THE SOFTWARE DEVELOPERS' MAGAZINE

EXE

MAY 1997

£3.50

Beyond C: Ada vs C++

Program for fitness

VCL and the Delphi double-act

Flash! SuperCede for Java

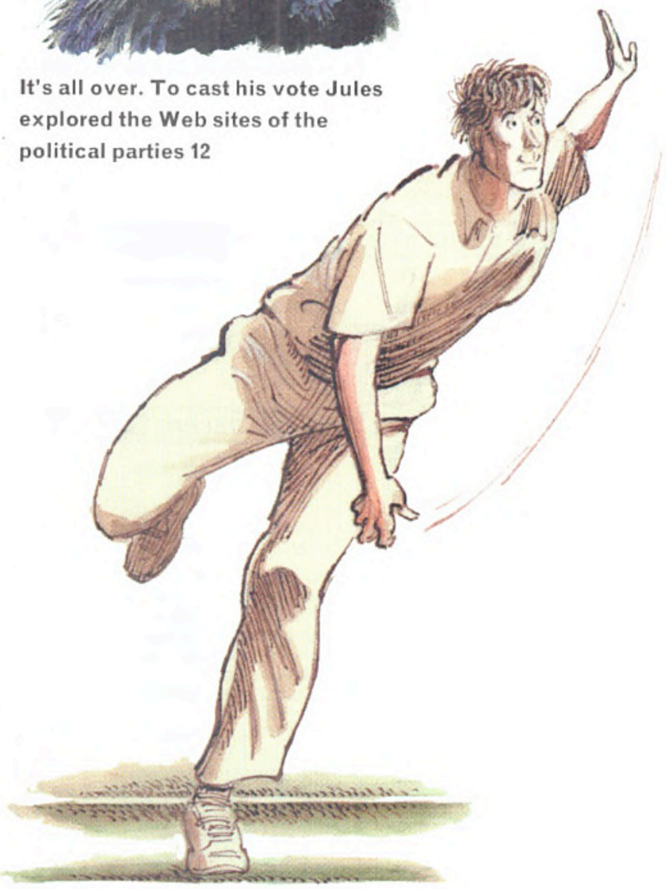
Will your PC crash in 2000?

05
9 770268 687015

STL: The rebirth of C++

It's all over. To cast his vote Jules explored the Web sites of the political parties 12

Get a real-life with Ada 25



Want to improve your cricket skills? Learn all about Hermites 53

The secret interfaces of the Delphi IDE 45

Bill Proctor's potted history of the Internet 81

Editor: David Mery **Features Editor:** Neil Hewitt **Technical Editor:** Mike Ingram **Contributing Editor:** Will Watts **Production Manager:** Kate Adams **Production Editor:** Mark English **Group Advertisement Manager:** Mark Parker **Display Sales Executive:** Ian Sinclair **Online Sales Executive:** Rob Cullen **Office Administrator:** Jacqui Ramrayka **Group Art Director:** Colin McHenry **Publisher:** Declan Gough

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 **Fax:** 0171 437 1350 **Advertising email** markp@dotexe.demon.co.uk **Subscriptions** Tel: 0171 292 3706 Fax: 0171 439 0110 email: execirc@centaur.co.uk . *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 all 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 readers. These may be edited. Information contained in *EXE* is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next issue.

Repro & Typesetting: Ebony **Printer:** St Ives (Roche) Ltd. **Front Cover Illustration:** Simon Fell **ISSN:**0268-6872

http://www.exe.co.uk

C & C++ FOR WINDOWS

Comms

COMM-DRV/LIB 16.1	£140
Fax C++ SDK for Win16/95	£935
Greenleaf CommLib 5.2	£235
OnNet SDK 4.0	£370

Compression

Crusher! Win 16-bit w/Source	£235
Greenleaf ArchiveLib 2.1	£210
PKWare Data Comp Lib Win32	£210
TCOMP/Multi-Platform 2.12	£105

Database

CodeBase 6.1	£295
CXBase Pro	£500
DBTools.h++ for ODBC	£1250
DISAM96 for Win95	£715
List & Labels for Win (Pro)	£410
POET Personal SDK 4.0	£649
ProtoGen+ Client/Server Win	£1470
Raima DBM Engine+EADS 4.0	£740
Velocis + EADS (Offer)	£250
Visual SQL	£990

Graphics - Charting

Charting Tools for Win 2.0	£180
Essential Chart for Win	£275
GraphiC/Win 7.0	£360
Graphics Server 4.0 Dev Kit	£235
Real-Time Graphics Tools	£360

Graphics - Images

Ad Oculos (Image Analysis) 2.0	£325
Image SDK Plus for NT	£610
ImageMan DLL 32 & 16-bit 5.0	£620
LEADTOOLS Win32 Pro 6.0	£715

Graphics & GUI

3d Graphics Tools 5 (32-bit C)	£230
ProtoGen+ Pro for Win32/16	£415
WinGKS	£575
WinMaker Pro 6.0	£725
zApp	£575
Zinc Engine & Win32 Key 4.2	£658

Maths & Stats

IMSL C Numerical Libraries	£495
IMSL Math Module for C++	£495
Math.h++ 5.1.3	£495
Money.h++	£1075

Sundry Components

C++ Booch Components	£575
Creative Controls Tree Control	£255
HeapAgent 16 & 32-bit Combo	£655
TG-CAD Prof 6.0	£1235
Tools.h++ 7.0	£340
WinWidgets++	£220

Tools

CC-RIDER Visual for Win16	£250
KPWin++	£625
Newi Solo Intro	£295
Visual Parse++	£289

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

BASIC LANGUAGE

PowerBASIC DLL Compiler	£106
XBasic Pro (Win32)	£540
VisualAge for Basic	£80

VISUAL BASIC 3 ADD-ONS

Comms - Async

Fax Plus for Win	£195
FaxMan SDK	£390

Comms - Network

Distinct TCP/IP Visual Internet	£265
dsSocket 1.25 Intro	£65

Database

ADE/VBX	£350
Smithware VBX for Btrieve	£180
VB/ISAM MU for Win 16-bit	£145

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	£305
ImageMan/VBX 5.0	£230

Multi-Function

Muscle (Win)	£125
VBlite 1.0	£130
VBTools 5.0	£99
Visual Developer's Suite (16 bit)	£216
WinWidgets/VBX	£150

Sundry Components

CADControl	£365
d-Barcode Dev Kit (lim runtime)	£104
VB/Magic Controls	£120

Sundry Controls

EDI-VBX 1.0	£705
Gantt/VBX	£195
VBX Artist	£240
Visual Instrument Panel Cntrls	£150

Text Editor Controls

TX Text-Control Standard	£205

Tools

TMS Tools 1.1	£99
VERSIONS/VB 1.1	£135

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.

VISUAL BASIC 4

Visual Basic Enterprise 4.0	£755
Visual Basic Professional 4.0	£365
3d Graphics Tools 5 (32bit VB4)	£149
Apiary Dev Suite for NetWare	£195
AutoCoder	£55
BetterState Pro w/VB CodeGen	£249
ButtonMaker	£75
ClassAction (VB4)	£110
ClassAssist (inc Oblets)	£175
CodeBank (VB3/4)	£88
Code Complete	£195
Crescent Internet ToolPak 2.0	£145
Designer Widgets 2.0	£99
EnQuiry 2.0 (VB4)	£94
ERwin/Desktop for VB4	£399
GeeWiz 2.0	£75
Helping Hand for VB	£150
Into Code (VB4)	£55
List & Labels for VB3/4 4.0	£295
PDQComm for VB4	£135
PowerPak Enterprise for VB4	£505
PowerPak Professional for VB4	£505
SplashWizard	£75
Spyworks - Prof 4.1 (inc Sub)	£225
Tear Off Menus for VB4	£115
VB AppFramework	£132
VB Assist 4	£130
VB Compress Pro 4.0	£98
VB Language Manager Pro 3.0	£132
VB/FailSafe Pro	£195
Visual Expert Developer	£250
XREF 2.0	£95

VISUAL BASIC 5

Visual Basic Enterprise 5.0	£995
Visual Basic Professional 5.0	£403
Visual Basic Learning 5.0	£77
Visual Studio 97 Professional	£830
Visual Studio 97 Enterprise	£1215

C & C++ FOR DOS

Comms

Essential Comm 5.0	£235
CommTools for DOS	£215
SilverComm "C" Asynch 4.06	£210

Database

c-tree Plus 6.6A	£565
SoftFocus Btree/ISAM	£75

Graphics & GUI

Fastgraph 4.0 (Ted Gruber)	£195
GX Graphics 3.0	£155
MetaWINDOW-DOS 5.0	£215
Zinc Engine & DOS Key 4.2	£658

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

General & Systems Libraries

GX Sounds	£165
MTASK	£215

Tools

C-DOC Pro 6.0	£240
C-Vision for C/C++ 4.0	£145
CodeCheck (Professional)	£475
PC-Lint for C/C++ 7.0	£135

DELPHI

Learn to Program with Delphi	£33
Delphi 2.0	£65
Delphi Desktop 2	£245
Delphi Developer 2	£390
Delphi Client/Server Suite 2	£1260
ABC for Delphi 1.0b	£99
Apiary Dev Suite for NetWare	£195
Asynch Pro 2.0 for Delphi	£135
Borland RAD Pack for Delphi	£125
Charting Tools for Win - Delphi	£180
Component Create	£162
Conversion Assistant Database	£98
DialogPROS	£210
Eschalon Power Controls 2.0	£135
Helping Hand for Delphi	£150
HyperTERP/Std	£120
ImageLib 16-bit	£109
InfoPower 2.0	£179
InnoView MultiLanguage	£125
KingCalendar Pro	£65
List & Labels for Delphi 4.0	£295
Mobius Draw Kit	£99
Mobius FastSprites	£105
OCX Expert	£195
Orpheus 2.1 (32 & 16-bit)	£135
Pumpkin Project Manager 2.0	£86
Real-Time Graphics Tools	£360
SysTools	£115
Transform: Component Expert	£125
VB2D Standard	£119
VisualPROS 1.1	£125
WinG Sprite Kit	£99

C++ COMPILERS

Borland C++ 5.0	£245
Borland C++ Dev 5.0	£337
Borland C++ Dev + Design 5.0	£585
C++Builder Standard	£69
C++Builder Professional	£397
C++Builder Client/Server	£1292
Learning to Prog w/C++Builder	£40
Optima++ Developer 1.5	£329
Optima++ Professional 1.5	£659
Optima++ Enterprise 1.5	£1295
Salford C/C++ Dev Bundle	£295
Symantec C++ 7.2	£375
Visual C++ 4.0 Learning	£83
Visual C++ 4.x Sub	£379
Visual C++ 4.2 Enterprise	£795
Visual Studio 97 Professional	£830
Visual Studio 97 Enterprise	£1215
VisualAge for C++ 3.5	£309
Watcom C/C++ 11.0	CALL

EIFFEL

Graphical Eiffel Student (Win32)	£85
Graphical Eiffel Pro (Win32)	£465
TowerEiffel for Win32 Individual	£495
EiffelCase for Win32	£310

News & Views

VISUAL STUDIO 97 PROFESSIONAL

All You Need to Build Apps for Client/Server, Web & Windows

Just as Office bundled all of Microsoft's essential apps together, so Visual Studio 97 bundles all their core development tools together:

- **Visual Basic 5.0 Professional** with native code compiler
- **Visual C++ 5.0 Professional** with enhanced compiler, VBScriptable IDE and native COM support
- **Visual J++ 1.1 Professional** for creating Java applets
- **Visual FoxPro 5.0 Professional** can use ActiveX controls
- **Visual InterDev** has all you need to develop data-driven web sites
- **MSDN CD-ROM Special Edition** contains all the technical info

Visual Studio 97 looks set to become the standard toolset for the serious developer - only £830 + VAT

VISUAL STUDIO 97 ENTERPRISE

Multi-Tier, Team Development & Enhanced Data Access

If you are involved in larger scale development, Visual Studio 97 Enterprise is what you need, adding the features in the Enterprise editions of Visual Basic & Visual C++ to provide a complete set of tools for building large, multi-tiered systems:

- **Microsoft Transaction Server 1.0, Developer Edition** makes it easy to deploy server components
- **Microsoft SQL Server 6.5, Developer Edition** for database
- **Visual SourceSafe 5.0** for teams
- **Microsoft Repository 1.0** will integrate forthcoming CASE tools
- **Visual Database Tools** lets you visualise & edit database objects
- **Transact SQL Debugger** - debug stored procs & triggers on clients
- **Remote Data Objects 2.0** for efficient access to ODBC data

How much for this box of goodies? Only £1215 + VAT. Place your order now for delivery in early April

WATCOM C/C++ 11.0

MMX, Faster Build Times & More

The latest version of Watcom's compiler is the first to support Intel's new MMX instructions and boasts a host of other improvments:

- **Latest C++ Features** including namespaces , RTTI, mutable, bool, explicit & C++ cast notation
- **MMX Support** including complete debug support - single-step through MMX instructions & view MMX registers
- **Updated MFC Support** with source, samples, and docs
- **Updated Visual Programmer** now supports the Win95 controls
- **And Lots More:** faster build times via incremental linking & improved support for pre-compiled headers; new Pentium Pro optimizations including branch prediction; multibyte and wide characters; long longs; and updated Win32 SDK support

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Fortran PowerStation Std 4	**£485**
Visual Basic Prof 5.0	**£403**
Visual Basic Enterprise 5.0	**£995**
Visual C++ 4.x Sub	**£379**
Visual Studio 97 Professional	**£830**
Visual Studio 97 Enterprise	**£1215**
C++Builder Standard	**£69**
C++Builder Professional	**£397**
Delphi 2	**£65**
Delphi Developer 2	**£390**

WITH FULL TECHNICAL SUPPORT

CIRCLE NO. 221

The future is Windows... or will it Be?

Windows is on every desktop. You should put all your eggs in one basket and develop only Windows software, right? Is that really the case? Has the whole industry effectively become so monopolistic that any other choice is a mistake?

These days it is rare for a developer to even look at other operating systems. Like a horse is fitted with blinkers to avoid being frightened and attracted by all that surrounds it, developers are pushed by commercial pressures not to even try any alternatives to Windows.

Weird situation since alternatives fill the past of anyone who has spent more than a couple of years developing software. Some started on BBC computers, some on Sinclair ZX80 or Spectrum, some on Apple II... not to forget mini and mainframes. If you had blinkers at that time you would never be using Windows today.

Even Microsoft managed to look outside the restricted view of its blinkers. Back when it was involved in OS/2 with IBM and decided to divide the world between them, it was advocating OS/2 as the only OS developers should use whatever the target OS was. The rest is history.

History is filled with now successful products which took years to enter the field of vision of most developers.

Nobody predicted the Web though Ted Nelson started his hypertext crusade back in the mid seventies.

Nobody predicted Java though bytecode interpreters, object-oriented languages and garbage collection have been in regular use for more than fifteen years.

Nobody predicted symmetric multi-processing on the desktop... the list goes on.

What is happening today that will fill this list tomorrow? I urge you to take your blinkers off and look at alternatives. Today, I decided to take off mine and buy a BeBox: a computer which can only run BeOS (and SMP Linux). BeOS has attracted a few thousand developers and no users so far, definitely not in the same league as Windows.

BeOS is a small OS, the kernel and associated tools take less than 10 MB, less than the size of a Windows application. It is SMP from the start, fully multi-threaded and comes with an integrated database. Ok, Windows is multi-threaded too but can you move a window displaying a movie without stopping anything on the desktop? And the registry is more a mess than anything else.

So why did I suddenly go for BeOS? For two reasons: first I was impressed by its speed, architecture... and really cool graphical demos. Second, it's got the same kind of elements which made the Web or Java possible: most of its technology is not new but it's the first time all of it has been integrated in a lean piece of software available on the desktop.

What is the future of BeOS? Hard to say. The fact that Microsoft has abandoned support for NT on the PowerPC might be a boon for BeOS unless of course it kills the PowerPC altogether. Be is rumoured to have started work on an Intel version but, even if there's any truth in that rumour, it won't be available this year.

What should be available to developers in May is the first architecturally frozen version of BeOS called Developer Release 9, or BeOS preview. It will be freely available to anyone in June. The cost of entry is a PowerPC machine and some are reaching the mythical $1000 limit. The BeBox itself is not produced anymore and is slightly more expensive.

