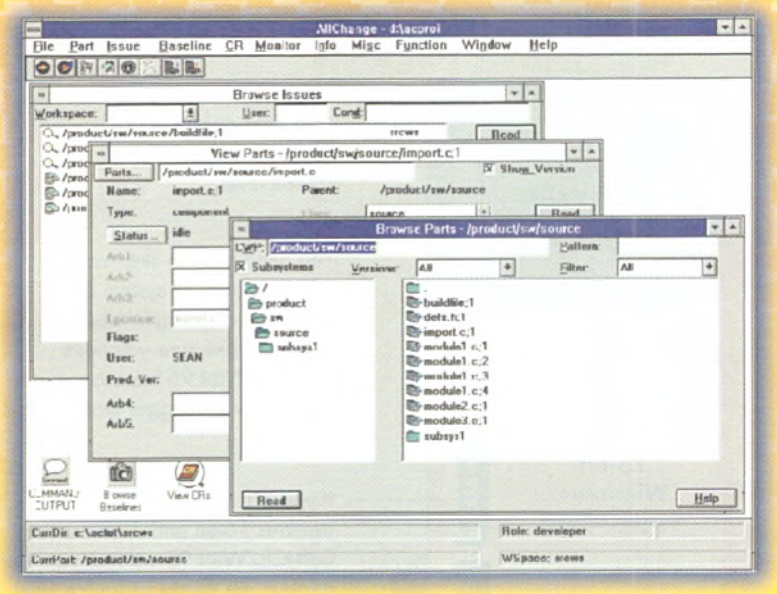
If you want the full-blown *InstallShield* 3.0, it will cost you £465. This version does everything that the SE (SDK Edition) will do, but also has a number of other features, such as the ability to install to all three Windows platforms. For information on the availability of all these installers, contact a software tools distributor such as System Science on 0171 833 1022 or Grey Matter on 01364 654100.

■

Dave Jewell is a freelance consultant / programmer and technical writer specialising in systems-level Windows and DOS work. You can contact him as djewell@cix.compulink.co.uk.

Bring it all together

with AllChange for Configuration Management



*"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

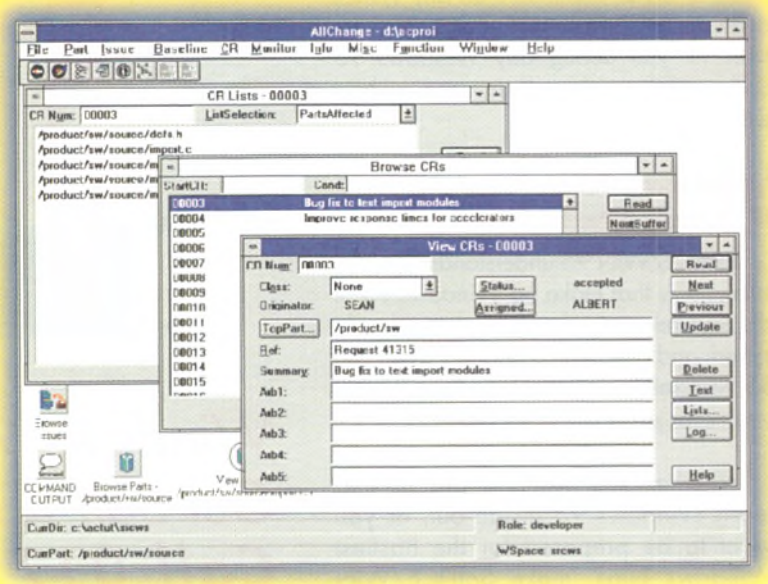
*"We chose **AllChange** because we were impressed with the flexibility of the product, reassured by the manuals and pleased with the brainy support"*
– Simon Dalley, Solartron Transducers

*"We chose **AllChange** because it had already proved itself elsewhere within Racal and it matched our requirements"*
– Dave Harmer, Racal Research

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.

- 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



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.

NEW VERSION 3

CIRCLE NO. 399

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

Nu-Mega
TECHNOLOGIES INC

Windows 95: See It All With Nu-Mega's Soft-ICE!

Your 32-bit Application Debugger Is Stuck In The Win32 Subsystem, But The Bugs Aren't!

When chasing a tough bug through the multiple layers of Windows 95, you need a debugger that can easily follow it. You need visibility and debugging power to chase bugs anywhere they go. When your conventional application debugger falls short, your only option is Soft-ICE/WIN 95

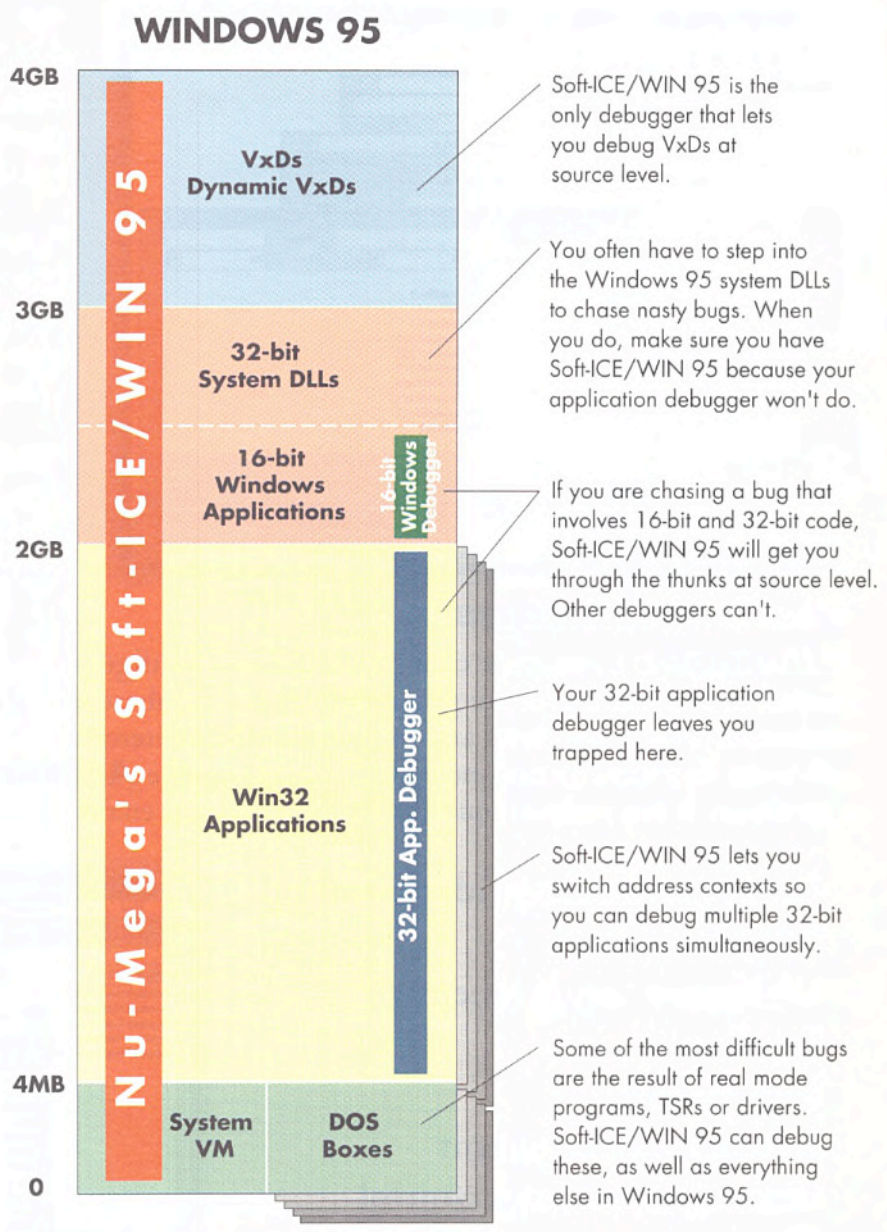
System crash bugs are particularly frustrating without the right tool. Without visibility and control, these can take days, or even weeks to solve. Soft-ICE/WIN 95 gives you visibility with it's internal system commands. It can show you what led up to the crash with its back trace history capability. Soft-ICE/WIN 95 also gives you real control because it can debug through any code in any part of Windows 95.

Soft-ICE/WIN 95 sits right on the metal. It is not dependent on any system code. Its In-Circuit-Emulator (ICE)-like features let you debug any Windows 95 code without side effects, and without expensive hardware.

Windows 95 Is Here And It Means Change.

Soft-ICE/WIN 95 understands all of the subsystems that make up Windows 95. It displays relevant information in each subsystem and gives you a bearing when you find yourself in a part of Windows 95 that you never expected to end up in.

Whether you want to dig in and learn Windows 95 inside out, or you want to be prepared for the nastiest Windows bugs, make Soft-ICE/WIN 95 a part of your tool kit.