Talking about collectors items, last year I won an original IBM PC Jr joystick in its original box with unopened documentation. Does anyone want to get rid of an IBM PC Jr?

David Mery

Lost script

Cast: a software developer who's been in the industry for ten years.

Set: a cubicle featuring a desk, a large screen on it, keyboard, mouse and under the desk a tower computer (brand name or type can't be read). The computer has only a 31/2" disk drive.

Props: punch cards, green and white old style perforated listing paper, a double height hard-disk, 8" floppy disks, 51/4" floppy disks (plenty of them). Lots of code in many languages calling libraries from many operating systems.

Story: the main character is about to start a new project. Then suddenly he remembers that he wrote a similar routine for another project 10 years earlier. He then starts dreaming about all the software he wrote in his life as a developer.

He first sees in an accelerated motion all computers he ever worked on – small ones, big ones, terminals, embedded systems...

...His vision starts to slow down and moves to excerpts of software he wrote: some Fortran code on punched cards. Yes, that sort routine was so neat, if only I still had those cards...

...Some Basic on perforated listing paper, [the code starts to appear in mid-air], some Ada... ... some assembly (6502, 6809, 8086, 68000...). The way he implemented this complex communication protocol in 6809, that could be really useful today...

...some PL/1, some APL, some LISP, some Smalltalk, some Z, pure lambda calculus, some C, some C++ [the precise list and order of it should be decided depending on what excerpts can be found].

If only he could access all the code he wrote in his lifetime.

Suddenly the developer wakes up from his dream still remembering it. He knows he coded the same function.

The developer tries to shove an 8" floppy disk in his 3"1/4 disk drive.

End.

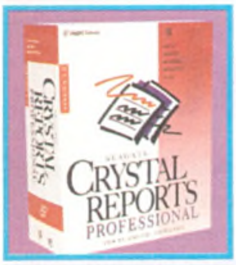
David Mery

develop to advantage

Build better applications
Add performance, functionality & versatility

CodeWright Professional 5.0

The programmer's favourite programmer's editor, now has a user interface polished for Windows 95. Favourite features, such as multi-file, multi-window editing, including search & replace, chroma-coding and more.. are enhanced with the new API Assistant and Button Links to tag notes etc. Available for Windows 3.1 and Windows 95 & NT in a single pack. Now available on CD or 3.5" — £179



Crystal Reports Professional 5.0

Now design unlimited types of reports! Create Subreports, Conditional reports, Form-style, Multiple summary cross-tab and BackOffice reports. "Plus, publish reports to the Web! (Instantly export to HTML)." New Object technology. Access PC and SQL databases. Integrate reports into database applications using ActiveX Control (OCX), VBX, VCL and MFC. FREE runtime. 16 and 32 bit versions. Upgrade £125

Powersoft Portfolio

Portfolio provides a comprehensive tool suite for designing and deploying fast workgroup applications. It includes three full-featured products: S-Designor AppModeller Desktop for application generation and physical data-modelling, PowerBuilder Desktop for creating fats Windows applications and Sybase SQL AnyWhere (4-user), the high performance workplace database. £340

MKS Toolkit ver 5.1

MKS Toolkit gives Windows NT3.5+ and Windows 95 developers a full suite of powerful UNIX tools including KornShell, awk, awkc, vi and visual diff for Windows, make, a windows scheduler, grep, sed, tar, cpio, and pax - more than 190 utilities and cammands for performing a variety of computing tasks, with support for NT & 95 long filename. For Win 95 & NT—Intel, Alpha, Mips on one CD. £259

Watcom C/C++ 11.0 – £239

The most up-to-date C/C++ compiler for cross-platform development, covering Windows NT&95, Windows 3.1, OS/2 16&32-bit and 16&32-bit DOS. Watcom C/C++ 11.0 has a new version of MFC for Windows development, incremental linking and run-time Type Identification, and the updated visual interface designer from Blue Sky. The traditional Watcom virtues of tight compilation are retained. Upgrades for Watcom 10.x user only £100.

SoftICE3

NuMega Technologies
SOFTICE3

Need to debug below the surface of Windows? SoftICE 3 offers single-machine, kernel mode, multi-thread and multi-process debugging, with Win32 fault trapping and exception handling. Version 3 has support for data structures local symbols, with full support for Windows NT and Windows 95 address contexts. SoftICE NT £525; SoftICE 95 £395; SoftICE NT & 95 £795

Data Junction Win 95/NT – £625

Data Junction excels at converting data from the widest range of data sources. Use Data Junction to convert from legacy COBOL systems, EBCDIC to ASCII, perform messy date conversions, work on packed data and populate SQL systems with data from diverse sources. Data Junctions point-and-click mapping of data source to data target saves 100's of hours of programming, and its extensive list of data transformations solves the toughest of data conversion problems. Data Junction Win 3.1 £189.

Scalable SQL ver 4.0 from Pervasive Software (Formerly Btrieve)

New Scalable SQL ver. 4.0 has ANSI SQL support, a Visual Basic Scripting Interface, and tools for creating and executing stored procedures. Scalable SQL is based on the same MicroKernal Database engine as Btrieve, and can be used to provide high level SQL organisation to a Btrieve based application. Scalable SQL ver. 4.0 is now available for Windows NT Server and Novell Netware. £call

● PLEASE CALL IF THE ITEM YOU ARE LOOKING FOR IS NOT LISTED ● CALL FOR OUR COMPREHENSIVE CATALOGUE ● PRICES ARE EXCLUSIVE OF VAT ● SHIPPING TO MAINLAND UK £8.00. 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

System Science, 1 Bradley's Close, White Lion St. London N1 9PN

your first choice for better development tools

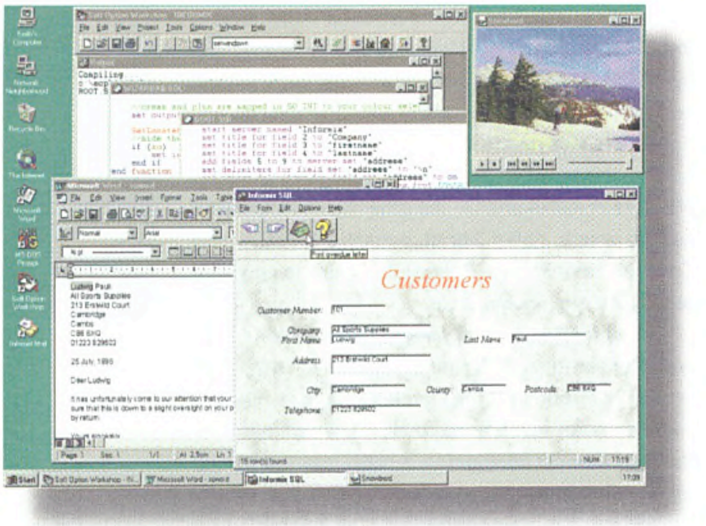
CIRCLE NO. 222

Soft Option increases strength of character

Cambridge-based Soft Option Technologies is readying version 1.4.1 of Soft Option Workshop, a tool for endowing legacy character-based applications with modern day all-singing, all-dancing, all-interoperating Windows front end user interfaces.

The package sits between the user and a terminal application, transforming the data and terminal control codes in the program's input and output streams into a fully functional Windows GUI, complete with fields, buttons, menus, dialogs and dockable toolbars. More significantly, Soft Option supports data interchange with other applications via DDE.

The mapping between the components of the text interface and the Windows functionality is done via Soft Option's own English-like scripting language. Scripts isolate fields and edit controls in the text interface based on position, drawing properties, and customisable semantic information, and 'glue' them to the appropriate Windows components.



Soft Option can interface with applications on any platform that supports terminal interfaces, and (via an optional 'DOS tube' module) with text-based DOS applications as well.

Pricing, including the accompanying three day 'Quickstart' programme covering training and consulting, starts at £8950. Soft Option: 01353 741641, fax: 01353 741341.

e info@so-tech.demon.co.uk

DevDays

Microsoft's Developer Days – a series of mass-training sessions on the Visual Studio component tools at various locations worldwide – arrived in London in March. It attracted a little over 2000 developers and managers (according to the organisers), all of whom somehow packed into the large main hall for the morning session, which consisted of taped video addresses from Bill Gates and Bob Muglia (the head of developer tools at Microsoft). The afternoon carried on with technical sessions on each of the key elements of Visual Studio.

Some might doubt the efficacy of such mass-training: because there are so many attendees, meaningful question-and-answer sessions become impossible, and the session is effectively a lecture. However, the fact that so many people wanted to attend seems to suggest that the Visual Studio tools are guaranteed a positive reception.

w URL: http://www.microsoft.com

C Y2K in hypertext; the metric system at last

As the Year 2000 deadline approaches, the problem of analysing and patching huge legacy systems on time is reaching nightmare proportions. Developers and managers are increasingly having to look to automated analysis tools for help, and it is into the waiting arms of this market that Quality Awareness' Legacy Systems Workbench (LSW) falls.

LSW comprises a set of scanning, documentation generation and code browsing tools, designed to ease the burden of updating complex legacy systems. The workbench's tools scan Cobol, CA-IDEAL and C code (at rates claimed to reach 400,000 lines an hour), generating a hypertext-searchable database of the files, dependencies, metrics, logic diagrams and other information. An analyser for CICS transaction systems is available as well.

The package includes a custom DOS-based hypertext browser, which performs functions like showing data usage in both data-used and users-of-data formats, navigating function call heirarchies, module dependencies, and browsing source files.

Other tools in the package include a date-field search utility (very reminiscent of grep), and automated generation of prologue comments describing function arguments in C code.

Conveniently, the database is stored as a combination of textual data and Access tables, so custom tools can be straightforwardly implemented. Version 2.0 of LSW is designed for 16-bit Windows platforms, but the upcoming 2.1 release will add native 32-bit versions of code. Quality Awareness: 0171 828 0300

w URL: http://www.qualityaware.com

The JavaOne conference saw the unveiling of a set of **Java Beans** components from **ProtoView** The DataTableJ, CalendarJ, TreeViewJ controls, and the WinJ component library offer more than 70 user interface effects. ProtoView: 01903 538058

Visual Basic for Applications comes to the **Web** with **EdgeworX** from Antares Alliance Group. The product's object-based design environment integrates **VBA** with the EdgeCOM library of **ActiveX** components, generating code compatible with Web servers running on Windows 95 and NT, and clients on any platform.
http://www.edgesite.com

Novell has licensed Visigenic's VisiBroker for **Java** and **C++** ORBs for integration with its **IntranetWare** platform. Future versions of IntranetWare will additionally be extended with Sun's Project Studio Java development technology.
http://www.novell.com

Version 3.0 of DataFocus' **NuTCracker** Unix to NT **porting** tool has been released. The product includes SCO Wintif technology for running ported X code with a native Windows 95 look, and support for NT 4.0 and OpenGL.
http://www.opengate.co.uk/opengate

Matra Datavision has announced release 1.4 of its CAS.CADE developer's toolkit for **2D** and **3D modelling**. The product includes both modelling object libraries, and the Software Factory tools for data modelling, prototyping and team development.
http://www.matra-datavision.fr

IE 4.0 signals war over Dynamic HTML

Coming your way in May – the new, improved **EXE OnLine**. Our Web site has been redesigned, revamped, and generally reorganised. You'll find new content, an online archive of Verity **Stob** (surely worth a Web site on its own!), and a growing selection of techniques and feature articles from previous issues. Point your browser at http://www.exe.co.uk

The latest upgrade to Object Design's **ObjectStore** object database features native support for **Java** and **ActiveX**. Enhanced versions of the **ObjectForms** and Inspector tools for point-and-click development of database-driven Web applications are shipping. http://www.odi.com

Just out from Intasoft is version 4.1 of **AllChange**, a change management package for Windows 95/NT, 3.x and **Unix**. The company claims this release to be 'Year 2000 compliant' and providing an improved GUI interface. Price from £3457. 01392 217670

RoseGUI is an **ActiveX** control for developing 'versatile and intuitive GUIs'. In reality this means constructing controls which simulate or resemble everyday objects rather than UI widgets. Available as both a 32-bit and a 16-bit OCX at $299. 00 1 415 254 0610

Version 11 of the well-respected **Watcom C++** development system from Sybase Powersoft is now available. **MMX** compilation support, RTTI and namespaces, and updated **MFC 5.0** support are among its boasted features. £240 from System Science. 0171 833 1022

The battle for browser supremacy has flared up again as arch rivals Microsoft and Netscape clash over so-called 'dynamic' HTML. Originally slated as a feature for Internet Explorer 5.0, Microsoft's implementation of DHTML was moved late last year to IE 4.0, the Developer Platform Preview release of which was posted to the company's Web site in mid-April. On the other hand, Netscape is making much noise about dynamic HTML in its Communicator Preview Release 3. Unsurprisingly, both 'standards' are incompatible.

Dynamic HTML actually began life as a World Wide Web Consortium (W3C) initiative, to which both companies have submitted their implementations. The proposal calls for extensions to HTML which allow for absolute element positioning, z-order layering of overlapping elements, and rapid, low-bandwidth interactivity. Among the demonstration files posted to the Microsoft Web site is a minute-long animated slideshow which actually consists of just 17 KB of HTML and script code, while Netscape has an interactive 3D building map constructed from HTML layers. The capabilities on offer are certainly impressive, but the mutual incompatibility is bound to cause headaches. To its credit, Netscape has pledged to support some of the Microsoft elements in future releases of Communicator, but Microsoft apparently has no plans to support Netscape-style layering in IE 4.0.

Together with 'push' technologies such as the Channel Definition Format (CDF) – based on the eXtensible Markup Language (XML) – dynamic HTML seems to have great potential for Web development. Before this potential can come to fruition, however, a single, common standard for DHTML will have to emerge.

w http://www.microsoft.com/ie/ie40/ w http://www.netscape.com

Databases without administrators

'Industrial-scale databases without a DBA' is the claim made for Scalable SQL 4. It is said to be so stable in operation that, after installation, no administrator is required to keep the databases running smoothly. As far as data migration goes, the product is compatible with Btrieve and with others via ODBC 2.0 drivers.

As well as ASCII-format SQL code, Scalable SQL is compatible with Visual Basic-style triggers. The database engine itself can be scripted with Inscribe, Pervasive's own scripting language, which is claimed to be VB compatible.

Support for NetWare and NT networks is available, although each version is separately licensed. The Developer Kit, available direct from Pervasive, includes both, together with all necessary SDKs and tools, including interfaces for Delphi and C++.

w http://www.pervasive-sw.com

Sun still controlling Java

At last month's JavaOne conference, Sun made the surprising announcement that Java is to be 'split' across multiple tracks, namely Enterprise Java, Personal Java, Embedded Java and SmartCard Java. Sun will produce different technologies and development kits for each.

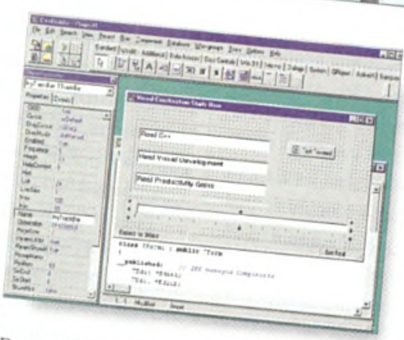
On the software front, Java Workshop 2.0 was previewed at the conference, with an 'early access' version available for download from Sun's Web site. Version 2.0's features a just-in-time compiler, support for JDK 1.1 and JavaBeans, and a plug-in facility for writing custom extensions to the development environment. Developers will be able to use Java Workshop in conjunction with another new tool, Java Studio, which Sun is planning to release for 'early access' in June. Java Studio is a JavaBeans assembly environment, with a 'join-the-dots' visual programming technique similar to IBM's VisualAge products. Developers and end-users will be able to construct Java applications from existing Beans without (have you heard this before?) having to write any actual code.

Somewhat mooting the Netscape/Microsoft IFC/AFC fight, Sun announced that its forthcoming Java Foundation Classes (JFC) will incorporate IFC, alongside IBM components, and additional Sun-sourced classes. A developer release of JFC is expected in June.

The other major initiative revealed by the company was for changes in the Java language and security model to extend the 'sandbox' concept to local filesystems. The hope is that this will allow secure access to and storage of data on users' machines without opening up opportunities for unscrupulous individuals.

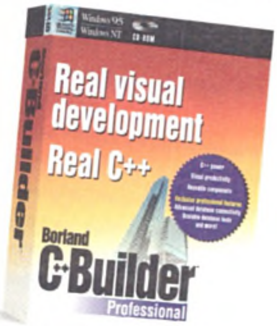
Sun Microsystems: 0161 9620000.
w URL: http://www.sun.com

The search for a truly visual, Rapid Application Development tool for C++ stops here with the introduction of Borland C++Builder. Only Borland C++Builder offers developers a real visual environment to construct applications with the power of industry standard C++.



Borland C++Builder includes over 100 reusable components with source, scalable database tools and the option to seamlessly interchange between visual development and straight code editing.

Whether you want to visually construct front ends for applications or develop enterprise-wide, scalable solutions, there is a version of Borland C++Builder for you.



To place your order contact one of our authorised dealers quoting reference BC9741EX on:

Action Computer Supplies	Call 0800 333333
Grey Matter	Call 01364 654100
QBS	Call 0181 9568000
Simply Computers Ltd	Call 0181 4982130
Software Warehouse	Call 01675 466467
System Science	Call 0171 8331022
Technomatic	Call 0990 134710
Watford Electronics	Call 01582 745555

For information on all Borland products call customer services free on 0800 454065

Only Borland C++Builder Combines Real Visual Development with the Power of Real C++

CIRCLE NO. 271

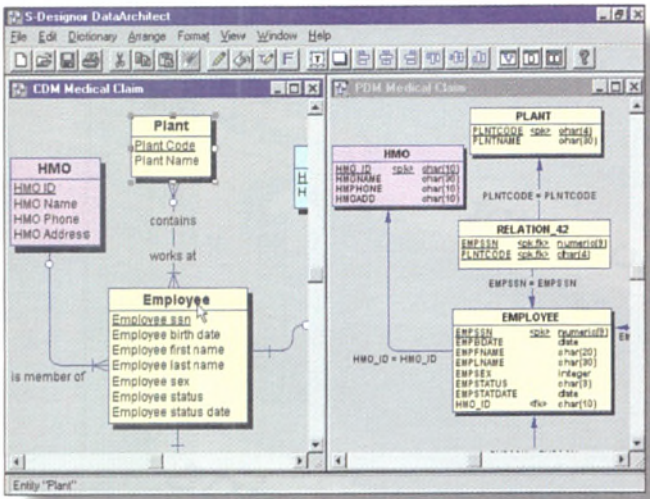
Borland

http://www.borland.com/uk

Borland International (UK) Limited, 8 Pavilions, Ruscombe Business Park, Twyford, Berkshire RG10 9NN.

BC9751EX

PowerDesigner 6.0 goes back and forth



Powersoft recently announced the renamed PowerDesigner 6.0, the latest upgrade of its S-Designor system modelling toolset. Version 6.0 has quite a few new features, including WarehouseArchitect, a new module for designing and building data warehouses.

WarehouseArchitect maintains a map between source information and the data warehouse, assisting the data extraction, cleansing and end-user query process. The module can reverse engineer and generate schemas from existing databses, and supports dimensional modelling (including star and snowflake schemas), aggregation, partitioning, and dimensional hierarchies.

AppModeler has added code generators for Powersoft's own Power++ and Delphi 2.0, and there is a Web module which can generate data-driven Web applications directly from a data model. ProcessAnalyst supports CRUD matrices for identifying the impact of processes on data.

ProcessAnalyst can work with business rules, and rules defined in ProcessAnalyst can be imported into the conceptual design phase in DataArchitect.

PowerDesigner 6.0 is expected to be shipping by mid-June, with pricing to be announced.

w URL: http://www.powersoft.com

Universal communication

Langner Software is to release in the UK a communication module which supports many protocols from a single API. Universal Communications API is claimed to provide a unique interface to all types of transmission media and communication protocols so that you can develop one application and let the user decide how he wants to communicate whether over ISDN or via TCP for instance. The module is available for C, C++, Delphi and Visual Basic. The first version supports TCP/IP (with FTP, SMTP, HTTP, Telnet, TCP and UDP), asynchronous and synchronous serial communication (including ZModem, fax, ISDN, X.75, DTMF recognition, HDLC, BSC, SDLC, and frame relay).

The price has not yet been fixed but should be around £800. Distribution of the library is royalty free. For more information contact Silicon River on 0181 317 7777, sales@siliconriver.co.uk

Visual computing

Intel introduced its new vision for the PC platform last month. According to Craig Barret, Executive VP and COO, the PC has evolved from a 'Productivity PC' to a 'Multimedia PC' to a 'Connected PC'. The next step is the 'Visual PC', the three basic stones of visual computing being 3D graphics, digital imaging and video. Of course, Intel announced some new components for building this Visual PC, including the Pentium II with MMX technology and the Advanced Graphic Port (AGP 1x has a 266 MB/s bandwidth, AGP 2x goes to 533 MB/s and AGP 4x will reach 1 GB/s – compared to 133 MB/s for PCI).

To tout the greatness of this new vision, the presentations (which made use of *very* aliased fonts) included some demonstrations which were supposed to be similar to movies and real-life visual experience. If these were representative of what's to come then you should start to get used to seeing life with polygons.

Moshe Dunie, VP Windows OS at Microsoft, promised to ship AGP support in Memphis and NT 5.0. DirectX 5 is planned for June. Microsoft is working with Intel on Direct3D, OpenGL, Talisman, AGP and MMX – the former delivering the software and the latter the hardware.

The most impressive demonstrations were done by MetaTools, the first consisting of scanning a face with a laser and building a colour polygon representation – in real time. The second one was of Goo, Metashow and Soap, all graphical programs which are easy and fun to use (being targeted at children, high-level executives and developers). MetaTools co-founder Kay Krause explained that software interfaces need serious rethinking (and simplification): 'we're not CPU-bound, we're idea-bound'.

w URL: http://www.intel.com

Select Enterprise 5.1 is expected to be available by the time you read this. The new version of Select's component modelling tool includes support for **UML**, ERwin, **CORBA** IDL generation and improved **C++** and reverse-engineering support. http://www.selectst.com

Microsoft enters the model-based development arena with the launch of **Visual Modeler**. Developed jointly with Rational Software, Modeler is designed to integrate with the **Visual Studio 97** range of tools, and will be available for **download** by registered owners of Visual Basic 5.0 or Visual Studio 97.

New from Nat Systems is version 2.1 of **NatStar**, the client/server application development environment. Additional support for **mainframe** databases has been added including **Bull** machines as well as for the Object Modelling Technique (**OMT**). 0181 232 9810

Pure Atria has produced new versions of its **ClearDDTS**, **ClearCase** and **ClearGuide** change-request and software configuration management tools, promising tighter **integration** between tools. http://www.pureatria.com

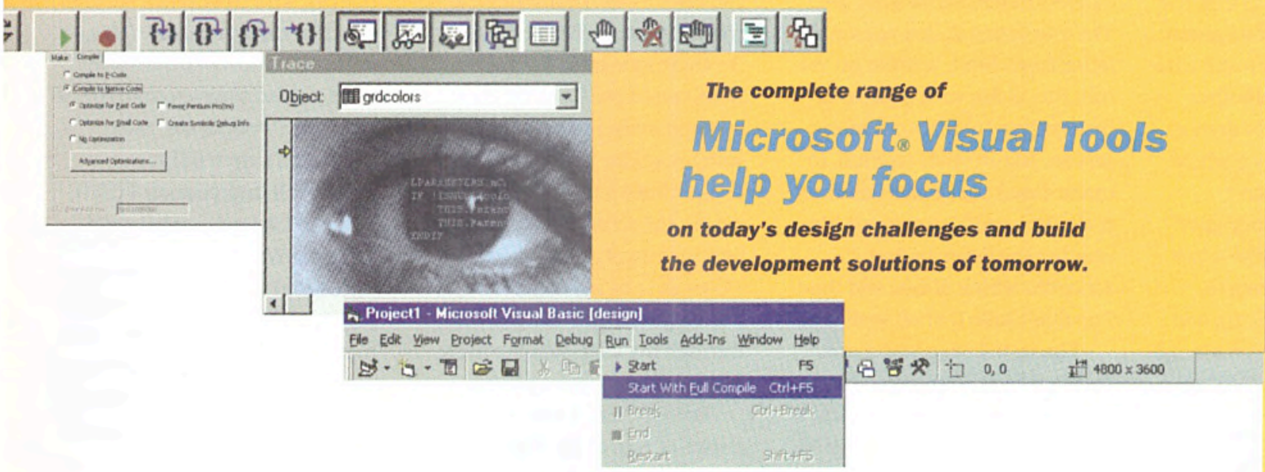
The new releases come hot on the heels of **Rational Software's** announcement that it is to acquire **Pure Atria**, with the intention of integrating the two companies' products. Details are available on Rational's Web site at http://www.rational.com

Ron Peck, Intel's NetPC Marketing Manager admits that the '**NetPC is not secure**'. Security was not an objective of the first NetPC reference specifications.

Microsoft®

Visual Studio 97

has all the tools you need to build the next generation of killer apps.



The complete range of

Microsoft® Visual Tools help you focus

on today's design challenges and build the development solutions of tomorrow.

Microsoft Visual Studio 97 *(Professional Edition)* includes:

Each of these products are also available separately.

Microsoft® Visual Studio™ 97 is a comprehensive suite of Visual Tools that will help you integrate client/server solutions with the latest Internet technologies. Microsoft® Visual Studio supports industry standards for components, data access, development languages, and the Web.

For large-scale development, Microsoft® Visual Studio 97 Enterprise Edition also features Visual Database Tools, Microsoft® Visual SourceSafe™ and the Developer Editions of Microsoft® SQL Server™ and Microsoft® Transaction Server. And for fast, easy reference you can take advantage of the Microsoft® Developer Network Library Special Edition. Best of all, when you get Microsoft® Visual Studio 97, you'll have an entire suite of Visual Tools for approximately the price of two. To discover more about what people are already calling the "developer deal of the year" be certain to visit us on the Microsoft UK Developers web site.

www.microsoft.com/uk/msdn/

Microsoft®

where do you want to go today?™

For more details contact your local reseller or call:

GreyMatter 01364 654100 PTS Direct 01928 579900 QBS Software 0181 956 8000 System Science 0171 833 1022

CIRCLE NO. 223

Mayhem

Jules looks at the one issue that really matters.

By the time you read this, the general election will probably be over. But, as I'm writing this, the election has just been announced, and the only real prediction I can make is that the pollsters will get it wrong again.

I can't make up my mind who I want to win. I think another Tory win will be bad for democracy, because even now a whole generation of voters can't remember what a Labour government was like. On the other hand, New Labour seems to have been dragged kicking and screaming into a position almost identical to the Tories (did you know that 'Tony Blair MP' is an anagram of 'I'm Tory plan B'?). This leaves the LibDems, who were originally the centre party, looking like the old Loony Left.

I'm supposed to be fond of computers, so I thought I'd check out the Web sites belonging to the two and a half main parties. The LibDems are at, not surprisingly, *www.libdems.org.uk*. The site is lightly sprinkled with monochrome thumbnails of Paddy being listened to intently by faceless audiences (I guess he's competing with Major in the grey stakes), but apart from that isn't highly designed. Most of the site is made up of policy statements, press releases, and links to reference information. There's a huge amount of information there, all of it cross-linked and referenced. Only one problem: while the lead stories in the *news* were about Paddy criticising the other two for continuous insults, the site still has nothing good to say about either of them.

The Tories are at *www.conservative-party.org.uk*. I suppose only the Tories would consider the need to assert the fact that they are, in fact, a party in their address. On the other hand, rumour has it that they are in reality two parties, so I guess the address makes sense after all. The site opens with a personal letter from John, written on vellum tile, along with a photo of him staring intently out of the screen, Big-Brother fashion, to make damn sure I was listening. It worked, until I imagined him reading the letter in that soporific whine of his, and then I couldn't take it at all seriously.

The index page was a confusing mix of bitmapped buttons. No manifesto, no policy; a single button for achievements leading to a list of achievement sectors, and each of those leading to a single page of about 50 words ('We've increased home ownership. We've got an idea to get teenagers off the streets – might work. Return to home' Very funny!) One of those achievements is called 'the Labour threat' or some such: presumably the Tories are taking the credit for Labour reinventing themselves.

Oh, one other button led to a funny page – if you like Andrew Rawnsley and Vincent Hanna, then you'll probably find this sidesplitting. It was the only (intentional) humour on any of the sites, so I guess they deserve credit for trying.

What was noticeably absent from the Tory site was any acknowledgement of the cabinet. No Michael Hesletine, no Michael Howard and no Kenneth Clarke (who is probably the most bloke-like in a PR sort of way among the lot of them). This was in complete contrast to the Labour page, which positively assaulted me with face after duotoned and anamorphed face right through the session.

Now, the labour page on *www.labour.org.uk* is something you have to see. Dominating a credit screen even more Big Brotheresque than the Tories, Blair stared out from the shadows and made me a *pledge*. After watching it for just long enough to guarantee a week of nightmares, the screen was replaced by a picture of John Prescott, who was to be my constant friend, guide, and companion through my visit. Reassuringly, John also made me a *pledge*.

Trouble was, I was taking no notice of him at all. The screen was alive with waving flags, twinkling election logos, successions of flashing faces, and frames dancing around. Only trouble was, the page had no links on it – I had nowhere to go.

Eventually, I found it: a single solitary link at the bottom of the page, tucked away where only Hercule Poirot could find it. I clicked. John came back, more faces came back, I got even more frames and now I had *five pledges*, each numbered, and none more than about twelve words long. And one link.

Now, finally, it offered me some content. It listed four different manifestos to look at (one for each of England, Scotland, Wales, and Wales in Welsh). So I guess that Labour have at long last made their intentions clear about the Union. I selected the English manifesto, and guess what I got – those five bloody *pledges*, each with a link to tell me (in *bigger letters*) what the *pledges* meant. I had no idea if they were any use, because the assertions directly contradicted the Tories assertions, and neither had quoted or linked to any supporting reference at all.

But, content is not the point of this site: it looks like Radio 1 in drag, and feels like a second-rate learning program. It's not designed to answer questions. It's a guided tour around Top of the Pops ('Follow me. Now cheer and clap wildly!')

Overall impressions: I think only the LibDems understand what the Web is about. All three have hired Web designers to make their pages, but the two main parties seem to think that the profile of Web users is the same as Sun readers, so either they've been persuaded by pretty pictures, or they're deliberately trying to avoid saying anything definite at all. The other thing is that, while the two main parties have built their pages around soundbites, they're not the same ones that they use on the telly. Remember when each was saying 'We're the party of law and order!' 'No you're not, we are!'? Perhaps they've realised that people prefer justice and freedom, or perhaps they've realised that neither party has any control over crime at all, but whatever the reason, law and order is totally absent from the sites. Same goes for Europe, civil rights, financial policy, Ireland, and morals.

So, after having looked at the pages, I think they have helped me to make up my mind. I hope none of them wins. I hope we get a hung parliament, and the subsequent wheeling and dealing will prevent the formation of a government for at least a year. It will save the taxpayer loads of money, and we'll probably all be able to get on with our lives without anyone noticing the loss. ■

Jules is not standing for any office at all. Vote for Jules!

We welcome short letters on any subject 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

Start up – I disagree

Dear Sir,
I bought *Start up* (reviewed in the March edition) on a trip to the USA. When I got home and read it, I found it a fascinating insight into how the venture capitalist world operates, and how the big companies like IBM, Microsoft and Apple play with the little companies that threaten them. My favourite part of the book was where he states that they knew running the company would either be 'a hell of a joyride' or a 'joyride to hell'. I liked the book so much I couldn't stop reading it – I stayed up until 4am on a Sunday night to finish it so I could get back to normal work. But then maybe the fact I was six months into starting my own software company made it more relevant to me?

I recommend it to anyone interested in how hi-tech companies operate.
Matthew Jones
mattjones@cix.compulink.co.uk

Hmmm. I don't think I have any big arguments with what you say. My dislike of the book was caused by factors fairly independent of the ones that you like.

I confess I was very irritated by Kaplan's attitude. As it seemed to me, the whole Go! fiasco was Kaplan's fault – for example choosing IBM not H-P as a partner, and then being surprised when he got bogged down in time-wasting political manoeuvres and lawyer nonsense, when (as he more or less says) this was completely predictable. Indeed, the emphasis he puts on certain incidents (eg birth of the pen computer idea on Kapor's jet), you would think that Go! had succeeded and that he had become a billionaire. Or even that someone, anybody, had made a pen computer that worked satisfactorily.