Windows 95: See It all!
Get Soft-ICE/WIN 95
Call 0171 833 1022

ONLY £295! PLUS VAT

RISK = NULL 30 DAY MONEY BACK GUARANTEE

1-6 BRADLEYS CLOSE
WHITE LION STREET, LONDON N1 9PN

SYSTEM SCIENCE

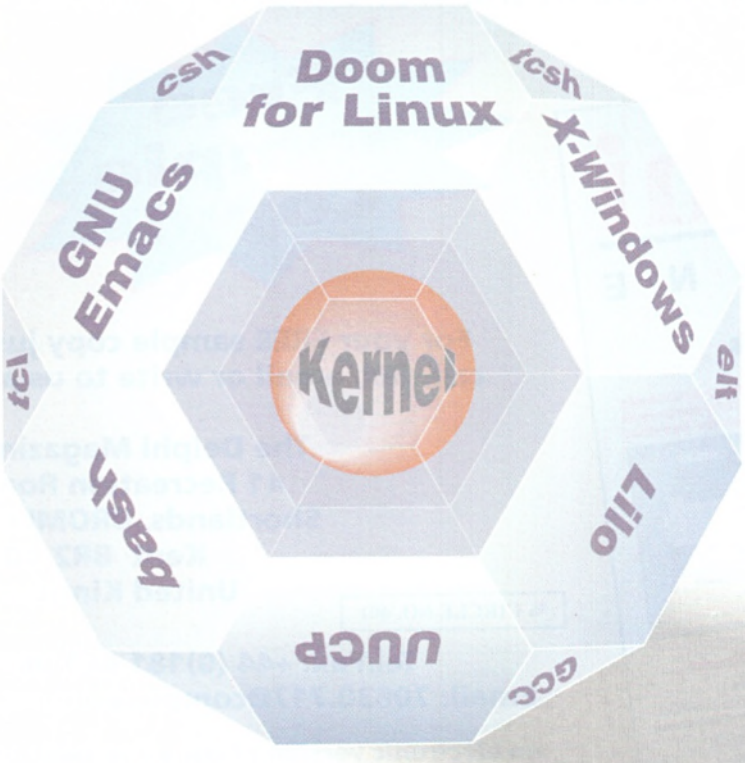
TEL: 0171 833 1022
FAX:0171 837 6411

CIRCLE NO. 400

A complete Unix operating system for free - with source code - is what you get with Linux. **Paul Dunne** gives you a taste of its requirements and possibilities.

Linux

Linux is a Unix clone, originally developed for the 386-based PC but now ported to a variety of hardware platforms. It looks like Unix, it works like Unix, but it is not like other 'Unices'. It is unique in being a free product, developed by the collaboration of a host of programmers worldwide communicating on the Internet, and available from many ftp sites. It is free, but not in the public domain - it is protected by the *GNU Public Licence*, a legal instrument that requires those who pass on a program to include the rights to use, modify, and redistribute the code. It is not the only free Unix, FreeBSD being the other well-known implementation, but it is the only one that is developed in a completely open manner: anyone can join in.



From Linus to Linux

Linux is the creation of a Finish computer science undergraduate named Linus Torvalds, who, dissatisfied with the Minix system, wrote the first version of Linux, 0.01, in 1991. Minix is a small academic Unix-clone developed by Andrew Tanenbaum for use in the courses he was teaching and in his book *Operating Systems: Design and Implementation*, published in 1987. In the early '90s, before its christening, 'Linus' Minix' was a bare kernel and not much else. As interest grew, Linux ceased to be merely Linus' personal hobby, and became more and more the work of a host of volunteer programmers worldwide, who devote time and effort gratis to its extension and improvement.

Linux was officially in Beta testing until the release of version 1.0 on 14th March, 1994. At the time of writing, two quite different versions of the kernel are available: the *1.2* series, which is the stable, production release, and the *1.3* series, which is at the cutting edge of development, a real hacker's kernel still. The rest of this article is based on version 1.2. Linus remains in charge of kernel development. Linux has grown into a modern, stable operating system used for 'real life' applications by both commercial organisations and non-profit educational institutions. It now even has its own magazine, the *Linux Journal*.

The kernel

Linus's kernel, though written to conform to POSIX specifications, and thus to be a Unix system, is free of the years of accumulated stuff that other Unices often have to carry about with them. Linus, in 1991, started from a clean slate, with nothing but the Unix system calls as a restriction.

Linux memory management is (or, in the light of the many porting efforts now in progress, was) heavily based on the capabilities of the Intel 80386 processor. Paging is supported, of

Delphi Developers:

- You read the reviews...
- You bought the product...
- You stayed up all night amazed...

Now you need the Magazine!

The Delphi Magazine is produced by Delphi developers for Delphi developers. We aim to provide the best technical material to help you develop better applications more quickly. Published 6 times per year, each issue includes a free disk with all the code and examples plus great shareware/freeware components and tools. All this for only £25 per year in the UK. We also airmail anywhere in the world – ask for prices.

Free Sample Copy!

For your FREE sample copy just call, fax, email or write to us at:

The Delphi Magazine
41 Recreation Road
Shortlands, BROMLEY
Kent BR2 0DY
United Kingdom

CIRCLE NO. 401

Tel/Fax: +44 (0)181 460 0650
Email: 70630.717@compuserve.com

An electronic version of Issue 1 is available for FREE download from the CompuServe™ DELPHI forum, as file TDM01.ZIP in Library Section 22 (in Adobe Acrobat™ format)

You can do anything with Linux that you might contemplate doing with Unix

course. Up to 256 MB of swap space can be allocated on disk, and when the system requires more physical memory than is present, it will swap out inactive pages to disk. The page size is 4 KB. The Linux kernel supports demand-paged loaded executables. That is, only those segments of a program which are actually used, and only when needed, are read into memory from disk. Also, copy-on-write pages are shared among executables, meaning that if several instances of a program are running at once, they will share pages in physical memory, reducing overall memory usage. The kernel implements a unified memory pool for user programs and disk cache: all free memory is used for caching, so the cache is reduced when running large programs.

Linux uses dynamically-linked shared libraries, meaning that programs share common library code in a single library file (similar to Sun's shared library mechanism). This allows binaries to occupy much less space on disk and in memory. Use of the shared libraries is not mandatory: there are also statically-linked libraries for those who wish to use object debugging or who need their application to run even if shared libraries are not available. Linux shared libraries are dynamically linked at run-time, allowing the programmer to replace library modules with their own routines.

To reduce further the system's footprint, Linux implements loadable modules. These are parts of the kernel which are not linked directly to it. They are loaded only when required (typically on system start-up by an rc script).

Life with others

Linux will happily co-exist on its own partition alongside one or more other OSs. Which one of them is loaded at bootup is controlled by *lilo*, the Linux boot manager. Further, Linux can actually be installed and run on a DOS partition, using the *UMSDOS* file system. UMSDOS is FAT compatible but also supports long filenames (see below). The neat little package 'mini-linux' works this way; it includes UMSDOS as its only file system.

The most common file system in use on Linux systems is the *ext2*, or *extended 2* file system. This is a modern, sophisticated file system offering support for file names up to 254 characters long, variable disk-block size, crash recovery, and files of up to 4 TB in size. Support for other file systems is extensive: ISO-9660 (CD-ROM), MS-DOS (full reading and writing), OS/2 HPFS (read-only), NTFS (read-only), Xenix, Minix and SCO.

Support for UNIX System Laboratory's ELF (Executable Loading and Linking) format for executable files is now included in the kernel, and the standard Linux C compiler, gcc, can produce ELF binaries. This gives Linux developers a much easier time when developing shared libraries and implementing dynamic loading.

Resources

So I need a big, expensive machine to run Linux on, right? Wrong! Linux will run happily on today's entry-level PCs. Most PCs will run Linux, as it includes support for MFM, RLL, IDE, and many SCSI controllers, ISA, PCI and VESA buses, many common proprietary CD-ROM interfaces, SoundBlaster cards and compatibles.... the list goes on. Work is in progress on a port to the Alpha chip architecture, to the Power PC and MIPS chips. In this article, I will concentrate on Linux as it runs on a PC, as that is the platform I am most familiar with.

Linux can run on a 386SX system with 2 MB RAM and a single high-density floppy drive. Such a configuration cannot do much, but it is a cheap and cheerful way of trying out Linux! Indeed, one of the nice things about Linux is its scalability: it likes lots and lots of everything, but it will still work with much less. Add a 40 MB hard drive to the above minimal system and you do have a working system. The laptop I use when I'm out and about has a 386SL, 4 MB RAM, and an 80 MB HD, and I manage quite nicely on it, thank you!

In practice, at least 4 MB of RAM is desirable - and as with any modern operating system, the more the better. For X Window, the minimum physical memory requirement should be upped to 8 MB. With less than 16 MB of RAM, a swap partition on the hard disk is well worthwhile. The machine I am writing this on has 16 MB of SIMMs and another 16 MB in a swap partition. How much hard disk space you will need depends completely on your requirements: typical Linux set-ups consume between 60 and 200 MB of space. According to some statistics gathered on a sample of more than 5,000 Linux users, 22% of users have a 486-33 and 32% a 486-66; 40% have between 5 and 8 MB of RAM and 37% between 9 and 16.

Practical applications

You can do anything with Linux that you might contemplate doing with Unix. That's a broad claim, and I know that there are those who are sceptical of the stability and security of an OS that's given away. Lets take a look at some of the things people are actually doing with Linux.

Linux is ideal for academic institutions, offering them a cheap way of exposing their students to the code of a real operating system. But it is by no mean limited to the academic or hobbyist worlds, it has now made inroads into 'real world' commercial environments, with hospitals, for instance, using it on a daily basis.

The use of Linux in all areas of Internet service provision has exploded over the past year. David Dennis, the maintainer of the best FAQ (Frequently Asked Questions) on being an ISP (Internet Service Provider, natch) says: 'My present system networks a 85 MHz Sun clone with a Linux PC; the Linux PC is connected to the Internet through a 28.8 Kbps SLIP connection. Although it wasn't frightfully easy to connect, everything is now working surprisingly well, with little trouble. Even after over a year of operation, the system has successfully withstood quite heavy loads... the system stayed up for 47 days without crashing.' (The latest version of his fascinating introduction to the world of an ISP can be found at the URL *http://www.amazing.com/internet/faq.html*.)

Who uses it

Sellers of Linux CD-ROMs shift 30,000 to 40,000 units per month in total according to the *Linux Journal*. Harald Alvestrand in Norway compiles statistics on the use of Linux worldwide. In his own words, as of 2nd October 1995, 'there are 20,967 regis-

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

Parallel/ Serial

PCMCIA

Mac ADB

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

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

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

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

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

DESkeys are available in a variety of low-cost high-speed models which include the standard DK12, the DK37 PCMCIA, the DK8 Mac DESkey and the immensely sophisticated and versatile DK2 and DK96.

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

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

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

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

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

Member of FAST
Promoting the legal use of software

Data Encryption Systems Limited
Silver Street House, Silver Street, Taunton, Somerset. TA1 3DL

Telephone 01823 352357
Fax 01823 352358
BBS 01823 352259
deskey@silvercityscape.co.uk

CIRCLE NO. 402

The onus is on the system administrator to make the system secure

tered Linux users. I estimate this as being between 0.2 and 5% of the total number of Linux users, giving a total community size of something between 419,340 and 10,483,500 members.' I recommend his Web page at *http://domen.uninett.no/~hta/linux/counter.html*.

Pluses and minuses

For the user of more traditional versions of Unix, Linux will naturally have caveats. *What about support?* they cry. And security?

Most adapter card manufacturers in the PC Unix market are concentrating on Linux and SCO. This is one area where the distributed nature of Linux' development is a major plus: as long as someone, somewhere, has the card, a Linux driver is probably being written. With the growing Linux community, hardware manufacturers are not waiting anymore for Linux support to be implemented by volunteers - some are already taking the plunge.

Technical assistance is readily available nowadays. The Linux newsgroups on USENET will usually provide friendly answers with useful information within hours of your posting a question. If more help is needed, then there are more and more companies springing up around the UK who will be glad to help (in exchange for a small contribution, that is).

Security is one of the primary concerns with any operating system. With Linux, the concern is emphasised - it doesn't spring from the box ready-configured. Instead, the onus is on the system administrator to make the system secure. That said, given a competent sysadmin, there is no reason to think that Linux is less secure than any other Unix. Bryant Durrell, in the Linux section of the ISP FAQ, gives some advice: 'You will have to make sure you follow the various Linux newsgroups if for no other reason than to stay up to date with that issue.'

Regarding Linux' stability, the main issue is the hardware it runs on. Like any Unix, it is much less tolerant of imperfection than dozy old DOS. Given due care in the setting-up and installation of components (there are now companies that provide ready to run Linux boxes), a production Linux kernel can be relied on to run for weeks and weeks without rebooting. According to the Linux *Info Sheet*, 'One site had a computer running version 0.97 patchlevel 1 (dating from the summer of 1992) for over 136 days without an error or crash. (It would have been longer if the JCB operator hadn't mistaken a main power transformer for a dumpster...) Others have posted uptimes in excess of a year.' Many of the software tools available with a Linux system, such as gcc, have been developed by the Free Software Foundation as part of its GNU project. (See the reviews of gcc in the February (pp 55-59) and March (pp 42-43) issues of *EXE*.) In a survey of Unix tools, the GNU set were found to be up to 40% more reliable than commercial equivalents, in terms of not crashing, not dying on bad input, and so on. Some have a tendency to distrust anything that doesn't make a substantial hole in one's budget, but even so the quality of the GNU programs surpasses many commercial products. (A full copy of this report is available at the URL *ftp://grilled.cs.wisc.edu/technical_papers/fuzz-revisited.ps.Z*).

Unix-heads have traditionally poured scorn on the limited hardware of the PC world. Historically, this attitude had much to commend it; of late, however, PC hardware has been improving: faster chips (Pentium), better motherboards (PCI), cheaper and better SCSI, bigger and cheaper disks. A Unix workstation can be built from PC components and use Linux as its OS, and compare on favourable terms with a similarly-priced Sun workstation.

Everything, including source

Linux is now a complete system with a large committed user base. In addition to getting a powerful and stable system, the possibility of getting the complete source of the full system makes it very attractive. It offers the technically-minded user access to what other systems keep hidden. Those whose job is to provide computer solutions for business are increasingly realising the advantages of an OS where everything is out in the open, since it gives them more scope for providing their customers with exactly what they want.

■

Paul Dunne runs a computer consultancy firm called Information Unbound. The company is based in Surrey and specialised in Linux and the Internet. Send email to info@tiny1.demon.co.uk or call 01483 577250.

Linux' distribution

Linux is distributed by its author only as a kernel. Others have packaged the bare kernel with oodles of supporting software. It is in this form that Linux is commonly obtained on CD or from the Net. There are a number of distributions commonly available; Slackware is reputed to be among the best. It is a collection of standard packages compiled to run on the current version of the kernel and distributed with a full installation procedure. It is also the most popular, more than 79% of the users accounted for in the Linux counter got their system as Slackware. Most CD resellers now stock, at least a Linux disk or two. Lasermoon (on 01329 834944) specialises in Linux.

Demon (*ftp.demon.co.uk*) carries a mirror of the popular Slackware distribution. The official sites in the US, *sunsite.unc.edu* and *tsx-11.mit.edu*, are mirrored at Imperial College (*ftp.ic.ac.uk*). Depending on what supporting software you want, it might be cheaper to buy a CD than spend hours, if not days, online.

The primary Linux documentation is on-line. It contains the e-books of the *Linux Documentation Project*, a plethora of 'how-to' files, and much more. This year has seen a deluge of Linux books from publishers. (See the review of *Running Linux* in the Books Reviews section in this issue). Your humble author's own effort will be appearing early next year.

BOOKS

From Chaos to Classes - Object Oriented Software development in C++ reviewed by **Mary Hope**

Up until recently books on object technology tended to focus either on the analysis and design end of the life-cycle (eg Rumbaugh, Booch) or the implementation end. Those covering implementation in C++ have moved on from tentatively mentioning classes halfway through to more modern ones that throw in a final chapter on templates and exception handling. The strength of *From Chaos to Classes* is that it bridges the chasm between OOA/D theory and C++.

The benefits of this all-embracing approach are obvious but the danger is that the text becomes of monstrous size. This book has not, at a quite readable 360 pages. This is achieved by adopting a concise bullet-point style and assuming the reader has a good grounding in C++. What other book gives you an implementation of an ADT (Abstract Data Type) using a template and mentioning exception handling by page 17! As you would guess from this, it covers a lot of ground. But it is clearly written and well set out and, as computer books go, is a 'good read'.

It starts, inevitably perhaps, with a look at OO concepts, but does this in more depth than most other books. Duffy is good on categorisation and describes nine different types of ISA links (eg subset/superset, a kind of, conceptual containment, role value restriction etc), six types of aggregation relationships (eg component-integral, material-object, portion-object etc) and a hierarchy of types of polymorphism. It is not that this is new, but that it is all the best gems gathered together with clarification through examples. What I have not seen so clearly written about before is the mapping of these concepts to C++.