But I concede it is clearly written. And I did recommend it for the humor (sic).
Will Watts

Trad vs RAD

Dear Sir,
I read this article with great interest. I currently work for a large company which historically has used SSADM as a method of development. Due to a considerable change in the tools we are using to develop applications, SSADM would seem to be most inappropriate and we have certainly suffered a lot of time overhead using this method.

Recently we have been looking at the 'fusion method' and the 'unified method' which are much better suited to object oriented development. They also very much promote the idea of an iterative process of development.

My only concern with a RAD approach to design and development is the inability to see the wider picture and perhaps being focused to early. This can also mean that maintaining the finished product can turn out a bit of a nightmare.

I was just wondering whether you may have any suggestions which make it possible to effectively design the overall system including a sound foundation of common use functions, but at the same time being able to adopt the prototype approach.

Another problem which has been raised in my company is, if a unit or system has been developed using the RAD approach does it not mean that less or no documentation is available – should this be produced retrospectively? In particular, does one produce a detailed program specification before prototyping or is this left until after the prototype phase?

I personally believe a RAD approach is the correct way to develop a lot of systems, however I think whatever the method chosen if it is not welcomed and understood by the company as a whole ie by both the IT department and potential users, then the method will fail. With this in mind it is necessary to be aware of the change the company may need to make to support any given method and in addition the size of the training problem should be fully appreciated.
Tony Hewitt
TonyHewitt@compuserve.com

Tony, I think your concerns with a RAD approach are all valid. I do not believe that there is (yet) a methodology that can combine the speed of RAD techniques with the integrity of an SSADM development. If anyone can produce one, he will soon be a rich man!
John Watson
jwatson@esoc.esa.de

Microsoft and Macintosh support

Dear Sir,
Thank you for publishing my letter about Microsoft Visual C++ 4.0 cross-development kit for Macintosh and raising the issue of support for the product with Microsoft.

Since last month, we have been told by a member of Developer Services at the recent Microsoft Developer Day in London that the product and all support had been discontinued. Another Microsoft employee said it was rumoured that the product was still being used internally by Microsoft for some projects.
If this is the case:

1. Why will Microsoft not simply issue a statement to this effect?
2. Why is the product still mentioned on Microsoft's Web site without the warning that 'this product contains known problems that we will not fix'?
3. Is there any way for marooned users to get hold of a working compiler (one presumes that the version used internally by Microsoft has had its bugs fixed)?

Interestingly, the product is now offered for sale at $199 (10% of the original price). Perhaps we can see why. If you want a good laugh, read the promotional material for this special offer at *http://www.microsoft.com/visualc/startpgs/productst.HTM#macver*. However, as a user who paid full price for the product (which is small beer compared to the time invested in it), I'm closer to tears.

Other developers in a similar position might be interested to know that there is an American company, CoreTek, who has developed an inexpensive library that does work around some of the problems of the Microsoft VC++ for Macintosh MFC implementation, in particular with respect to file handling and COM ports. Its web address is: *http://www.coretek.com/*.
Anonymous

The rebirth of C++

From the birth of C++, class libraries have tended to strongly reflect the legacy of Smalltalk, but the Standard Template Library is set to change that. Kevlin Henney explains why C++ finally has an idiom of its own.

Depending on your background as a C++ developer you will either have been using templates for some time (for generic container classes, in most cases), or will still be under the impression that they are some recent and advanced feature of the language that you can probably live without.

Originally defined in the ARM (*The annotated C++ reference manual*, 1990), they have been refined and developed in most mature and capable compilers and class libraries for some time now (an honour from which Microsoft tried hard until relatively recently to exempt itself). That they are something every C++ programmer should know about is emphasised further by looking at the draft standard (see *The ISO C++ Standard* boxout). The most significant part of the draft standard (about 350 pages of the document) is the library – and it's nearly *all* templated.

Even for more template-wise programmers it may come as a shock that generic, type-safe containers constitute an almost minority usage of templates. The purpose of template classes and functions is normally to generalise a data structure or algorithm with respect to a type, but template parameters may also be constants or objects with external linkage. Templates eliminate much of the need for clumsy and error-prone macros, and code obscured with casts and *cut-and-paste* programming.

```
typedef basic_ios<char>      ios;
typedef basic_istream<char> istream;
typedef basic_ostream<char> ostream;

extern istream cin;  extern ostream cout;
extern ostream cerr; extern ostream clog;

typedef basic_ios<wchar_t>      wios;
typedef basic_istream<wchar_t> wistream;
typedef basic_ostream<wchar_t> wostream;

extern wistream wcin;  extern wostream wcout;
extern wostream wcerr; extern wostream wclog;
```

Listing 1 – Stream types and objects in <iostream>

```
template<typename Numeric> class complex {
public:
    Numeric real() const;
    Numeric imag() const;
    complex &operator*=(const Numeric &);
    template<typename Other>
      complex &operator*=(const complex<Other> &);
    ...
};

template<typename Numeric>
  numeric abs(const complex<Numeric> &);
...
```

Listing 2 – Overview of the complex class.

Library structure

Table 1 contains an outline of the library clauses, standardese for chapters. (Note: the traditional C standard library is fairly implicitly included in the C++ one, with only minor differences significant enough to be mentioned in the standard).

Once you've gone over this (fairly sizeable) shopping list of features, it is possible to start discerning certain principles and strong factors in the library's design. Wherever you see the words *iterator* or *algorithm*, you can be sure that templates are involved.

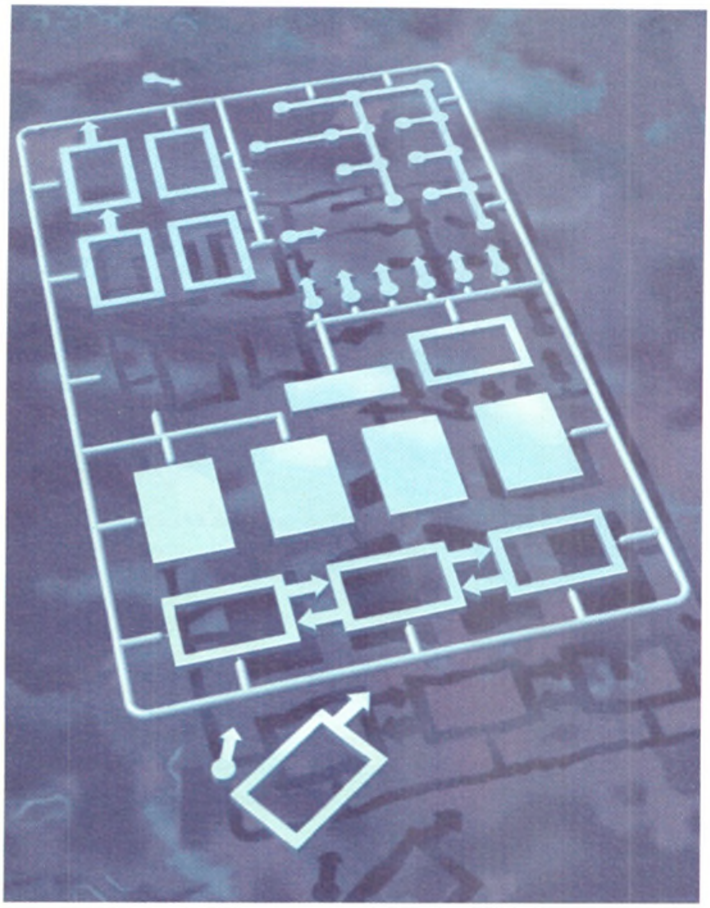
With the exception of a couple of macros (inherited from the C library), the features of the standard library all live in the namespace `std` to prevent collision with identical names in other class libraries (most popular for a string class? `string` of course – putting your code in a namespace solves this). The standard library headers have no suffix – I guess that's one way to resolve the cross-platform suffix holy war.

Lateral thinking

Whereas inheritance can be considered to be *vertical* generalisation, templates represent *horizontal* generalisation: similar data and functional structure with variant element types. The library makes good use of this, leaving it open for extension with new data types.

A good example of this is the string classes: originally `string` was a class containing a sequence of `chars`. This may seem natural enough to many Western programmers, but the internationalisation of string and character handling hinges on wide character types (`wchar_t` in C++) which can represent extended character sets such as Unicode. To address this, a wide character string `wstring` was added, which looked pretty much like a *cut, paste and replace* job from `string`. But what if someone came along with a user-defined character set? Even though classes are supposed to be first-class citizens of the language, these would be excluded for use in strings.

The solution adopted by the library was to introduce a template class `basic_string<>` that describes the general structure and operations on a string type, but leaves the character type as a template parameter. `string` and `wstring` survive as `typedefs` of `basic_string<>` like:



```
typedef basic_string<char>       string;
typedef basic_string<wchar_t>    wstring;
```

In truth, there are a couple of other template parameters for `basic_string<>`, but these are defaulted (see *One last thing* in this issue) and for most purposes can be ignored. One of them is a *trait class* to be used for the character type. A trait class is a specialised template which provides a number of static constants and functions, allowing compile-time lookup on the properties associated with a type.

A similar design makeover was applied to the I/O streams, resulting in some of the common type names and objects shown in Listing 1.

Complex but simple

The design of the complex number class in the numerics section of the library was considerably simplified by template generalisation. When designing a maths library, the most important consideration to make is to what precision should its calculations be. Overloading can help to some degree in making this transparent, but what about complex numbers? Should they be `float` for space efficiency, `double` for common use and precision, or `long double` for maximum precision? Or perhaps another user-defined number type?

```
int              abs(int);
long             abs(long);
float            abs(float);
double           abs(double);
long double      abs(long double);
```

The solution is to describe complex numbers as a structural family based on a supplied numeric type, as in Listing 2. This again leaves the library generalised and fully open to extension with user-defined types. In some ways templates can be considered an extended form of overloading: a template function defines an infinite family of functions with the same name and same structure, differing only with respect to some parameter(s). A template class defines an infinite set of structurally similar types with similar motivation. Just as template functions can be specialised for a particular type, template classes can be specialised to provide a more efficient or completely different implementation of a class for some parameter.

Although it was introduced for other purposes, the `typename` keyword can be used in place of `class` for specifying template parameter type names. Not all compilers support this feature yet, but it makes more sense to use it in the long run. As you can see from Listing 2, the template functions can work with mixed precisions of complex numbers, so it is possible to write code like the following:

```
complex<double> whisky;
complex<float>  malt;
whisky *= malt;
complex<my_number_class> mac;
whisky *= mac;
```

STL

What most people now think of when 'template' occurs in the same sentence as 'standard library' is the standard template library or STL. Developed by Alex Stepanov and Meng Lee at HP, the STL is a library of generic components – algorithms, function object types, object adaptors, containers and iterators – based on thorough operational specifications. Andrew Koenig, project editor for the ISO/ANSI committees, suggested that it should be put together as a proposal for the C++ standard library, and the rest has become history. The term STL is occasionally used loosely to refer to the whole standard library, but strictly speaking (although it constitutes a major part) it is a proper subset.

As its name suggests, the STL makes heavy use of templates — in some places using features that have been recently standardised but are not widely supported by compilers – and *no* use of runtime polymorphism. The non-inheritance approach comes as a surprise

```
template<typename RandomAccessIterator>
  sort(
    RandomAccessIterator begin,
    RandomAccessIterator end);
template<class InputIterator, class OutputIterator,
         class T>
  OutputIterator remove_copy(
    InputIterator begin, InputIterator end,
    OutputIterator result, const T &value);
template<class InputIterator, class T>
  InputIterator find(
    InputIterator first, InputIterator last,
    const T &value);
```

Listing 3 – A couple of STL algorithms.

```
double data[size];
... // set elements
sort(data, data + size);
```

Listing 3a – Sorting an ordinary array.

```
istream_iterator<string> input(cin), end;
ostream_iterator<string> result(cout);
remove_copy_if(input, end, result, "foo");
```

Listing 3b – Filtering elements from input to output.

www.SILICONRIVER.co.uk

Video Training for C and C++

Over 5,000 Silicon River video training courses have been sold to companies, universities and individuals wishing to learn or improve their C/C++ skills. Why not join them in getting the most cost effective, high quality C/C++ training available?

C Video Course Personal Edition — £199.95
(6 videos with over 13 hours of training, 365 page workbook and source disk)

C++ Foundation Course Personal Edition — £199.95
(6 videos with over 12 hours of training, workbook and source disk)

C++ Advanced Module - STL — £49.95

C++ Advanced Module - Exception Handling — £49.95

Call for details of Company and Educational Packs.

GREENLEAF SOFTWARE

Now Distributed by Silicon River

Greenleaf CommLib v5.2 — £259
The industry standard asynchronous comms library for C, C++, Delphi and Visual Basic. This feature-rich product supports file transfers under XModem, YModem, ZModem, Kermit, Compuserve B+ and ASCII. Also includes numerous other features including handshaking, port status and monitoring. Supports all PC platforms and comes with full source.

Greenleaf Comm++ v3.0 — £229
Written in C++ and assembly language, provides extensive functionality similar to CommLib, but designed from the ground up as an object oriented alternative. Also supports all PC platforms and comes with full source.

PREVIOUSLY £389 - NOW JUST
Greenleaf ViewComm v1.0 for WIN 95/NT — £279
ViewComm for Windows offers the ability to see what's on the line when you need to work on protocols, comm software development or debug an asynchronous communications link. ViewComm comes complete with additional cables. (DOS version also available.)

These and other Greenleaf products are available from stock.

POET ODBMS v4.0

The Object Database for Desktop, Workgroup and Internet Applications.

OBJECT DATABASE MANAGEMENT GROUP

BEST OO-PRODUCTS 1995 — READER'S CHOICE OBJEKTspektrum

JOLT Cola PRODUCT EXCELLENCE AWARD 1995

POET 4.0 is the no.1 object database for Windows users. POET not only allows you to store the objects you create in C++, Java or Visual Basic, but also allows you to specify the relationships between these objects.

POET is also able to scale with your application. Use the same API regardless of whether you are writing for single user Windows or in a multi-user, multi-processor environment.

You can use POET v4.0 with:
C++ - Visual Basic - Java - OLE Automation

DOWNLOAD YOUR FREE POET EVALUATION VERSION. AIM YOUR BROWSER AT:
www.siliconriver.co.uk

SYMANTEC INTERNET TOOLS

Cafe for Win 95/NT — £79
A complete Java Integrated Development and Debugging Environment, with productivity tools for the more experienced developer.

Visual Cafe for Win 95/NT — £139
Visual Cafe's Rapid Application Development (RAD) environment provides the easiest way ever to create Java applications.

Visual Cafe Pro for Win 95/NT — £call
The Pro version of Visual Cafe provides simple integration of major databases into your applet or application (Sybase SQL, MS SQL Server, MS Access, Oracle and OBDC) using dbAWARE controls and wizards.

dbANYWHERE Server Now Available — £call

SELLING VERY FAST AT JUST
Symantec C++ v7.5 for WIN 95/NT — £69
The award winning previous version (v7.2) cost £409. This improved version costs just £69 (a £340 saving). Where else can you get a C++ compiler with MFC 4.2b for so little money?

Call for special Educational Pricing on Symantec Internet & C++ Products.

Silicon River
(0181) 317 7777

Silicon River Video Training Courses, Greenleaf Software Products and Symantec Internet Tools are available direct from Silicon River or from most leading development tools specialists.
Silicon River Ltd, 106-108 Powis Street, London, SE18 6LU. TEL: (0181) 317 7777, FAX: (0181) 316 7778, email: sales@siliconriver.co.uk www.siliconriver.co.uk
All trademarks are recognised. Prices exclude delivery and VAT.

CIRCLE NO. 225

to many, but this should not be taken to mean that the library components are inflexible. On the contrary: components are heavily parameterised, but most of the parameterisation is at compile time – giving rise to the concept of *compile time polymorphism*. It is easy to build an efficient polymorphic container hierarchy using STL containers as the underlying implementation, but not vice-versa. As such, the STL constitutes the more fundamental approach to library components.