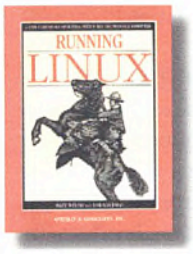
The middle section of the book looks at the process of software development using three case studies. The main method described is Rumbaugh's OMT, but Duffy supplements this heavily with other approaches. He extends Rumbaugh at the front end of the lifecycle with a comprehensive coverage of requirements determination and at the other end with more detail about design options. As well as looking at object design in C++ there is a pragmatic chapter on implementing an object model using a relational database. To complete the implementation options, Duffy explains the concepts of object oriented databases and gives a flavour of the syntax of two OODBs (ObjectStore and Poet).

Finally, Duffy gives some guidance on good practice and summarises the software development process. On reflection the range of this book is tremendous. It is not an introductory book but it is thorough and well constructed. It also has a most readable style. You have to admire someone who has obviously lived and breathed C++ for several years and yet can, in the final chapter, include a mangled James Joyce quotation 'C++ has a great future behind it'.

✔ **Verdict:**
An informative and perceptive book.

Title:	*From Chaos to Classes; Object Oriented Software Development in C++*
Author:	*Daniel Duffy*
Publisher:	*McGraw-Hill*
ISBN:	*0-07-709118-3*
Pages:	*365*
Price:	*£29.95*

Running LINUX reviewed by **Edward Kenworthy**



Despite having a horror of UNIX, the LINUX operating system has always held a fascination for me. A complete UNIX compatible - which is to say grown up and mature - operating system developed by volunteers on the Internet and available for free, or nearly so. And what UNIX tools aren't available for it, thanks to the GNU foundation and others, aren't worth talking about. What could be more exciting?

My only problem had been the aforementioned horror of UNIX (as optimised by *vi*, spit) and the lack of a machine to run it on (mine being taken up completely by a nice, user-friendly operating system - MS-DOS! Well, you can play Doom on it, can't you? *On Linux, too - Ed*) The latter problem was overcome when hard disk prices came tumbling down and I installed an obscenely large hard disk on my PC. What to do with all that space? The former was addressed by buying *Running LINUX*, published by O'Reilly. After all, any book that claimed to be able to make me into a LINUX power user must be good, mustn't it? My only concern was that I would suddenly lose my dress sense, grow a beard and sprout sandals!

Running LINUX is really a very good introduction to this UNIX clone. It begins with the expected history and general overview of LINUX, its hardware requirements and comparison with other operating systems. And then, it's straight into the most important part: installing it. I had absolutely no problems at all re-partitioning my hard disk using an 'experimental ... non-destructive' - surely an oxymoron - piece of software. Anyway it worked perfectly and despite dire warnings about everything that could go wrong with the installation, especially as I was using (presents cross and garlic bulbs) a Diamond video card! (Apparently they were a bit slow to publish the specifications necessary to write LINUX drivers.)

The book then moves on to cover such interesting topics as system administration (people really put up with that?) and the general use of LINUX and programming. It even covers how to build a new kernel, from obtaining the source to re-compiling and installing it. And, of course, 'What to do in an emergency' as the final topic of essential system management. Truly re-assuring!

My only real complaint is that it isn't critical enough of the operating system - it tends to ignore those little foibles, inconsistencies and pitfalls present in both LINUX and its inspiration. But this seems to be a common trait of most books written about any operating system. Aside from that, *Running LINUX* is an interesting and accessible introduction to the truly fascinating world of LINUX.

✔ **Verdict: Recommended**

Title:	*Running LINUX*
Authors:	*Matt Walsh and Lar Kaufman*
Publisher:	*O'Reilly & Associates, Inc.*
ISBN:	*1-56592-100-3*
Pages:	*578*
Price:	*£18.50*

Powerful. Efficient. Reliable.

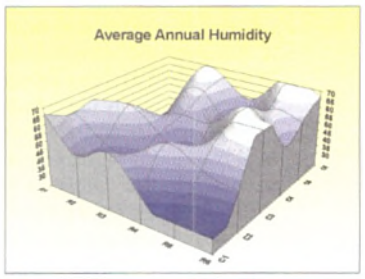
Introducing OLE Custom Controls for Windows 95 applications

Cutting edge development tools for Windows developers

Formula One™ 3.0

£195

- The best spreadsheet component just got better!
- **NEW!** Now available as a 16-bit and 32-bit OLE Custom Control.
- **NEW!** Read and write Microsoft Excel 5.0 worksheets.
- **NEW!** Support for Excel-style workbooks including the tabbed worksheet interface; include up to 256 worksheets in a workbook! Plus, use three-dimensional cell references.
- Enhanced calculation engine. The fastest spreadsheet component just got faster!
- Larger color palette.
- **NEW!** New worksheet functions including SUMIF, COUNTIF, CONCATENATE, ROUNDUP, and ROUNDDOWN.
- **NEW!** New events including Mousedown, Mouseup, Mousemove, ObjGotFocus, ObjLostFocus, and Modified.
- Enhanced printing support.
- **NEW!** Autofill worksheets with days, dates, numbers, or create your own custom fills.
- **NEW!** Draw chart objects directly on worksheets.
- **NEW!** Data validation rules.
- Royalty-free run time distribution.



First Impression® 2.0

£195

- Now, get even more charting and graphics power!
- **NEW!** Now available as a 16-bit and 32-bit OLE Custom Control.
- **NEW!** New chart types include XYZ, Contour and 3D Surface charts.
- **NEW!** New data grid allows you to view and interactively change chart data.
- **NEW!** Chart Wizard takes you step-by-step from raw data to finished chart. Plus, it includes an easy-to-use gallery of chart types.
- **NEW!** Complete object-based API.
- The only charting component that gives you a photo-realistic rendering engine. Create 3D charts in true perspective.
- Royalty-free run time distribution.

VisualSpeller™ 2.0

£129

- **NEW!** Now available as a 16-bit and 32-bit OLE Custom Control.
- **NEW!** Improved and streamlined API.
- Includes comprehensive standard dictionary with more than 100,000 entries.
- Easily create custom dictionaries with built-in dictionary maker.
- Royalty-free run time distribution.

Visual Components propels you into the next generation of Windows-based application development with the introduction of our OLE custom controls. Available as 16-bit and 32-bit products, reusable OLE components from Visual Components are the perfect tools for building state of the art, cutting edge applications in Visual Basic, Visual C/C++, Access, Visual FoxPro, and other OCX supported environments.

Look to Visual Components for robust development tools that provide unmatched functionality and the reliability you require for mission critical applications. It won't take long for you to understand why we're absolutely the leader in component software.

VISUAL COMPONENTS™

The Leader in Component Software

CALL

+44 1892 834343

LENEXA HOUSE, 11 ELDON WAY,
PADDOCK WOOD, KENT
TN12 6BE, UK

TEL: +44 1892 834343 • **FAX:** +44 1892 835843
BBS: +44 1892 835579

SUBSCRIBERS CLUB

More 32-Bit Cut Price Madness!

Now you have an even wider choice of titles to get you up to speed in Windows 95 and Windows NT. Find out what Windows 95 can do for you on the 'Net.

Windows 95 Programming Unleashed

by various.

1100 pages

Normal Price: £46.95 incl. VAT

*** EXE Subscribers Price: £35.20 ***

Every developer of retail or corporate software will have to learn the details of programming for Windows 95. Windows 95 adds several new wrinkles into the already complex businesss of Windows programming. All these are covered: MAPI, TAPI, WinSock, Plug and Play, OLE Automation and more. Authored by the SAMS programming/development group, each chapter is written by an expert on a particular topic, leading to the most complete and comprehensive reference available.

Includes bonus CD ROM

32 Bit Programming

by Ben Ezzel

800 pages

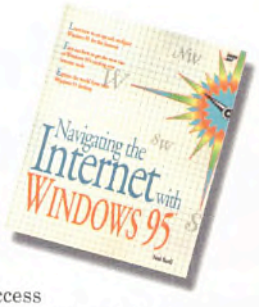
Normal Price: £37.51 incl. VAT

*** EXE Subscribers Price: £28.15 ***

The ideal book for C/C++ developers eager to port their applications to 32 bit. This title goes beyond API references and functions to emphasise design and functionality, discussing how 16 bit apps can be adapted to 32 bit. *32 Bit Programming* is a hands-on illustration of WinNT and Win95 API services, offering representative coverage of every category of function: message box dialogs, the resource editor, menus, toolbars, status bars, custom dialogs and the rest. By Ben Ezzel, author of over a dozen programming titles.

Includes bonus CD ROM

Navigating the Internet with Windows 95



By Ned Snell

400 pages

Normal Price: £22.95

*** EXE Subscribers Price: £17.20 ***

A clearly written and easy to understand introduction to using Windows 95's exciting lineup of Internet access tools. Coverage includes numerous real-life examples of what each tool can do on the Internet. The book offers various listings of the best Internet resources in each category. You'll find out how to select a service provider, configure Windows 95 for Internet access, use Windows-based Internet software tools, and navigate to various types of resources and popular destinations.

Includes bonus disk of Internet utilities.

Selection	*RRP*	*Your Price*
Windows 95 Programming Unleashed	*£46.95*	*£35.20*
32 Bit Programming	*£37.51*	*£28.15*
Navigating the Internet with Windows 95	*£22.95*	*£17.20*

Descriptions of all books below can be found in EXE Magazine, Feb to Oct.

Selection	*RRP*	*Your Price*
Windows 95 Secrets	£38.99	£29.25
Power PC Programming for Intel Programmer	£48.99	£36.75
Foundations of Visual C++ Progr for Windows 95	£38.99	£29.95
201 Principles of Software Development	£20.95	£16.80
Oracle: The Complete Reference, Third Edition	£25.95	£20.80
Teach Yourself Delphi in 21 Days	£23.00	£17.25
Delphi Unleashed	£35.50	£26.65
Visual FoxPro Developer's Guide	£41.67	£31.25
Heavy Metal Visual C++ Programming	£37.99	£28.50
Delphi Programming for Dummies	£18.99	£14.25
5 OLE Wizardry	£28.95	£23.30
C: The Complete Reference	£25.95	£20.80
The Visual C++ Handbook	£25.95	£20.80

Selection	*RRP*	*Your Price*
Guide to the Best UNIX Tips Ever	£23.95	£19.20
Object-Oriented Graphics Programming in C++	£29.95	£22.50
Video Compression for Multimedia	£29.95	£22.50
The Fuzzy Systems Handbook	£34.95	£26.25
Leaping from BASIC to C++	£27.00	£20.25
The GUI Style Guide	£29.95	£22.50
Agents Unleashed	£29.95	£22.50
Database Developer's Guide with Visual Basic 3	£41.67	£31.25
Object Oriented Programming in C++	£27.50	£20.65
Win 32 API - Desktop Reference	£46.30	£34.75
Developing PowerBuilder 4 Applications	£35.50	£26.65
Heavy Metal OLE2 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

Title	QTY	PRICE
_____	____	_____
_____	____	_____
_____	____	_____
_____	____	_____
_____	____	_____
Shipping at £3.50 per order		£3.50
TOTAL		_____

Name:_____

Address:_____

_____

_____Post Code:_____

BOOK PAGE ORDERS

My Subscription Club Number: _____

Payment Options

Cheques or purchase orders only for the moment.

❑ I enclose a cheque for _____ (payable to EXE Magazine and drawn on a UK bank)

❑ I enclose a company purchase order. Please send an invoice.

Simply fax to 0171 437 1350 (with purchase order) or post (with cheque) this form with your order to:

EXE Book Page, Centaur Communications Limited, Freepost 39 (WD 1414/29), St Giles House, 50 Poland Street, London W1E 6JZ

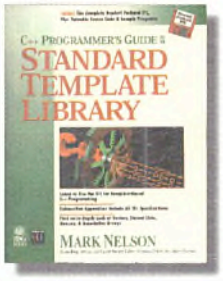
Please allow 28 days for delivery.

Crucial Competitions

* FREE BOOKS *

C++ Programmer's Guide to the Standard Template Library

By Mark Nelson



This book provides a thorough introduction to the STL even before compilers begin shipping with this new tool. Using the STL's ANSI-sanctified set of reusable C++ components you can now build template-based programs that are as sleek and as powerful as those coded by hand. And because STL adds the containers and algorithms found in pure object-oriented languages, it makes C++ more potent than ever. Watch your code performance soar now that you can concentrate your time and creativity on functionality - instead of reinventing the programming wheel!

- Inside, find the resources you need to start using the STL today. Get a comprehensive introduction to the STL just as compilers begin to feature this new library.
- Understand how generic C++ components have evolved into work-expediters rather than code bloaters, surpassing the twin tests of programming flexibility and efficiency.
- Explore STL from inside out: a comprehensive look at template container classes and algorithms - including vectors, lists, queues, stacks, and associative arrays.
- Discover how templates can turn your C++ compiler into a factory that builds new classes and functions as needed.
- Learn how to use C++ templates to create customised code that out-performs code built from scratch.
- Find expert advice, illustrations, and real-world programming examples to help you exploit the powerful tandem of C++ and the STL.
- A free disk features the complete Hewlett-Packard STL, including specifications. Save time with valuable Source Code Programs and Sample Programs. Published by IDG books (0181 231 6661)

To win a copy send a post card to the freepost address, marked "STL", to reach us by the 30th. The first card to be drawn will win.

Advanced C++ *By Namir Shammas*

The comprehensive guide to powerful C++ programming. The next step for programmers who have achieved proficiency in C++ and want to learn more.

It covers extensive programming issues, object-oriented programming and data structure tools. Text boxes and icons highlight programming notes, tips, tricks and warnings. It includes a disk containing complete source code for all examples in the book. Covers all major compilers.

784 Pages. Published by Prentice Hall (01442 882163)

To win a copy of this book send a post card to the freepost address marked 'Advanced C++'. The first card to be drawn out of the hat on 30th of this month will win.

* FREE SOFTWARE *

Totally Twisted Screen Saver

After Dark has been entertaining millions with its screen savers since its debut in 1989. Now Berkeley has taken this best-selling screen saver and gone psycho! Instead of Flying Toasters, Toilets wing their way across your screen, cows bungee jump, and chameleons do the conga across your desktop, pausing only to spontaneously combust. Featuring 13 outrageously warped and fully animated displays including an interactive Mime Hunt, Totally Twisted After Dark is hilarious entertainment for those who like to walk on the slightly weirder side of life.

Founded in 1987, Berkeley Systems, Inc. is most recognised for its widely popular, award-winning After Dark and other animated screen savers, such as The Looney Tunes, The Simpsons and Star Treck: The Next Generation Screen Savers. The company has since diversified with Triazzle Living Puzzles and the new pop-culture trivia game You don't know Jack, due out in October. To find out more about these products call Berkeley on 001 510 540 5535, or in the UK call Softline on 0181 401 1234.

To win a copy of Totally Twisted After Dark, send in a post card to the freepost address marked 'Totally Twisted'. The first card drawn on the 30th will win.

WEBSITE - *FREE copy to give away*

WebSite allows Windows NT 3.5 and Windows 95 users to begin publishing on the Web with an affordable, powerful, all-in-one package. Its intuitive graphical interface is natural for Windows and NT users. Features include:

- 32 bit HTTP server that lets you maintain a set of Web documents, control access, index your desktop directories, and use a CGI program to run other applications from within a Web document.
- WebView provides a tree-like display of the documents and links on your server; icons for file type, access state and broken links; a graphical editor for enhancing images in Web documents; wizards that automatically create common Web documents; a search button; an indexing tool that allows users to search for terms on your server; and multiple windows to view several Web sites simultaneously.
- Enhanced Mosaic 2.0 with progressive display of documents, a tool bar, security frame works and secure HTTP, support for sound and external viewers, display of in-line JPEG and other graphical files, and support for DDE.
- Complete documentation: a 325-page book that clearly describes how to use these resources, including tutorial sections, task descriptions, and chapters on using CGI to run external applications; and on-line Help that describes all the product features.

WebSite is published by O'Reilly & Associates. Telephone 0171 497 1426 for further details and stockists.

To win a copy of this prize, simply send in a postcard to the freepost address answering the question, 'What does CGI stand for?' The first correct entry to be drawn will win.

Competition Winners September Issue: MKS Internet Anywhere - S Reeve, Sutton Coalfield Essential Internet Starter Kit - Julie McCarthy, London: Top 10 CD Combo - Paul Richter, Kent: Pretty Polly - Patricia Rogers, Islington: Crossword - David Harrison, Wigan

Suzanne Chamberlain, FREEPOST 39 (WD1414/29), EXE Magazine, Centaur Communications Ltd, St Giles House, 50 Poland Street, London W1E 6JZ.

Information supplied by publishers.

ADVERTISERS INDEX

UNIVERSITY OF BATH

Video Coding Group
RESEARCH PROGRAMMERS

Two posts are available for R&D Collaborative work in still and video image coding with industrial partners in the UK, Spain and Ireland. Work will be within an active group developing applications and refining methods of leading edge technology invented at Bath University. The appointments are for 18 months with prospects for extension subject to funding by research grant.

Graphical applications experience with the Win 32 API is essential.

Salary in the range £14,317 to £21,519 per annum depending on qualifications and experience.

Closing date for applications is 14th November 1995.

Informal contact may be made with Professor DM Monro, School of Electronic and Electrical Engineering, Fax: (01225) 826073 or e-mail: d.m.monro@bath.ac.uk.

Application forms and further particulars may be obtained from the Personnel Department, University of Bath, Claverton Down, Bath BA2 7AY or phone Bath (01225) 826092 (e-mail A.J.Witcombe@bath.ac.uk) quoting Ref No. 95/227.