An important feature of particular algorithms is their complexity, ie their relative performance related to the number of elements they are operating on. The three categories of performance comprise algorithms that operate in *constant*, *linear* and *logarithmic* time. The STL specifically defines the relative cost of each operational expression. This, as well as the notion of interface, is used to fully define what a type is – the STL is firmly based on abstract data types, rather than strict traditional object-oriented concepts. The STL supports *generic programming*, based on compile-time polymorphism, ie common name and common purpose.

The emphasis on algorithms is what endows the STL with its own particular flavour, once and for all giving C++ containers and iterators a specific style. Of late, C++ libraries have been growing their own standard idioms and becoming less like Smalltalk hand-me-downs. The STL's fresh and alarmingly simple approach completes the picture. Or is that *old* and alarmingly simple? Choosing the data structure to simplify the algorithm is hardly new advice, but it is this approach more than any other that typifies the library.

Iterators

Iterators are the key to understanding how algorithms, containers and the other library components fit together. The motivation for iterators is described by Gamma, Helm, Johnson and Vlissides (in *Design patterns*) as:

A way to access the elements of an aggregate object sequentially without exposing its underlying representation.

Thus they are objects that allow a user to traverse an object – such as a stream or a container – encapsulating both the traversal strategy and the representation of what it is iterating over. The STL iterators form a sort of glue layer between algorithms and the containers they operate on.

STL iterators have a straightforward pointer-like interface, going against the trend in other class libraries for ever fancier and more all-knowing iterators. Many operations on containers are specified in terms of iterators rather than indices (see Listing 3): `find` returns an iterator to the first occurrence of the element to be

```
vector<double> data;
... // set elements
sort(data.begin(), data.end());
```

Listing 4a – Sorting a `vector<>`.

```
list<string> words;
... // set elements
list<string>::iterator found =
    find(words.begin(), words.end(), "foo");

if(found == words.end())
    cout << "Not found" << endl;
else
    cout << *found << " found" << endl;
```

Listing 4b – Finding a value in a `list<>`.

The ISO C++ standard

The ISO C++ standard is nearing completion (see *One last thing* in this issue). Extensions to the library and language were effectively closed long before the first committee draft release in April '95, and the standard is now in second committee draft release (CD2). After that it will progress to draft international standard (DIS), before becoming an international standard – 1998 in the current schedule. Although the standardisation effort started off as a solely ANSI venture, it quickly became an international one. Hence it will be an ISO standard that is produced and not an ANSI one.

It is important to understand that what the C++ standards community has been doing in the interim is debugging the language and library – the standard is just like any other piece of software or specification in this respect!

Clause	*Title*	*Contents*	*Headers*
Clause 17	Library Introduction	Describes library contents and conventions	
Clause 18	Language Support	Defines types and operators intrinsic to the language, such as `new` and `delete`, RTTI, exception handling, and numeric implementation limits	`<new>` `<typeinfo>` `<exception>` `<limits>`
Clause 19	Diagnostics	Standard exception classes	`<stdexcept>`
Clause 20	General Utilities	Defines general template parameter requirements, memory management classes such as `allocator<>` and `auto_ptr<>`, memory management algorithms and iterators, `pair<>`, templated relational operators, and templated function object types	`<memory>` `<utility>` `<functional>`
Clause 21	Strings	String and character trait classes templated on character type	`<string>`
Clause 22	Localization	Internationalization classes templated on character type	`<locale>`
Clause 23	Containers	Templated container classes covering sequences, such as `vector<>` and `list<>`, adapted sequences, such as `stack<>`, and associative containers, such as `map<>` and `set<>`	`<bitset>` `<deque>` `<list>` `<map>` `<queue>` `<set>` `<stack>` `<vector>`
Clause 24	Iterators	Iterator requirements, traits and predefined iterators	`<iterator>`
Clause 25	Algorithms	Templated algorithms defined to operate on iterators	`<algorithm>`
Clause 26	Numerics	`complex<>` for complex numbers, `valarray<>` for mathematical vector and sliced operations, and generalised numeric algorithms	`<complex>` `<valarray>` `<numeric>`
Clause 27	Input/Output	I/O stream classes templated on character type, providing a hierarchy of classes for input, output and manipulation on general streams, file streams and string streams	`<iosfwd>` `<ios>` `<istream>` `<ostream>` `<iomanip>` `<iostream>` `<fstream>` `<sstream>` `<streambuf>`

Table 1 – Breakdown of the C++ standard library by clause.

STL resources

ftp://butler.hpl.hp.com/stl contains the original HP implementation of the STL and the STL FAQ.

http://www.sgi.com/Technology/STL holds the source for the SGI STL as well as a complete online manual.

http://www.dinkumware.com/refxcpp.html provides access to the Dinkum C/C++ Library Reference. This is a complete online manual to the whole standard library. This site also contains a summary of the most recent changes to the draft standard.

http://www.maths.warwick.ac.uk/c++ holds a number of useful links to other C++ sites, including some online STL tutorials. The current draft of the standards is also downloadable.

ftp://research.att.com/dist/C++std/WP holds the previous and current committee drafts (CD1 and CD2).

http://www.ocsltd.com/c++ houses the C++ Beyond the ARM pages which are a good guide if you feel a little overtaken by some of the language changes that have happened since the original base document.

ftp://ftp.aw.com/cseng/authors/stroustrup/stroustrup2e, alternatively, holds an addendum to Stroustrup's original C++ Programming Language (2nd edition) which describes the changes that have happened to the language as a result of standardisation. This is available from the Addison-Wesley Web site or in recent reprints of the book.

http://www.objectspace.com/jgl holds details and a downloadable copy of the JGL (Generic Library for Java), an STL inspired library of containers, iterators and algorithms for Java.

STL Books

STL tutorial and reference guide – C++ programming with the Standard Template Library by David R Musser and Atul Saini, Addison-Wesley, ISBN 0-201-63398-1

This book has an accurate and self descriptive title. It is a complete overview of the HP STL with some modifications in line with changes made to the standard.

The STL <Primer> by Graham Glass and Brett Schubert, Prentice Hall, ISBN 0-13-454976-7

This book is pretty much a reference to the original ObjectSpace STL<ToolKit>, but still serves as a general guide to the STL and the standard string class.

C++ programmer's guide to the Standard Template Library by Mark Nelson, IDG, ISBN 1-56884-314-3

This is a chunky introduction and reference to the original STL, complete with a disk containing the HP implementation.

STL for C++ programmers by Leen Ammeraal, Wiley, ISBN 0-471-971812

This is a more up to date book concentrating on the programmatic aspects of the STL, rather than just being a raw reference.

The Standard Template Library by P J Plauger, Alexander Stepanov, Meng Lee, and David R Musser, Prentice Hall, ISBN 0-13-437633-1

This book is put together by the key people involved with the implementation of the STL from birth, through commerce, to standard – I guess it qualifies as 'from the source'. It should not be confused with Plauger's earlier book The draft C++ standard library which is a description of the pre-STL standard library – a little premature, as it turns out, because most of the book was invalidated with the incorporation of the STL and it remains interesting only from an academic perspective.

Standard Library and STL Implementations

***Rogue Wave Software* (http://www.roguewave.com)**

The *Standard C++ Library* from Rogue Wave is an implementation of the standard library that has been licensed by a number of leading companies – including Borland, Sun Microsystems, Hewlett-Packard and Silicon Graphics – for inclusion with their C++ compilers. Rogue Wave has also integrated its popular Tools.h++ library with the standard library, positioning Tools.h++ as a higher level wrapper.

***ObjectSpace, Inc.* (http://www.objectspace.com)**

ObjectSpace had one of the first commercial, cross platform implementations of the STL in the form of the form of its *STL<ToolKit>* library. It has now rebranded its broadening product line to reflect the move towards standardisation. The *Standards<ToolKit>* corresponds to the standard library and forms part of their *Foundations<ToolKit>*. Other products are built on this lower layer, but with an increasing number of compilers supporting the standard library the *Standards<ToolKit>* will eventually become redundant and can be swapped out in favour of the 'native' library.

***Plum Hall, Inc.* (http://www.plumhall,com)**

P J Plauger's standard library implementation is available through Plum Hall. This is the library that is available with the Microsoft Visual C++ 4.2 compiler.

***Dinkumware, Ltd.* (http://www.dinkumware.com)**

P J Plauger is president of this company through which his implementation of the library and a summary of recent changes to the standard are also available.

Hewlett-Packard Company

HP does not market the STL, but as the home of STL it still makes the original STL freely available from one of its sites. HP stopped development of STL at the end of 1995, and has commercially licensed Rogue Wave's standard library.

Silicon Graphics, Inc.

The story with SGI is in many ways similar to that for HP, except that Alex Stepanov is now working for them, and they are still funding development and research into the STL, which is freely available. The SGI STL implementation is based on (but contains many improvements over) the original, including hash tables, plus more efficient and thread-safe code. The principal version is usable with Unix compilers based on the EDG (Edison Design Group) front-end. A dumbed-down version is also available that will compile with the Microsoft and Borland compilers.

http://www.sgi.com/Technology/STL

Free Software Foundation

The standard library available with g++, the GNU C++ compiler, is based on the HP STL.

matched; `insert` and `erase` operate on single iterators or a range specified by two iterators. It is as if iterators are an abstraction of pointers into a container, in the same way that pointers can be used within a plain old C array.

A benefit of this is that algorithms can be written in a generic way, leaving it to iterators to deal with the underlying container types. They do not tie you down to a particular implementation, eg you need not inherit from `VendorSpecificClass`, and they can even be used with plain old arrays. Both of these points relate to efficiency which, despite C++'s high level features, is still something that must be considered as part of its 'spirit'.

The standard specifies five categories of iterator (ie five sets of requirements), depending on the kinds of operation supported (see Table 2). With the exception of input and output iterators, each category builds on the category before it. These categories are represented in the library as *tag types* which can be used with iterator trait classes to determine the properties and performance of an iterator.

The general algorithms for sorting, reversal, filtering, etc are written in terms of iterators rather than containers, decoupling them from the base data structures they are operating on. For instance, to fully sort a sequence all that is required is a random access iterator that points to its first element and one that points to the element just past the end, as you can see in Listing 3a. Through iterators, algorithms can even operate on traditional C arrays, or more abstract sequences such as a console input stream (as in Listing 3a).

One point of note: by 'just past the end', I mean that such an iterator is not legally dereferenceable as part of the container, but is notionally just after the last legal element. This means that two iterators (`begin` and `end`) can be used to delimit the range of iteration as `[begin, end)` – ie the end is not included. The *past-the-end* marker is a useful *out-of-band* value for denoting conditions like a failed `find`. Thus algorithms need be written only once.

Category	*Extends*	*Operations*	
Input		Equality, increment and dereferencing for reading	Simple iterators for single pass algorithms
Output		Equality, increment and dereferencing for writing	Single pass result output
Forward	*Input* *Output*		Useful for multi-pass, unidirectional algorithms
Bidirectional	*Forward*	Also supports decrement operator	
Random Access	*Bidirectional*	Full pointer arithmetic	All operations can be achieved in constant time

Table 2 – Iterator categories.

Expression	*Operation*
`c.front()`	Returns a reference to the first element, i.e. `*c.begin()`
`c.back()`	Returns a reference to the last element, i.e. `*--c.end()`
`c.push_front(x)`	Inserts a new element at the front of the sequence
`c.push_back(x)`	Appends a new element to the back of a sequence
`c.pop_front()`	Drops an element from the front of a sequence, no value returned
`c.pop_back()`	Drops an element from the back of a sequence, no value returned
`c[n]`	Index the nth element, counting from 0
`c.at(n)`	Safe index of nth element, i.e. throws `out_of_range` on failure

Table 3 – Optional operations for sequences.

Container	*Iterators*	*Description*
`Vector<>`	Random Access	The standard array class which supports operations at the tail end in constant time
`List<>`	Bidirectional	The standard doubly linked list class which supports general insert and erase operations in constant, but not direct indexing
`Deque<>`	Random Access	Like `vector<>` but supports length changing operations at both ends

Table 4 – Standard sequence classes.

Containers

STL containers are quite lightweight in terms of their specification, since they rely mostly on iterators and external algorithms for functionality. They define a number of parameterising types, such as the reference and pointer types used for the containee type, as well as a number of operations that the container must support (like default construction, copy construction, equality and size querying). These operations are specified in terms of valid expressions along with their expected behaviour and complexity. For instance, the complexity of the equality operator is linear: the time taken to determine equality of two containers is no worse than proportional to the number of contained elements. All containers support queries to access the start and one-past-the-end iterators in the form of `begin()` and `end()` member functions. Like the paramaterising types, a container's iterator types are nested within its class definition. Specifically, these are `iterator` for mutable access and `const_iterator` for read-only (or browsing) access.

Sequences are a specialised form of container, which are required to satisfy some constraints placed on the container, in addition to operations like insertion and deletion of elements. A sequence is a linear structure based on element position. A number of optional operations are also specified for sequences (shown in Table 3).

The library provides three standard sequence classes (see Table 4). The string class has also been revamped to support a number of sequence operations, most notably iterators. To be fair, the `basic_string<>` class never quite made the grade as a fully fledged container: it is a little disappointing, but history and evolution had a large part to play in this. The sort operation shown in Listing 4a can be written for other random access sequences, as in Listing 4b.

The standard also specifies a set of requirements for associative containers, with standard implementations for the common containers `set<>`, `multiset<>`, `map<>` and `multimap<>`. The `map` class implements what is sometimes known as a *dictionary* or an *associative array*, and the `multi` classes what are commonly referred to as `bags`.

Output stream

The C++ standard library is a large and powerful set of components, only a few of which have been touched on here. Most importantly, certain parts of its design philosophy clearly stand out as being quite radical but at the same time simple. In particular, the design of the STL has given C++ its own set of high level idioms – something that is discussed in more detail in *Foundations for native C++ styles* by Koenig and Stroustrup (*Software practice and experience*, 25(S4), 1995). Interestingly, the STL design has been adopted for the JGL, ObjectSpace's generic library for Java, proving that generic programming ideas can be translated to other languages.

The finalisation of the standard with its library is bringing C++ into its third era. The first was the early-adopter phase from release in 1985 to around 1990. The second was heralded in 1991 by the successful publication of a number of good quality books for practitioners. The third era sees a new style for C++ use quite different from the early C-like or Smalltalk clone attempts at using the language: C++ finally has an idiom all of its own. ■

Kevlin Henney is a member of the BSI C++ panel and runs the UML/OMT User Group. He works for QA training as a consultant, lecturer and course manager, and can be contacted at khenney@qatraining.com.

Attention Developers

Database Programmers Retreat

Programmers Teaching Programmers

In picturesque Painswick, Gloucestershire

Delphi

D101 - Programming Fundamentals
D201 - Advanced Programming
D301 - Client Server Programming
D401 - Programming 32 bit Delphi

Visual Basic

VB101 - Programming Fundamentals
VB201 - Advanced Programming
VB301 - VB 4 Upgrade

Visual FoxPro

VFP101 - Programming Fundamentals
VFP201 - Advanced Programming
VFP301 - Client Server Programming

Access

AC101 - Fundamentals of Access
AC102 - Application Development

CA-Visual Objects & Clipper

VO101 - Programming Fundamentals
VO201 - Advanced Programming
CL101 - Programming Fundamentals
CL201 - Advanced Programming

About *DPR*

DPR, run by Rick Spence, is a training, consulting, and development company with offices in St. Augustine, FL, and Painswick, UK. We specialize in teaching programmers how to develop database applications, and in developing applications for other companies. Our motto is "Programmers Teaching Programmers". We're programmers ourselves, and we only teach programming.

Classes

All classes are taught hands-on, on state of the art, Pentium PCs. Each student has his / her own PC, which is linked to an NT server running SQL Server, Oracle, and Watcom databases. Class sizes are small, ensuring individual attention. The beautiful surroundings provide a stress-free environment - ideal for learning.

On-Site

We can teach the same classes on-site, and we can customize them for small groups. Although we know you'll enjoy a visit to Painswick, *if you have 3 or more programmers it can make economic sense for us to come to you.* We can also augment our on-site visits with an element of consultancy - we're flexible, just ask.

Instructors

Our instructors are practicing developers with years of both development and teaching experience. Based on practical experience, they supplement course material with real world experience and examples. Rick Spence is well known for his top selling database books, his dynamic and entertaining speaking style, and as a contributing editor to Databased Advisor magazine.