Working towards Equal Opportunities

Advancing Learning and Knowledge in Association with Business and Industry

MS SQL/ GUPTA SQL

Surrey **£18k-£30k**

ICARUS Systems Limited, a small but dynamic software house with an impressive client portfolio, require several SQL Base and SQL Server Developers utilising both MS and Gupta products to design, implement and maintain Relational Data Models within an MS-Windows environment. Cross training can be given in Sybase and Windows NT. Also the opportunity to mix in-house development with customer facing consultancy projects, progress to Team Leader status.

To apply, forward your CV to Robin Phillips at The Windows Connection, The Elms, 26 Broad Street, Wokingham, Berkshire RG40 1AB.
Tel: 01734 892444. Fax: 01734 893322.

Email: mail@winjobs.win-uk.net.

The Windows™ CONNECTION

WINDOWS DEVELOPERS

Visual C++ v4 & MFC v3.5
North Kent/M25 **£20,000 to £50,000**
If you are degree qualified and possess at least 2 years commercial experience of C++ (any version), then you're just what my client is looking for. Using the very latest Visual C++ & MFC tools you will be developing this company's PC based Global Investment Management applications. any other experience of Sybase, Financial systems Object Orientated Design or UNIX would also be useful.
Ref: DL/191

Visual C++, Data Capture/Image Processing
Slough **to £28,000**
Our client is the leading supplier of the most advanced Data Capture Technology today. this company is currently developing a complete solution incorporating such features as OCR, DIP, ICR and MICR. They are now seeking Software Engineers to help them maintain market leadership. You will need experience of Visual C++ in an MS-Windows environment. Any experience of Real Time systems or an interest in Data Capture or Image Processing would be an advantage although not essential. Ref: JJ/193

Creative C++ Developers
Cambridge **£17,000 - £27,000**
This company develops high profile CD ROM/Multi Media systems encompassing both Windows and Mac technologies. A minimum of one years C++ programming is required for the junior positions with an interest, or 'some' experience in Windows. For the more senior posts you will have two years of 'C'/C++ along with MFC or SDK and a good degree. Exciting moves into NT, Mac and on-line Internet systems development is already planned.
Ref: PH/194

Motor Racing Systems
Birmingham **to £25,000 + Benefits**
This well known "name" in the provision of software systems to the motor racing industry is currently looking for a Developer to work on their Windows based applications. You will need to have gained at least 18 months experience of Visual C++ & MFC coupled with either an understanding or a background of PC hardware and architectures.
Ref: DL/192

Object Oriented Senior Developer
City **to £42,000**
We require a senior developer to work on the very latest in communications technology. this company is one of the leading solution providers of LAN-based integrated communications systems, using the very latest technology in Voice Processing, Fax and Electronic Mail. We are now looking for a Senior Developer to work on new complex and innovative solutions. You will need at least 3 years solid C++ in a MS-Windows environment and have some skills in Object Oriented environments. Ref: JJ/195

Client Server Development
West London **£22,000 - £28,000**
This well established market leader develop a range of software for the travel industry. They are seeking developers to work on their new client server application written in C/C++ and 4GL's with an MSWindows client served by Oracle and Gupta. Opportunities to progress into internal project management, client liaison and the evaluation of new products and tools, is available for those seeking a springboard for their career. Ref: PH/196

Visual Basic Developers
Milton Keynes **To £24,000 + Excellent financial benefits**
If you have a minimum of 18 months experience of Visual Basic under MS-DOS or MSWindows these positions are well worth pursuing. Working on critical financially based systems for the banking fraternity, these opportunities offer a first class career move. Technical skills which include some of the following will guarantee you an interview: Visual Basic, Visual C++, MSWindows, MS-DOS MSAccess, MSMail, MSOffice.
Ref: PH/193

Visual Basic V3 moving to V4
Middlesex **£17,000 to £22,000**
This company develop a range of software and hardware products for industrial testing. they are currently developing their software systems with Visual Basic V3 but future projects will use Version 4. They are looking for a Software Engineer with one years experience of Visual Basic with any of 'C', C++, API programming, DLL's, Device Drivers or MFC who is interested in gaining new Visual Basic experience.
Ref: DL/194

OLE2
Farnborough **up to £25,000**
OLE2 Windows Developer urgently required. This vacancy would suit a graduate with a minimum of one years OLE/MFC experience who is seeking their first career move. The position is to join a very friendly team developing shrink wrapped Windows packages for mass consumption. The company offer a flat management structure coupled with salaries of up to £25,000 for only 18 months experience!!
Ref: TJ/191

VISION COMPUTER RECRUITMENT

These are a small selection of our current vacancies. Please call or send/fax a CV for more information.

VISION Computer Recruitment, 70A High Street, Stony Stratford, Milton Keynes MK11 1AH.

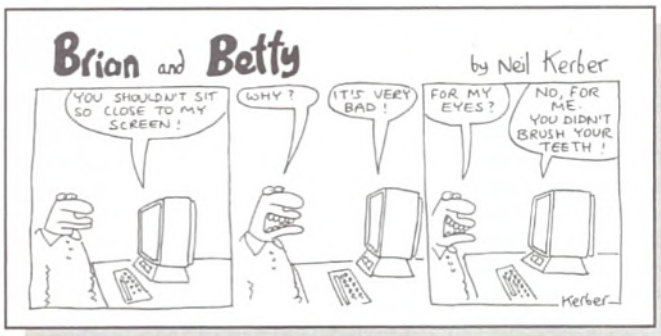
Telephone: 01908 260910 Fax: 01908 260098

VISION

Ctrl Break

Please send your rants, raves and competition entries to:

Ctrl/Break
EXE Magazine
50 Poland Street
London W1V 4AX



'Safe' the net

In the run up to World AIDS Day on 1st December netsurfers are being encouraged to 'safe' the net by tagging celebrity-designed red ribbons onto their emails and other communications.

Designs available include a red ribbon of lipstick kisses by Joanna Lumley, a red ribbon shirt from clothes designer Paul Smith and red ribbon boxing gloves from Barry McGuigan.

If you'd like to help spread the word for this highly commendable charity event a red ribbon screen saver is also available to download. Ribbons and screensavers are available from *http://www.wad.hea.org.uk*

P-P-P-Pick up a Pentium (sic)

Intel has named its next-generation processor the Pentium (R) Pro Processor. Formerly code-named P6, Ctrl-Break cannot help but wonder if the 3 missing Ps were a reference to the sometime Perplexingly Poor Performance?

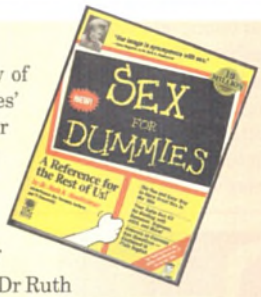
It's tough at the top

Computer Associates and the CEO Institutes held their first 'technology bootcamp' for CEOs in Switzerland last month. The aim: to bring about a clear understanding of technology's effect on global business environment. Each CEO was given:

a colour laptop,
a portable printer,
business communications software,
a suite of office productivity software
and a personal technology adviser.

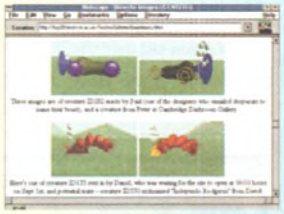
In true bootcamp style the technofest was held at Le Montreux Palace, Montreux. Still, it's a relative bargain really: the $7,000 registration fee includes the technology package, meals, special events and spouse/companion programmes. Ctrl-Break feels a need for some extra training coming on.

RTFM - with a twist



It will have been pretty hard for any of you to have missed the IDG 'Dummies' series of books for novice computer users. The publishers of which appear relentless in their aim to rid the planet of technophobia once and for all. However, it may still come as something of a surprise to hear of their latest title: *Sex For Dummies* by Dr Ruth Westheimer. It would appear that IDG are moving on from computers to pastures new. Included are tips on finding a partner, keeping romance alive, cybersex and more. All in all it looks set to be slightly more interesting than the average users' guide. 'Perhaps IDG has at last identified the true needs of the computer user,' comments the press release. Ctrl-Break is pleased to note that the committed EXE team is prepared to review the book 'in the interest of a broader editorial spectrum'. *Sex For Dummies* is available from November 1995, priced £15.99

Postcards from the edge



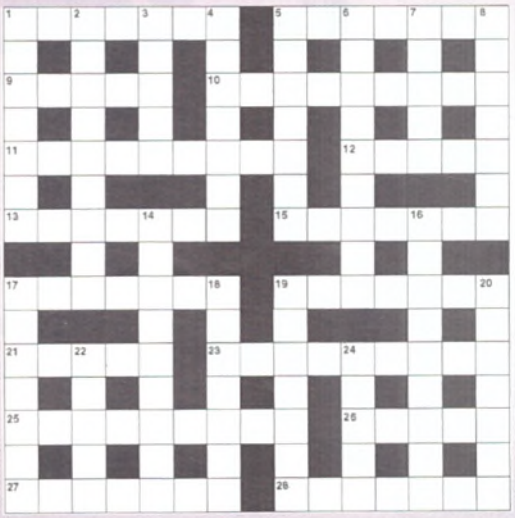
Ctrl-Break has been eagerly awaiting its email each morning with a more poignant air than usual. The reason? Ctrl-Break now has its very own pet. A multi-eyed, three-legged charmer with a squirly red body, name of Fred. Who may have gone AWOL.

As is only fitting for the new mascot of EXE, Fred is an electronic lifeform who resides in a Technosphere. Users get to design their very own 'beastie' and then receive email 'postcards' at times of importance - mating, evolution, death, that kind of thing.

But it's been a whole two weeks now... and still no sign of him. Has he forgotten how to write? Is he just having too much of a good time to bother? Could he be lying dead in a ditch somewhere?

Fred, if you're out there. Get in touch. Ctrl-Break is worried about you. Technosphere has been funded by The Arts Council of England, Film and Video Umbrella, Cambridge Darkroom Gallery, Digital Workshop Ltd. Check out *http://lcp20.lond-inst.ac.uk/Technosphere/*.

PRIZE CROSSWORD



ACROSS

1. Very small machine at the crown of the tree (7)
5. Hunting dog indicates where you are in a WIMP environment (7)
9. Labels the first two fields in a contact file (5)
10. It tells the processor to STOP NOW! (9)
11. Friendly opening interaction between two systems (9)
12. @sum gives you it (5)
13. I.n..a..s.i.l.l.y..w.a.y.. (7)
15. Someone who never works with a computer (3-4)
17. Go from first to second with a key to access new characters (7)
19. What analysts analyse (7)
21. Code much less restricted geographically than its name implies (5)
23. Boundless firm without financial safeguards (9)
25. Produced a binary version with a scanner, say (9)
26. Greeting in zero, nothing in Rome ... (5)
27. ... showing that much activity for a long time (7)
28. Jerky motor in the drive (7)

DOWN

1. Struck a blow for old media (7)
2. Injured man I steal with plywood and similar (9)
3. Posh-sounding jobs (5)
4. The first key at Key Stages 1 and 2 (7)
5. Gently slap bird as it's in a regular array (7)
6. They cause excessive stimulation, even annoyance (9)
7. Don't reject the first sign of corrosion (5)
8. Snake may be something to make you really worried (7)
14. Starts off the process with values (9)
16. Very hot place to work, and without much pay (9)
17. Nude Ida comes round without a help file ... (7)
18. ... but all tied up like 7 it seems (7)
19. / united America (7)
20. Horse trader with a miserable start (7)
22. Traps in a box of bars for a hundred periods (5)
24. Go affectedly round the coin factories, I hear (5)

SOLUTION TO OCTOBER'S CROSSWORD

ACROSS: **1.** KEYWORD **5.** ANDGATE **9.** REENTERED **10.** NOISE **11.** EXDRUID **12.** RAMPAGE **13.** STEPS **15.** VORACIOUS **17.** UNDERFLOW **19.** AISLE **21.** DRESSED **23.** ILLEGAL **25.** TESSA **26.** NOISELESS **27.** SHYSTER **28.** GUTTERS

DOWN: **1.** KERNELS **2.** YIELD **3.** OUTPUTS **4.** DAREDEVIL **5.** ADDER **6.** DYNAMIC **7.** ANIMATORS **8.** ELEVENS **14.** ENDLESSLY **16.** REWRITING **17.** UPDATES **18.** RESTART **19.** AILMENT **20.** ENLISTS **22.** DONOR **24.** GEESE

Four Yorkshiremen

The following is a verbatim account of a conversation that happens every night in the snug at the Duck & Dongle.

The names of the participants have been changed to protect their identities.

Graham Chapman: ...put some more five-and-a-quarter inch disks on the fire and let's have a bit of a blaze. These kiddywinks today, with their multi-media data sensitive CD-ROM Windows 95 for Playgroups multi gigabyte gloves and triple-soft easy-to-use Inter-bloody-active Development Environments, they don't know how to make a computer program if you gave them chapter 1 of *K&R* and typed in *Hello World* for them.

Terry Jones: Not like in our day. Four PC/ATs between the five of us, with those old EGA cards that flickered like strobe lights and a hard disk so small, if you could fit more than one project on it at a time, you were in clover. The compiler crawled, you knew you could smoke a whole fag every time you typed BASCOM, so it paid to keep the compilation errors down. The only debugger we had was a copy of SID someone had pirated, and that didn't work properly because an overlay was missing. And you know, the software we wrote was just as good as what you get now, 'cept a darn sight quicker and more reliable. *(Wipes away foam moustache with back of hand.)*

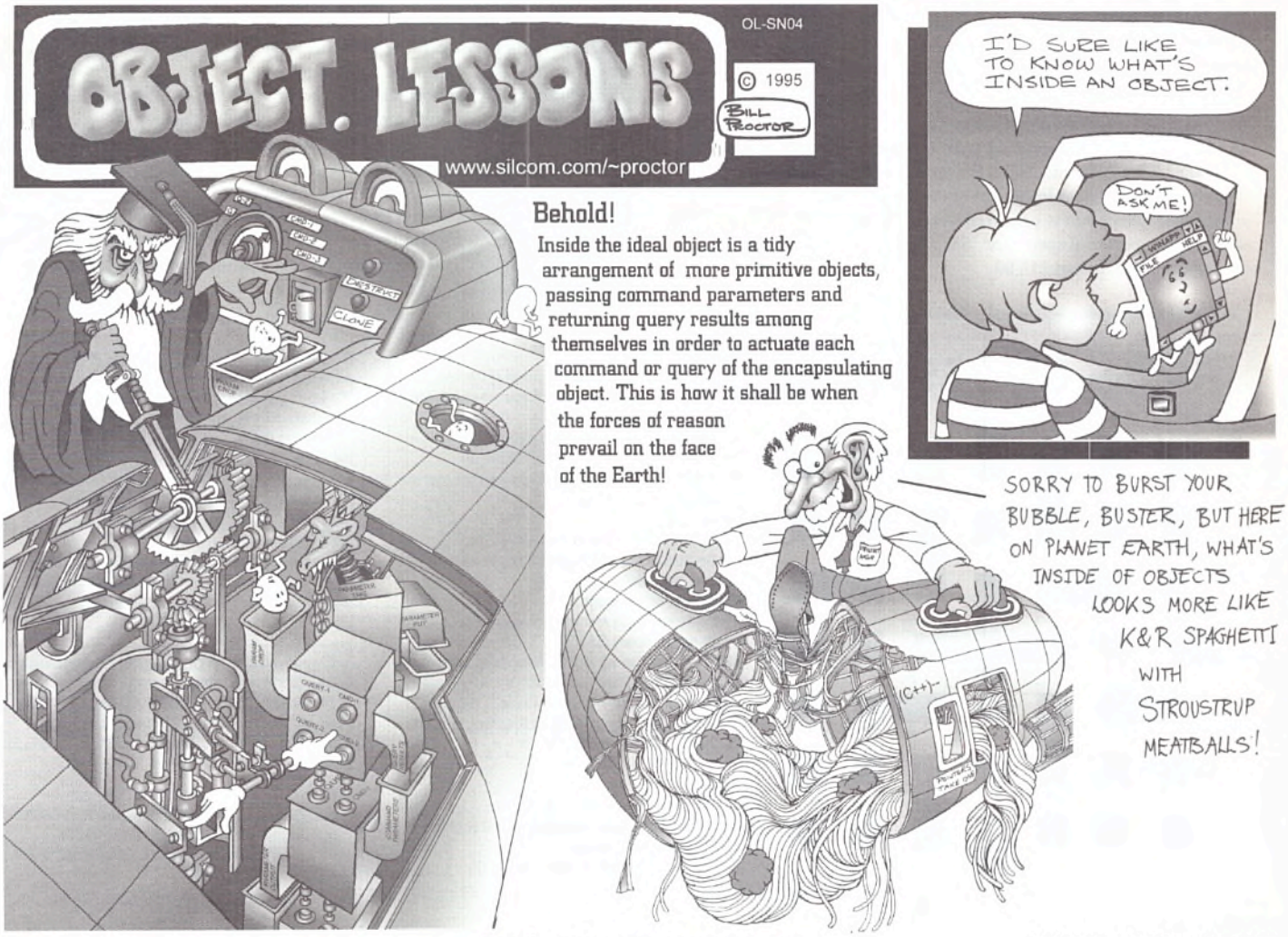
Michael Palin: Hard disk? Debugger? *(Puts on an effeminate voice)* Listen to 'im with 'is lickle hard disky-whisky and debugger-wugger. *(Normal voice.)* Now we had to do *proper* programming. Three 256 KB PC/XTs between the six of us, running MS-DOS 1.0 mark you, so no subdirectories to muck you up, not that we had room for subdirectories in 180 KB. A compilation took 20 minutes, and you had to do at least twelve disk swaps, so there was no chance of us snatching a sneaky fag in the rec room, my son. If you did, the 'Please insert disk 3 into drive A:' would burn into the screen's phosphor. Three syntax errors cost you half an hour, so we chose - and you may think this is old fashioned - we chose to bash our programs correctly first time. As for debugging, we used PRINT statements - none of this namby pamby special tool stuff. And we never had any complaints.