DPR

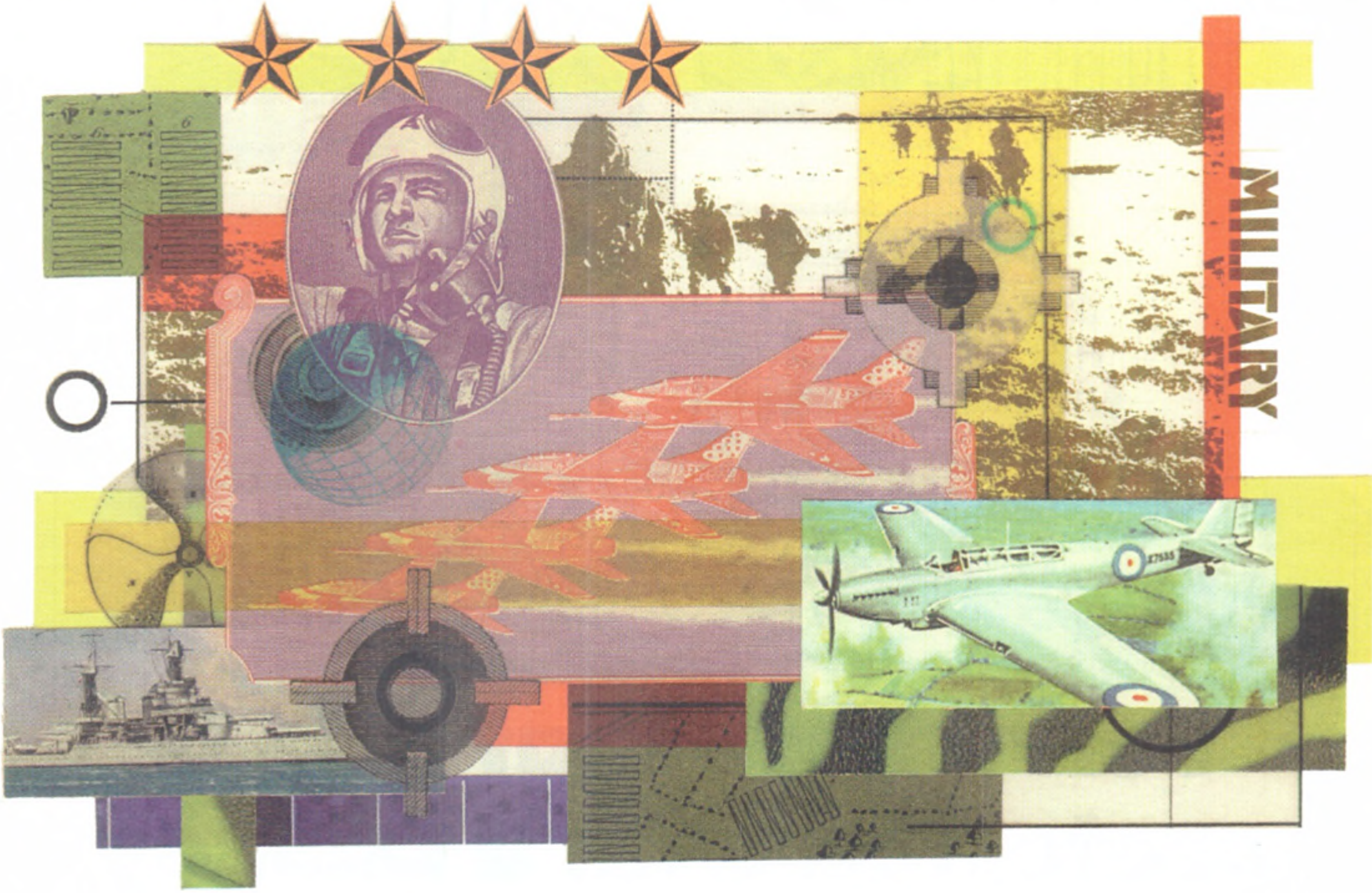
In Europe: Call +44 (1452) 814 303 or fax +44 (1452) 813 918
In USA: Call 1-800-279-9717 or fax (904) 824 4622
Email: 71760.632@Compuserve.com

Borland *Connections* Partner

CIRCLE NO. 270

VISUAL FOX PRO / VISUAL OBJECTS / DELPHI / CLIPPER / VISUAL BASIC / ACCESS

Ada Better than C++?



When moving beyond C, C++ tends to be the language of choice for current industrial programming, but Ada could be a more robust alternative. Gavin Smyth rediscovers the language.

My first encounter with Ada was about nine years ago, and I found it a very large cumbersome language to use – mainly, I think, because I was trying to employ it on a problem for which it was not particularly suited. The application I was working on was intended to run under the X Window system, and Ada and X did not get along at all well. At around that time, C++ was starting to make its presence felt, and seemed overall to be a much slicker way to move beyond the strictures of C.

Almost a decade down the line, I am reconsidering. C++ has become a much larger language than I anticipated; Ada has recently experienced a rejuvenation; and I have had a lot more experience of 'real world' programming. For many problems, Ada may well be my first choice instead of C++, and I will try to explain why.

The major factor that stimulated my current interest in Ada was the availability of a low-cost, high quality compiler which runs under DOS. In an earlier article ('Go-faster sprites' *EXE*, July 1996) I discussed DJGPP, an excellent port of the GNU C++ compiler for DOS. There is a GNU-based Ada compiler, called GNAT (developed at New York University, but now supported and maintained by ACT – Ada Core Technologies), which happens to be useable with DJGPP. GNAT also runs on various Unixes (including, of course, Linux), and Windows 95/NT.

Common criticisms of Ada

People often discount Ada because of some common beliefs and misconceptions. Let's go through some of these:

1. The Ada language is very large – yes, that is true. However, the formal language definition, even at over 500 pages, is comparable in

```
#include <iostream.h>

class A {
public: virtual void print() { cout << "I am an A\n"; }
};

class B: public A {
public: void print() { cout << "I am a B\n"; }
};

void printIt( A& a ) { a.print(); }

void main() {
  B b;
  printIt( b );
}
```

Listing 1 – C++ class example.

Sorry Sir, – that title is protected with a DESkey

You don't need to travel to find cheap software; international pirates now offer mail order, millions of pounds worth of commercial software is duplicated on 'Gold' CD-ROM's in the UK and Europe. Such is the disregard for software licence agreements by the business community, that emulators for lesser dongles are advertised in the European Computer press.

DESkey security is built into the hardware. ASICs and microprocessors running proprietary algorithms provide real protection. A comprehensive range of drivers and our software protection utility, DESlock, work to bring the highest level of security with the minimum of effort.

Call today for product information, demonstration units and technical advice.

Don't just Dongle it – DESkey it

PC MAC PC Card UNIX etc

Designed for Microsoft Windows 95 · PC CARD · Member of FAST Promoting the legal use of software · QUALITY ASSURED FIRM ISO 9001 SGS · ACS ASIC · Mac OS · Ready for OS/2 WARP

DESkey

CIRCLE NO. 229

Data Encryption Systems Limited *Silver Street House, Silver Street, Taunton, Somerset TA1 3DL*
Telephone 01823 352357 Fax 01823 352358 BBS 01823 352259 E:mail sales@des.co.uk www.des.co.uk

size to other language descriptions which encompass standard support libraries, and is in fact smaller than the draft C++ standard.

2. Ada is difficult to learn – certainly, if you are moving from C to C++, it seems that the steps are small. However, coming at C++ or Ada from scratch, there may not be much of a difference, though I think it is true that you need to know more Ada to write your first Ada program than C++ to write your first C++ program. One extra thing to take into account is programming conventions – there are whole books full of informal rules for C++. You get that for 'free' with Ada: there is only a single style definition.
3. Ada compilers are expensive – Not true. GNAT is made freely available (though you will get better support and more up to date releases of the tool set if you pay).
4. Ada is not object oriented – this used to be true, but is no longer the case (as we'll see later).
5. Ada is inefficient – it used to be difficult to do low-level programming (for example: pointers were not very flexible; and interfacing with other languages, such as assembler, was well nigh impossible) but again, this is no longer the case. Another criticism is that features like rigorous range checking are expensive, but compile-time analysis can remove redundant checks, and GNAT permits many of the checks to be disabled via command line switches. It is still true that in providing a common definition of, say, tasking, the language cannot exactly match a particular architecture, but that will be true of any portable system, and can be worked around in specific cases regardless of the language.
6. Lack of library support – there are more C++ libraries in existence than Ada bindings. This may be partly because the Ada definition encompasses many features which C++ programmers must turn to libraries for, such as tasking and numeric processing. However, there are many areas where C and C++ get bindings first (and Ada later, if at all), such as links to GUI systems like Windows and X.
7. Ada is only for military applications – well, the American DoD was the driving force behind Ada, but so what? Ada was designed to have general purpose applicability and the only real worry is the language dying and leaving you with a huge amount of unsupportable code. The sheer amount of C in existence assures its long term future in the same way as Cobol lumbers on, but I believe that Ada has as long a life expectancy as C++ (although I will be the first to admit that I have no hard evidence to support this).

File classes.ads.

```
package Classes is
  type A is tagged null record;
  procedure Print( Aa: in A );
  type B is new A with null record;
  procedure Print( Bb: in B );
end Classes;
```

File classes.adb.

```
with Text_IO; use Text_IO;

package body Classes is
  procedure Print( Aa: in A ) is
  begin
    Put( "I am an A" );
    New_Line;
  end Print;

  procedure Print( Bb: in B ) is
  begin
    Put( "I am a B" );
    New_Line;
  end Print;
end;
```

File main.adb.

```
with Classes; use Classes;

procedure Main is
  procedure Print_It( Aa: in A'Class ) is
  begin
    Print( Aa );
  end Print_It;

  Bb: B;

begin -- Main procedure
  Print_It( Bb );
end Main;
```

Listing 2 – Ada class example.

The evolution of Ada

You may be familiar with the original Ada definition, which was quite restricted. For example, although it included *generics* (a bit like C++ templates), it did not support any useful form of derived classes; and although tasking was part of the language definition, the mechanism was very cumbersome and people were likely to write their own lightweight threads if they required such facilities. This version of the language is generally referred to as Ada 83 these days, since a new definition (Ada 95) has been developed. Ada 95 extends Ada with some object-oriented features, more lower-level control, much improved pointer (access type) manipulation, more sophisticated interoperation with other languages, more flexible and lightweight tasking facilities, and extensions to the language covering such things as numerics and distributed processing.

Many of these features have been provided in the past by vendors, in different ways, but they are now a standardised part of the language.

Comparison with C++

I could spend many tedious pages listing all the differences between Ada and C++, but instead I choose to concentrate on the small number of areas which I consider the most fundamental for the sort of systems I tend to program – that is, real-time multi-threaded programs that need to make efficient use of limited resources.

First of all, I will look at what I consider are the more useful aspects of object-orientation in the two languages: inheritance and dynamic dispatching. C++ classes are, roughly speaking, known as tagged records in Ada. Listing 1 illustrates a short C++ example: two classes with a single virtual function (and a lot of good programming practices thrown out the window for brevity). As expected, this prints out 'I am a B'. Listing 2 shows the equivalent in Ada (with the minor change that I have not used pointers). So what are the differences? The Ada program is a lot longer – for a start it requires an explicit public interface

Assembler source code.

```
.globl _addtwo
_addtwo:
  movl 4(%esp),%eax
  addl $2,%eax
  ret
```

Ada example using this assembler routine.

```
with Text_IO; use Text_IO;

procedure Asmbit is
  package Int_IO is new Integer_IO( Integer ); use Int_IO;
  function Add_Two( Val: in Integer ) return Integer;
  pragma Import( Assembler, Add_Two, "addtwo" );

begin -- main "Asmbit" procedure
  Put( "Add_Two( 34 ) is " );
  Put( Add_Two( 34 ) );
  New_Line;
end Asmbit;
```

Listing 3 – Importing a routine.

definition for the overrideable functions in a package, so I created `classes.ads`, the equivalent of a C++ header file. (I could get away without a header in the C++ example because it was so trivial).

That besides, I prefer C++'s explicit labelling of dynamically dispatchable functions as `virtual`. Ada has implicit run-time type information (RTTI), but can optimise dynamic dispatching to static linking. It may be useful to be able to avoid the overhead of RTTI in some situations, but it is a valuable debugging aid in complex systems. I also feel more comfortable with C++ treating a class parameter differently to other parameters (putting it in front of a **.** or **->**).

Ada does not support multiple inheritance, something I have thus far found little use for in C++. C++ has a simple to use constructor/destructor mechanism (well, simple to use until you try to make it interact with inheritance trees, when you usually have to remember to make the destructor virtual; or until you use exception handling, when you have to take care to avoid resource leaks). Ada, on the other hand, encourages explicit create/delete functions. Package bodies can execute start-up code (a bit like static variable initialisation in C++), and types can be derived from the `Ada.Finalization` package, giving something like constructors/destructors, though it does not appear to be as flexible. In summary, there seems little to choose between the languages for practical object-orientation.

Another area I am interested in is low-level machine access: I write most of my interrupt service routines in assembler for efficiency, but need to interact with them from higher level code (Ada does include some interrupt handling support, but I haven't felt the need to explore it yet). Generally, I make the point of communication a shared static variable – easy in C/C++ (just reference the variable as `extern volatile`), impossible in Ada 83 without stepping outside the language definition. In Ada 95, however, it is simple: just declare a native Ada variable, and use the language pragmas:

```
My_Var: Byte;
Pragma Import( Assembler, My_Var, "myVar" );
Pragma Volatile( My_Var );
```

Similar pragmas are available for sharing variables and routines

```
with Text_IO; use Text_IO;

procedure Tasks is

  task type Char_Printer( Ch: Character );

  protected Printer is
    procedure Print_Two( Ch: in Character );
  end Printer;

  protected body Printer is
    procedure Print_Two( Ch: in Character ) is
    begin
      Put( Ch );
      delay 0.1;
      Put( Ch );
      New_Line;
    end Print_Two;
  end Printer;

  task body Char_Printer is
  begin
    for I in 1..10 loop
      Printer.Print_Two( Ch );
    end loop;
  end Char_Printer;

  Task1: Char_Printer( '*' );
  Task2: Char_Printer( '+' );

begin -- main "Tasks" routine
  null; -- Nothing to do here!
end Tasks;
```

Listing 4 – Synchronised tasks.

among a number of languages, and the Ada 95 specification includes scope for inline assembler (Listing 3). An interesting feature of GNAT is that it can directly use C++ objects, so if you really want to, you could attempt to use those huge C++ libraries you have with Ada code!

Ada includes tasking support as part of the language definition, so multi-threaded code is much more portable between systems, assuming that the facilities provided by the language are sufficient. (There are difficulties: the scheduler is not specified in details, and handling task priorities is not well defined). Listing 4 shows how Ada's protected types can be used to enforce mutual exclusion between tasks. The program has two separate tasks, each printing lines containing pairs of characters. Without any synchronisation, the output from the two tasks would be mixed together, resulting in lines like '+**' being printed. The protected type `Printer` prevents this, since it only allows its facilities to be used by one thread at a time, preventing each thread's output interfering with the other.

Perhaps surprisingly, GNAT's DOS variant supports Ada's tasking features (a rarity in the DOS world), but not as well as other environments – although there are two tasks in the example, the first one to start runs to completion before the other gets a look in (ie delay does not cause any rescheduling). The only way I have found so far to cause 'fairer' scheduling is to employ the heavyweight rendezvous method (which was the only synchronisation method available in Ada 83).

In C and C++, I make a lot of use of the `assert` macro, for such things as numeric range checking, but Ada provides many checks of this type for free (I value such defences against my own stupidity!). The run-time cost of these checks obviously causes concern, but Ada has a `Debug` pragma for turning on the more expensive checks, and GNAT permits them all to be disabled.

Thus far I have concentrated on major areas, and aspects where C++ and Ada differ, but what about other reaches of the two languages which are similar?

- Both have templates (called generics in Ada).
- Both have function and operator overloading. (Plus, in Ada, you get '/=' – not equals – when you define the equality operator.)
- Both have default subroutine parameters.
- Both have union types, though some effort is required to define an anonymous union in Ada.
- Both support arbitrary pointers, but Ada gives you considerably more control over what a pointer can legally access. The downside of this is that it is more difficult, though not impossible, to define complex static structures in Ada, like 'structures containing pointers to variable sized vectors of structures of...'. (I am afraid I have been guilty of using such things in writing parser control tables).
- Both support `goto`! One of the common uses I have for `goto` in C++ is to break out of nested loops. In Ada, loops can be named, permitting a more tidy exit, as in Listing 5.
- Both have exception handling.
- Both support function inlining, but Ada is able to detect more cases automatically, so it is less likely that the programmer will have to specify a function as inline.