Eric Idle: You pair of wet girls' blouses. PCs? Compilers? *(Belches.)* We did our coding on a real machine, a Commodore Pet. Keyboard like a pub piano, screen as bright and readable as an LCD display in the midsummer sun, disk drives in a separate unit as big as a Dansette Gramophone - weighed 50 lbs and took an hour to format a disk. Not that we needed disks - we were writing in six-five-oh-two *assembler*, like God intended, and you had to load our assembler from this cassette tape. Now that *were* programming. As for debugging, we never needed any special tools - if the program crashed (and it never did) you just switched the machine off and on again - the RAM was that slow, your program would still be there, and you could work out what had gone wrong by PEEKing it from BASIC.

Graham Chapman: Well, while you boys were poncing around with your personal rubbish, some of us were working. *(Thumps table, knocking ash-tray into Palin's lap with upswing.)* I started off with half-an-hour a week on a teletype. You punched a paper tape - the punch for hole 5 was unreliable, so we used to have to literally bite in corrections - and shipped it off to the bureau, who would send you back the printout in as little as a month. The code ran on an old English Trivalve-Norton 201: 55 bit words with four 11 bit registers, and 613 words of store (and that's proper magnetic core store, mark you). As for an assembler, we did write our own assembler (in T-N 201 machine code) from a design suggested in the *Annual Review of Automatic Programming*, but I could never see the point of it. When your doing financial stuff - pounds, shillings, pence, farthings and groats - you need to be right, you need to be in contact with the machine.

Terry Jones: And if you tell the kids of today that, they just don't believe you.

All: Aye, aye, very true, whose round is it? etc. etc.



ANALYSTS & PROGRAMMERS URGENTLY REQUIRED

Maidstone, Kent **Salary - negotiable**

System C is a UK software company which since 1983 has been developing software for blue-chip clients, while also marketing the leading range of **Sycero** program generators. We are currently looking to expand our team of software developers working on an exciting new project for the UK hospital market.

Candidates should have at least two years' programming and development experience in a PC environment using one or more of the following: Sycero, C, C++, xBASE (particularly Clipper) &/or other 4GL products. Experience in the healthcare market is beneficial but not necessary, and full training in the use of System C's development tools (and Windows development in the next 12-18 months) will be given.

Ideal candidates will have an enthusiastic self-motivated attitude, an ability to communicate effectively and work best in an informal and innovative atmosphere.

Salary/benefits will be commensurate with skills and experience.

If you'd like to be a part of System C's small (but perfectly formed) team of analysts and programmers and make a key contribution towards the company's future success, please send your CV with a concise covering letter (including expected salary) to:-

Judi Holly, System C Ltd, 60-61 High Street, Maidstone, Kent, ME14 1SR. Tel: 01622 691616 Fax: 01622 691241

C++/UNIX

City **£20k-£40k**

This major International investment Banking Group are currently building what is undoubtedly the most technically advanced Front and Back Office Trading System on the planet and wish to hire several Systems Professionals from Graduate Programmers to Systems Architects.

Unlike most financial institutions where Banking/Trading applications are a recruitment pre-requisite, our Client recognises and rewards technical expertise. In addition to working with some of the best Systems and Applications Developers, extensive cross-training will be given in PowerBuilder, Sybase and Windows NT.

To apply, forward your CV to Robin Phillips at The Windows Connection, The Elms, 26 Broad Street, Wokingham, Berkshire RG40 1AB.
Tel: 01734 892444. Fax: 01734 893322.

Email: mail@winjobs.win-uk.net.

The Windows ™
CONNECTION

JOB
'C'/UNIX/TELECOMMS

LOCATION **SALARY**
Cambs £18k - £25k

Our client is searching for highly professional and committed Software Engineers with a minimum of one years experience of writing 'C' code under UNIX. Any experience of C++ and OOD would be advantageous. The ideal candidates will have experience of working in a large, well structured development environment. There are also opportunities for those Software Engineers with experience of real-time embedded software systems. Successful candidates will be working on leading edge network management systems. There is also the opportunity for limited European travel. Ref: DE/EXE/1

JOB
'C'/C++/UNIX/FINANCE

LOCATION **SALARY**
London/Surrey £20k - £28k

These positions will appeal to candidates interested in developing systems for the trading room environment. Ideally a graduate, you should have strong 'C' and/or UNIX programming skills gained in a financial environment. Additional experience with relational databases (particularly Oracle or Sybase) or Powerbuilder, whilst not essential, will be of interest. Working in a challenging development environment, there will be some client contact, so candidates should be articulate, presentable and confident.

Ref: JK/EXE/2

JOB
C++ & UNIX/WINDOWS

LOCATION **SALARY**
Bucks £20k - £30k

Seeking experienced C++ and UNIX software engineers, our client is a leading supplier of software to the oil and gas industry. Working on one of several projects applicants must be educated to degree level, preferably in a science based subject and have at least two years experience of developing C++ under UNIX. Additional requirements, though not essential, include experience of X-Windows and Motif widget development.

REF: JK/EXE/3

JOB
X MOTIF ENGINEER

LOCATION **SALARY**
Herts/London £25k - £35k

Our client a world leader in the provision of process based change control and version management systems, are searching for a key member of their team. They require a Unix engineer with a minimum of one year's X Motif experience, with excellent 'C' programming skills and a working knowledge of C++. The successful candidate will have programmed directly with Motif widgets. Any experience of SQL and Oracle, or Unix to PC connectivity would be useful. Excellent opportunities are available for the successful candidates!"

REF: DE/EXE/4

JOB
VISUAL C++/MFC - IMAGE ANALYSIS

LOCATION **SALARY**
Beds £20k - £30k

Our client, a specialist in the development of machine vision inspection systems are searching for a number of Software Engineers to develop and support these highly specialised systems. They require three Software Engineers with a minimum of one year's experience in developing Visual C++ software in a commercial environment with good knowledge of the object oriented life cycle. A good computing degree is a standard requirement. Any experience of prototyping using Visual Basic would be advantageous. Successful candidates will be encouraged to work under their own initiative, using their strong design and problem solving skills. Ref: DE/EXE/5

JOB
'C'/UNIX

LOCATION **SALARY**
Home Counties £18k - £32k

High calibre Software Engineers are required by our client, a leader in Open Systems Technology. Suitable candidates will have at least two years 'C' programming experience in a UNIX environment. Knowledge of UNIX operating system development (kernel, utilities), assembler level programming or compiler writing, although not essential, will be highly desirable. Working in a small, dynamic team, you will be involved in the development of new products and the support of existing products by investigating problems and creating solutions.

REF: LC/EXE/6

Logistix Recruitment Limited
Lamb House, Church Street
Chiswick Mall, London W4 2PD
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.

Tel: 0181-742 3060

Logistix

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, Workstation or network, your end-users will never know it's there.

SEE US AT
COMDEX/Fall '95
Booth: S-7422

VISIT OUR WEB SITE!
http://www.hasp.com/

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

1985-1995 10 YEARS

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
E-mail: aladdinuk@solo.pipex.com

North America — **Aladdin Software Security Inc.**
Tel: (800) 223 4277, (212) 564 5678
Fax: (212) 564 3377
E-mail: sales@hasp.com
WWW: http://www.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
E-mail: 100622.1522@compuserve.com

yes It runs with NetWare (Developer tested only) · Available and Ready for OS/2 · Microsoft Windows Compatible · Mac OS

■ **Aladdin Benelux** 08894 19777 ■ **Aladdin Japan** 0426 60 7191 ■ **Aladdin Russia** 095 9230588 ■ **Australia** Conlab 3 98985685 ■ **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 ■ **India** Solutions 11 2218254 ■ **Italy** Partner Data 2 26147380 ■ **Korea** Dae-A 2 848 4481 ■ **Mexico** SiSoft 5 5439770 ■ **New Zealand** Training 4 5666014 ■ **Poland** Systherm 61 480273 ■ **Portugal** Futurmatica 1 4116269 ■ **Romania** Interactiv 64 153112 ■ **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

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 controles 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 Microkerner 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 Europe
35 cours Michelet 92060 Paris la Défense 10 Cedex
Tél.: (33-1) 47 73 90 90
Fax : (33-1) 49 00 01 74

yes It runs with NetWare

Microsoft SOLUTION PROVIDER

0800 96 5497