A fairly significant facet of the language which I have deliberately omitted is the comparison of development environments. GNAT has the usual level of GNU toolset support (such as emacs and gdb) but this feels a bit primitive compared with more mainstream compilers' 'visual' IDEs. I have not examined other Ada compiler vendors' offerings, but I doubt if they are as slick as current mainstream C/C++/Java

tools on PCs. If you favour the old Turbo environment, there are a few fairly simple IDEs available for Ada under Windows/DOS.

The committee's decision

I hope I have shown that Ada is not the costly lumbering monster that many people consider it to be, and that C++ is really bigger than it looks. With the availability of GNAT, Ada is affordable to many, and the existence of high quality documentation and tutorials (mostly on the Internet) will help developers become familiar with the language.

I doubt if any single language is truly suitable for all kinds of development, so one important question is: what is Ada good for? Well, first I will answer the opposite question: I will not be using Ada for quick throwaway hacks – those I can do a lot more efficiently in C or Perl. I will not be using it for 'cutting edge' Windows development – there is a Windows 95/NT binding, but it is fairly basic and cannot handle recent subsystems such as DirectX (yet). Ada (or, to be more precise, GNAT) is not suitable for very resource limited systems – GNU compilers do tend to produce rather bulky code. However, I intend to apply it to just about everything else, though not necessarily on its own. I can always link in some C or assembler for the bits it cannot handle. An examination of the language reference annexes suggests that, as well as the type of programming I am interested in (mainly real-time systems), Ada will be able to address numeric processing (Fortran's forté), information processing (Cobol's domain), and portable distributed systems.

I think my programming benefits from the basic checking available, both at compile- and at run-time, and the extra portability of tasking will be useful. My initial concern when approaching Ada was about potential difficulty in expressing low-level designs – breaking through the language's strong typing to get code to do what I wanted. However, Ada 95 lets me do most of what I need. In conclusion, I think Ada is now a language worth adding to your toolset. ■

Gavin Smyth is a real-time software engineer and part-time Windows and Linux hacker.

C++ example.

```
for( x = 0; x < 10; x++ )
  for( y = 0; y < 10; y++ )
    if( done( x, y ) )
      goto abortLoop;
abortLoop:
```

Ada example.

```
Outer_Loop:
  for X in 0..9 loop
    for Y in 0..9 loop
      if( Done( X, Y )) then
        exit Outer_Loop;
      end if;
    end loop;
  end loop Outer_Loop;
```

Listing 5 – Breaking out of nested loops.

Useful links

The GNAT home page is to be found at *http://www.gnat.com* (the DOS version of GNAT is in the *ez2load* directory of the ftp mirrors site). ACT also has a European site, at *http://www.act-europe.fr*.
The DJGPP home page is at *http://www.delorie.com/djgpp/*.
Lovelace, an interactive Ada tutorial is provided at *http://www.adahome.com/Tutorials/Lovelace/lovelace.html*.
The Ada language reference manual and rationale may be found online at the Home of the Brave Ada Programmers (*http://www.adahome.com*) or in GNAT's ftp mirrors in the directories *rm9x-v5.95* and *rationale-ada95* respectively.
A valuable UK site is Ada Language UK at *http://www.adauk.org.uk*.

LEGACY SYSTEMS WORKBENCH

Time to solve the year 2000 problem

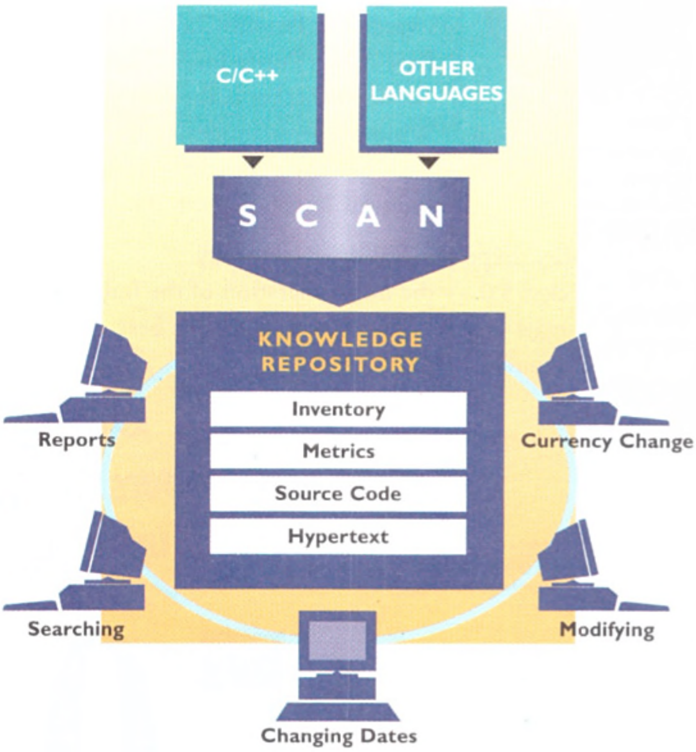
LEGACY SYSTEMS WORKBENCH (LSW) WILL CONVERT YOUR YEAR 2000 CRISIS INTO A MANAGEABLE MAINTENANCE PROJECT.

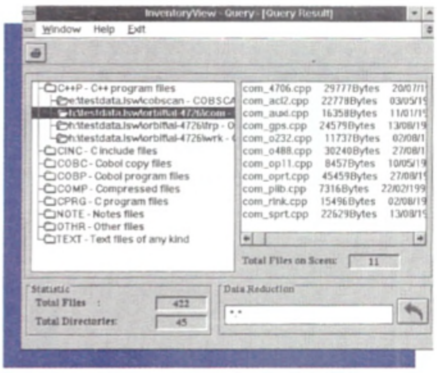
LSW is already widely used in the States by major companies, AT&T, Comsys, Amdahl, the US Ministry of Defence and Government Audit Office. This innovative product is revolutionising and standardising computer software maintenance.

How? LSW reads C/C++ and many other languages, FAST! What might in the past have taken you months can now be achieved in hours.

LSW scans source code and creates a total Knowledge Repository which incorporates a complete System Inventory, Metrics, Source Code and links between code objects, in Hyper-text and graphic representations - all in a client/server environment.

Applying this to your Y2K problem will enable you to make the best use of your resources as experienced programmers will be free to instigate date changes and test resultant changes.





Recognising the 'Millennium Bug' as a real problem is your first step. Applying LSW to your code will ensure the next steps are swift, successful and cost-effective.

Each passing day takes you closer to the Year 2000 Problem, LSW takes you closer to the solution.

IF YOU WOULD LIKE FURTHER INFORMATION, OR TO ATTEND ONE OF OUR BRIEFINGS PLEASE CONTACT JAN WALKER, MARKETING MANAGER.

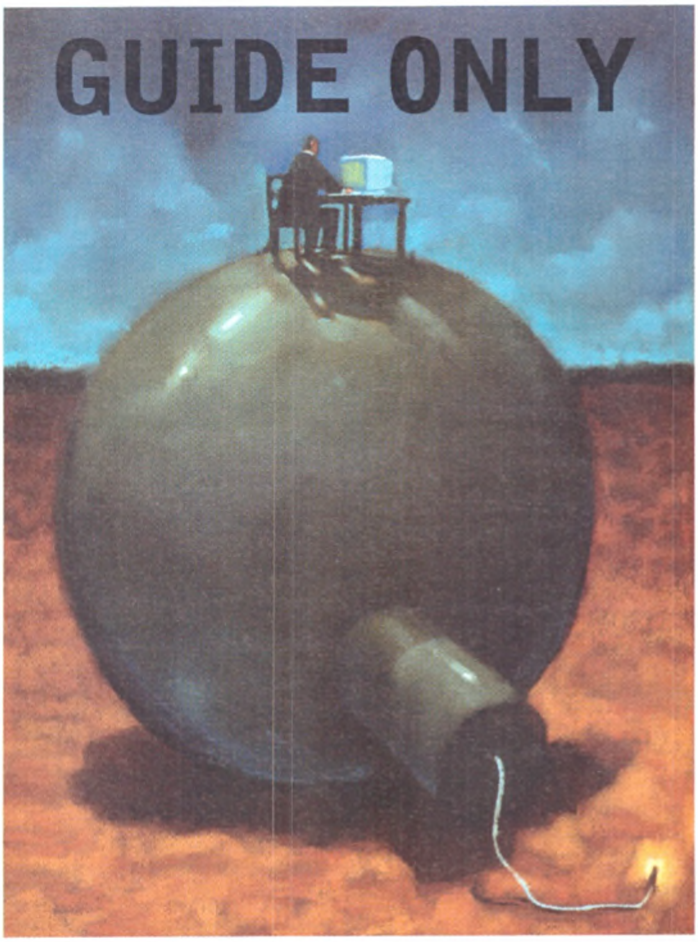
Quality Awareness UK Limited

50 Regent Street London W1R 6LP
Tel: +44(0)171 328 0300 Fax: +44(0)171 624 4060
Email: QAUK@QUALITYAWARE.COM

CIRCLE NO. 233

On to the year 2000

Despite the furore over the Millennium time-bomb, many people assume that the only losers will be huge mainframe applications written in Cobol. Peter Collinson examines some problems that might happen a little closer to home.



Jamel Akib

Loads of people are jumping up and down about the imminent death of our computer dependent world when we enter the last year of the millennium. Many systems are already experiencing serious problems, and absolute doom is predicted on the stroke of midnight on December 31st, 1999. This 'problem' is referred to as 'Y2K' on the Net, and there are several Web sites and forums dedicated to it. Actually, most of these sites seem to be full of theoretical scenarios rather than any real hard information. The Y2K problem looks to be something that is generating documents written in planning-and-marketing-speak and so there's often considerably more heat than light on the topic.

Nonetheless, the danger is real, and I have begun to wonder how safe my personal systems are. How about the non-programmable domestic and office systems that are lying about? Will my video tape recorder work? The answer is yes – and it even knows that 2000 is a leap year: if you're buying a VCR, then it's a good idea to check on its timing chips. My Samsung fax machine is a different issue. Ominously, you enter the year as a two digit number into its date setting routine, and when printing dates it doesn't print the full year. So I loaded '00', and was told that the day was Monday. Since the first day of 1900 was a Monday, I am assuming that it's got the wrong date.

That's some good news and some bad. What about my computers? Up to now, I've dismissed any possible problem as the domain of old business machines running Cobol programs written in the '70s. Many of those systems deal with two-digit dates, and so have problems with dates after the year 2000 being numerically less than dates in this century. Unix and DOS, on the other hand, deal with signed 32-bit clock ticks, so they should be OK – or are they?

DOS dating

Well, it turns out that all my PC machines have problems. When AT PCs were designed, they came equipped with a CMOS real-time clock that kept track of the date and time, so that you didn't have to enter them every time that you booted the system. The date is stored as three binary coded decimal numbers, each with two digits. There's one pair for the day, one pair for the month and the last pair store the last two digits of the year. Later on, storage of the top two digits of the year was added, allowing the system to store the complete day/month/year value. However, (presumably for backwards compatibility reasons) the century (ie the top two year digits) is stored in a different location and treated as a constant. It's not incremented when the year transitions from 99 to 00.

In practice, there is a wide variation in what will happen when the transition from 1999 to 2000 occurs. Most versions of DOS maintain their own time, and load this time from the machine's CMOS clock when the system is booted. DOS time is a 32-bit signed integer that stores the number of seconds from January 1st, 1980. On most current systems, if you leave the machine on as the date transitions from 1999 to 2000, then the date will be correct when you arrive for work after your New Year's Day hangover. However, the first reboot after the date transition will look at the CMOS and see a date of 1900. DOS will think 1900 is bogus and will adopt its default action when it gets a bogus date: it will set its date to that of its epoch in 1980. Setting the time by hand and a reboot will fix the problem.

Some more recent BIOSs will supply the system with a 'corrected' date, so the DOS time will be OK – but the CMOS settings will not. These CMOS addresses are publically available, and some applications certainly obtain their date information by accessing them directly, so these applications will fail after the transition

www.qbss.com

LOW PRICES VAST RANGE EXPERT SERVICE

Visual Basic

Addsoft *(Upgrades available)*	
Gantt VBX/OCX	£189
Resource Manager VBX/OCX	£189
Schedule VBX/OCX	£189
Apex *(Upgrades available)*	
TrueGrid Pro VBX	£99
True dBgrid OCX	£149
MyData Control	£149
Bennet-Tec	
Alltext Std/Pro	£115/265
TList OCX or VBX Std/Pro	£115/165
Crescent *(Upgrades available)*	
ClassAction	£99
DBpak	£185
Enquiry OCX	£299
FTP Internet Data	£75
Internet ToolPak	£135
NetPak Pro VBX/OCX	£129/139
PDQComm VBX/OCX	£95/139
PowerPak Pro VBX/OCX	£499/445
QuickPak Pro VBX/OCX	£135/159
RadBench	£85
Scope/VB	£75
VB Appframework	£185
VBNet NEW	£135
Desaware *(Upgrades available)*	
SpyWorks 4 Pro (OCX & VBX)	£185
StorageTools OCX	£100
Version Stamper VBX/OCX	£110
Farpoint	
ButtonMaker OCX	£70
Input Pro NEW	£99
Professional Toolbox	£239
Spread VBX/OCX	£210/210
Tab Pro VBX/OCX	£90
GreenTree	
ActiveX Suite	£375
DataList	£95
DataMask	£95
DataTree	£95
DataView	£95
XGrid	£95
ImageFX	
Fractal FX VBX+OCX	£150
FXPic VBX + OCX	£215
FXTools/VB Pro VBX + OCX	£285
Planet FX ActiveX control	£215
Vector FX OCX	£265
DFL	
LightLib Business STD/PRO	£189/339
LightLib Images STD/PRO	£189/339
LightLib MagicMenus STD/PRO	£69/99
LightLib Multimedia STD/PRO	£189/339
ProtoView	
Data Explorer	£135
DataTable OCX or DLL	£135
DataTable OCX & DLL & VBX	£265
InterAct Diagraming OCX or DLL	£275
InterAct Diagraming OCX & DLL	£345
WinX Component Library	£135
Sheridan	
3D Widgets	£75
Calendar Widgets	£90
ClassAssist	£169
Component Suite (Cal+Data+Des)	£235
Data Widgets	£90
Designer Widgets	£90
Sp_Assist	£415
VBAssist	£125
VideoSoft *(Upgrades available)*	
VS Data ActiveX NEW	£115
VSFlex VBX/OCX	£89/115
VS Reports ActiveX NEW	£115
VSVBX/VSOCX	£40/75
VSView OCX	£115
Visual Components *(Upgrades available)*	
Codebank for VB4	£99
DBComplete NEW	£Call
First Impression VBX/OCX	£115/175
Formula One VBX/OCX	£115/175
Geopoint 32 bit ActiveX GIS	£195
ImageStream	£115
SystemTools	£59
Visual Dev. Suite Deal VBX/OCX	£195/185
Visual Speller VBX/OCX	£115/140
VisualWriter VBX/OCX	£115/175
WebViewer	£175
Miscellaneous	
3D Graphics Tools 16/32	£119/139
CodeReview Standard/Pro	£149/299
Crystal Reports v 5 Pro Full/Upg	£289/149
ErgoPack for VB, VC++ and Delphi	£225
Erwin Desktop for VB	£395
FailSafe Standard/Pro	£149/299
ForeVB	£69
Helping Hand (Help for VB)	£99
HighEdit SDK	£275
Intelligent Paper	£125
LeadTools 7 OCX 16 bit or 32 bit	£299
List & Label forVB	£275
PowerBasic Developer Kit (PBDK)	£115
Rockwell RSAnimator	£159
Rockwell RSToolpak I/II	£149/189
List & Label forVB	£275
TX Text VBX/OCX/combo	£199/199/299
VB-Cert Text Exams	£99
VBCompress 4.0	£85
VB Language Manager 3.0	£135
VBtrv Btrieve from Classic Software	£185
Vision StoryBoard Pro	£135

MAY NEWS

Delphi 3 is out. And it's another winner. Delphi 3 demonstrates that once again, the engineers at Borland have paid attention to their developers' wish lists whilst keeping a close eye on the market. So loads of super-advanced Internet/intranet facilities incl. support for CGI applications and improved TCP/IP support. Addition of Web Bridge, an API for NSAPI/ISAPI. Total ActiveX support and also support for FoxPro and Access data formats. More than 30 new controls have been added to the Visual Component Library bringing the total up to well over 100. A new set of comprehensive database analysis tools. Smaller EXEs - you can now farm the VCL out to a centralised DLL. That's the guts of it. For more detail, check out Dave Jewell's excellent review in last month's *EXE*, and then place your order.

Microsoft Visual Studio. Wow. We like the look of this one too. All Microsoft's major development stuff in a box. **Visual Basic 5.0 Pro; Visual C++ 5.0 Pro; Visual J++; Visual FoxPro 5.0; Visual InterDev; Microsoft Developer Network Library**. That's the **Professional Edition** and you can upgrade to it from most Windows development systems for just £395. That's outstanding value. For £795 you can upgrade to the **Enterprise Edition** which has everything that the Pro does PLUS **Microsoft Transaction Server, Developer Edition; SQL Server 6.5, Developer Edition** and **Visual SourceSafe 5.0**

QBS Software News, Volume 7 Issue 1. The latest issue of our free catalogue for discerning software developers everywhere is out this month. If you don't receive QBS Software News automatically, call us now and we'll see you right.

Delphi

ABC for Delphi Pro VCL	£135
Apiary Dev's Suite for Netware	£365
Apiary NetBIOS Custom Control	£79
Apiary OCX Expert	£189
Apollo 3 Standard/Prof.	£129/199
Async Professional 2.0 (upgrade)	£135(£49)
BoundsChecker 5 for Delphi NT&95	£325
Component Create	£135
DBPower	£139
Delphi 3.0 Standard	£84
Delphi 3.0 Professional	£389
Delphi 3.0 Client/Server	£1389
Delphi 3.0 Professional Upgrade.	£239
Delphi 3.0 Client/Server Upgrade.	£1089
Direct Access	£189
Dr Bob's Experts v2	£49
Dynazip 16 bit/32 bit/combo	£175/210/289
Eagle CDK 16/32	£189/209
ImageLib 16 & 32 bit	£155
ImageLib@theEdge	£99
Infinity Report VCL	£69
Infinity Security Component	£69
Infinity MAPI Component	£69
InfoPower 2.0 VCL/+Source	£150/199
Innoview Multilanguage Std/Pro	£125/750
List & Labels for Delphi	£275
Memory Sleuth	£40
MK Query Builder (incl. source)	£149(299)
Piparti Reporting Std/Pro NEW	£175/350
Orpheus 2.0	£135
Quick Reports 2 with source	£79
Shoreline DialogPROS	£185
Shoreline VisualPROS	£115
Titan for Btrieve /+source	£295/525
TOLEAutomation Client	£39
TRPKCrypto Single User	£95

Internet

Borland IntraBuilder Std/Pro	£69/349
Borland IntraBuilder Client/Server	£1279
Catalyst SocketTools	£190
Dart PowerTCP Std/Speciality	£450/450
Distinct TCP/IP SDK 16 or 32 bit	£375
Distinct Internet Toolkit Visual Edition	£199
Distinct Runtime Ext./Ext.+ Winsock	£49/99
Mabry Internet Pack/(inc. source)	£99/(275)
Net-Install from 20/20 Software	£225
Sax Webster Control16/32	£105/105
Wise WebDeploy /Upgrade	£199/75

C/C++

Borland C++ v 5.0/Dev Suite	£225/315
Borland C++ Builder Learning/Std	£40/69
Borland C++ Builder Pro/CS	£399/1299
Borland C++ Builder Pro Upgrade	£249
Borland C++ Builder CS Upgrade	£1099
BoundsChecker Pro VC++ NT&95	£525
C-Vision from Gimpel	£179
CodeBase 6.0	£250
CodeSQL 6.0 5 user/Unlimited	£350/2420
Greenleaf Comm++	£205
Greenleaf Database Library	£195
High Edit SDK/PRO	£275/399
Leadtools7 WinPro 16/32 bit DLL	£605/765
MKS Toolkit	£285
Optima++ 1.5 Developer/Pro	£139/329
PCLint from Gimpel	£179
Object Master	£195
Stingray Objective Grid (16/32 bit)	£299
Stingray Objective Toolkit (16/32 bit)	£379
VBtrv for C++	£275
Watcom C/C++ no docs/with docs	£139/238

Microsoft

Office 97 Pro (Incl Access)	£449
Office 97 Dev. Upgrade from 97	£250
Visual Basic 5 Pro/Enterprise	£375/939
Visual Basic 5 Upgrade Pro/Ents	£195/549
Visual Basic 5 Comp. Upgrade Pro	£239
Visual C++ 5 Pro/Ents	£365/895
Visual C++ 5 Upgrade Pro/Ents	£239/545
Visual FoxPro 3 Std/Pro	£155/389
Visual FoxPro 5 Pro	£399
Visual FoxPro 5 Upgrade	£225
Visual J++	£79
Visual Studio Pro / Upgrade	£795/395
Visual Studio Enterprise /Upgrade	£1165/795

Databases

Borland *(Upgrades available)*	
InterBase 4.0 Workgroup Server	£790
InterBase 4.0 Additional single licence	£119
InterBase 4.0 Additional twenty licences	£1695
Paradox 7 Client/Server	£449
Visual dBASE Client/Server	£369
Visual dBASE Compiler	£229
Extended Systems *(Upgrades available)*	
Advantage Database Server	
CA VO/CA Clipper Interface Client	£75
ODBC Interface Client	£195
SDE 2.0 Interface Client	£215
Sixty day Server Evaluation Pack	£70
2 User Server Development Kit	£215
5 User Server Deployment Kit	£695
10 User Server Deployment Kit	£1145
100 User Server Deployment Kit	£3445
Microsoft *(Upgrades available)*	
Access 7 (or 2)	£265
Access Developer's Toolkit 2/7	£265/319
SQL Server NT 6.5 incl. 5 user lic.	£1035
SQL Server NT 6.5 incl. 10 user lic.	£1420
SQL Server NT 6.5 incl. 25 user lic.	£2845
SQL Server NT 6.5 incl. 50 user lic.	£5685
SQL Server NT 6.x Single Client Lic.	£120
SQL Server NT 6.x Twenty Client Lic.	£1950
SQL Server NT 6.x Workstation	£385
Powersoft/Sybase/Watcom	
PowerDesignor AppModeler	£695
PowerDesignor AppModeler Desktop	£195
Sybase SQL Anywhere Single User Svr	£190
Sybase SQL Anywhere 4 User	£395
Sybase SQL Anywhere 8 User	£650
Sybase SQL Anywhere 16 User	£1115
Sybase SQL Anywhere 32 User	£1975
Sybase SQL Anywhere Unlim.	£3300
Oracle	
Personal Oracle 7	£345
WorkGroup Server (5 users)	£1250
Developer or Designor 2000	£3350
Programmer 2000	£835
Power Objects Standard	£335
Power Objects Client Server	£1675

Notes

Domino Starter Pack Vs 4.5	£895
Domino Server Vs 4.5	£689
Desktop Client on CD/licence only	£85/65
Full Client on CD/licence only	£305/265
Object Prog. Interface (GlobeByte)	£695

PowerBuilder

Advanced Developers Toolkit	£279
Component Pack	£call
Desktop 5	£195
Enterprise	£3195
FUNCky for PowerBuilder	£159
PowerFrame App. Fr. Library	£305
PowerFrame Navigator Object	£99
PowerFrame Object Analyser	£99
PowerFrame TabFolder	£99
Powersoft Portfolio	£310

Clipper

Blinker Linker	£179
Clip-4-Win	£195
dGE Graphics	£189
FUNCky 2.5	£199
NetLib Network Library	£299
NovLib Library	£189
Scripton PostScript Library	£129
SilverClip SPCS Comms	£249
Summer '93 Code Optimiser	£159
Telepathy Clipper Comms	£169
T-BASE Graphics DOS or Win	£399

Java

Codebase 6 with Java Docs	£225
Jamba Professional	£189
Java Workshop	£79
JFactory for Win95/NT	£166
JMoney for Win95/NT	£89
JTools for Win95/NT	£89
JWidgets for Win95/NT	£89
Symantec Cafe	£119

FoxPro

Foxfire Developer's Edition	£225
FoxFix for DOS and WIN	£149
ReFox Decompiler	£295
SilverFox Comms (Win/ DOS)	£249
Xilights Editor Enhancement	£99
xCase for FoxPro & V.FoxPro	£295

Tools

Aegis v 3 (1 user)	£199
Arpeggio DTop/Developer	£395/515
Barcode Lib. DOS/WIN/95	£389/249/519
CBI Quick 16 bit/32 bit	£1535/2250
Cryptor Data Encryption	£299
dBest Barcodes for Windows	£345
DataDirect Dev Toolkit 2	£425
DataDirect ODBC Pack	£310
DemoQuick Sim. Plus 16/32 bit	£375/595
DemoSHIELD 16/32 bit	£245/349
Doc-To-Help 16 or 32 bit	£295
Doc Studio	£599
ED for Windows v 3.7	£145
ForeHelp v 2	£299
Graphics Server Developers Kit	£195
HelpSite Help to HTML	£225
Installshield 5 Full/Upgrade	£625/375
Installshield Express Pro	£215
Interactive Brochure/Catalog	£225/375
MKS Source Integrity	£450
MS SourceSafe 95 (platform)	£329
Novlib Network DLL	£189
MultiEdit Pro	£135
MultiEdit's Evolve	£109
PC-Install Win 16/32 bits	£295
PC-Install DOS&WIN super combo	£375
PVCS Version Manager	£420
R&R Report Writer DBF/SQL	£199/295
SoftICE	from £329
SoftSENTRY	£499
SOS Help! for Win Info Author	£185
Telephony Toolbox TAPI/TSAPI	£345/585
Track Record/ + per seat	£279/179
xBase++ for OS/2	£350
Wise Installation System ver 5.0	£149
SmartPATCH for Wise	£129
WebDeploy for Wise	£129

Please check our ftp site **http://ftp.qbss.com** or **BBS 0181 956 8011** for downloadable info, demos and evals of QBS Software products.

Call us today

QBS Software Limited

11 Barley Mow Passage, London W4 4PH
Ph: +44 181 956 8000, Fax: +44 181 956 8010
BBS: +44 181 956 8011, Email: orders@qbss.co.uk
Web: http://www.qbss.com, L/server : info@qbss.co.uk

QBS SOFTWARE

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

CIRCLE NO. 234

until the CMOS is corrected. So, most systems will need a reboot and a time reset after the transition to ensure that the CMOS settings are correct – we hope. What the applications will do is less certain.

The information on AT machines has been taken from Bob Stammers' Web page. He also supplies a small program, DOSCHK, that you can run on your machine to find out what will happen. It sets the clocks to December 31st, 1999 at 23:59:55, waits for 10 seconds and then tells you the result. All of my PC systems will need a reboot and a date reset after the New Year festivities are over.

The RighTime Company supplies a free TSR program that can be used to manage the rollover. It also has a detailed discussion of the problem on its pages (see the *Resources* box).

Unix time

Well, up till now I've felt smugly complacent about Unix systems, but I suspect that my confidence is slightly misplaced and there are more problems lurking on Unix systems than we all might think. However, these are probably more at the applications level rather in the kernel, and I'll guess that many of my worries about Unix applications may also apply to DOS and Windows applications. Incidentally, I've reported any problems that I have found to the appropriate people, so please don't flood them with bug reports saying essentially the same thing.

Obviously, the same PC hardware problems that are described above will affect Unix and Unix-like systems running on the identical hardware. I've given the source of BSD/OS kernel a cursory glance and it seems that it will need some work in the clock routines to make it work properly after the 1999/2000 transition. However, there's time for that: BSDI has made a new system release every year, and every year you seem to get another CD as part of the package. The code at which I was looking is the basis for all the freeware BSD Unix systems on PCs, and it's possible that all those systems will need work. I am currently unsure about Linux and SCO Unix.

Other hardware platforms have system clocks that are based on seconds, like (it seems) the hardware clock on my Sun system, so there should be no problem there. Like DOS, (or is DOS like Unix?) the system tends to use this clock on bootstrap to set its own internal time of day value, which is then incremented by the kernel. Time in Unix is nominally a 31-bit value with an epoch of January 1st, 1970, retrieved from the kernel counter with the `time()` system call.

Some systems will also deal in 'negative' time. On my Sun, for example, a time that has the top bit set is taken as a date before January 1st, 1970 – the routines allocate 0x8000000 as January 1st 1900 and run 'forward' from there. I wonder if this decision will prove embarrassing? Unix time runs out on January 19th 2038, at 03:14:07 – will Unix last that long? If it does, then I suspect that people will want to translate `time_t` from `long` to `unsigned long`, because it will be to complicated to change all the file systems where dates are stored.

The `time()` system call was present in the early Unix systems, and on the PDP-11 returned a `long`. At that time, very few other integer objects in C were 32 bits long. The original call was actually:

```
long ti;
ti = time();
```

The `time()` routine gained the extra (apparently redundant) parameter in Unix V7, presumably because it was hard to pass back 32-bit values as function results on some machines.

The early systems also had library routines that would take the kernel time and convert it to set of values that were easier to process. These are:

```
str = ctime(&ti);
```

that returns a string similar to the Unix `date` command. Unix V7 also had the familiar `localtime`, `gmtime` and `asctime` routines:

```
struct tm *tm;
long ti;
tm = localtime(&ti);
tm = gmtime(&ti);
str = asctime(tm);
```

The `tm` structure was passed on from V7 to all later Unixes, and is enshrined in the ANSI C standard, giving the programmer numeric values for the seconds, minutes, hour, day, month and year. There were also some useful derived values, like the day of the week, and the day in the year. It's easy to be critical in hindsight, but I've always believed that the structure should have also contained a pointer to the current time zone name. Instead, it has a flag that tells you whether daylight saving time is on or off. On Unix V7, this mapped back to a constant value in the kernel. Things have changed here.

The source of many Y2K problems is the year value that is returned in the `tm` structure. The value is returned as *(year - 1900)*, so many programmers have coded print statements like:

```
printf("%d/%d/19%2d\n",
  tm->tm_mday,
  tm->tm_mon+1,
  tm->tm_year);
```

which is OK now, but in 2000 will print:

```
1/1/19100
```

which is not a good idea. The correct coding is:

```
printf("%d/%d/%4d\n",
  tm->tm_mday,
  tm->tm_mon+1,
  tm->tm_year+1900);
```

Given the problem with the year from the `tm` structure, it's likely that there are many programs with problems that are embedded in their code. The `printf` format with `19%` is easy to spot. However, it's sometimes hard when looking at other people's code to determine whether a year value is absolute or relative to 1900. I've been looking at the date handling routines in `mh`, and they are somewhat muddled. My conclusion is that they work now, but will fail in 2000. One problem is that they assume the epoch value for dates in the `tm` structure is the same as the century. Also, they try to allow a routine to parse two-digit year values, loading the result into a `tw` structure similar to the standard `tm` structure. In effect, the `tw_year` value is heavily overloaded, and statements like:

```
printf("Year %d\n",
  tw->tw_year > 100 ? tw->tw_year : tw->tw_year+1900);
```

are present. Like the statements above, these work now, but will fail in 2000. The first step to get this code working is to decide on a single representation of the year value. The `tw_year` value should either be relative to 1900 or absolute. The value should be translated into the correct value when it is loaded, and converted into the appropriate representation when it is read.

Making a date

These days there are several 'new' date routines that make it easier for the programmer to deal with dates, but much of the `mh` code (and that of many other applications) was written before many of these routines were added to the programmer's palette by the ANSI C committee.

The `strftime` function permits formatted output from a `tm` structure. The routine was needed to allow international localisation of dates and times. I have many programs that contain vectors of month and day strings that can be completely rewritten in terms of the special formatting characters that `strftime` supports. Not only that, `strftime` makes the code much simpler too.

Perhaps a more significant ANSI C addition is `mktime`, the reverse translation of the `localtime` and `gmtime` routines. The routine takes a `tm` structure as input and generates a clock time as a result. Being able to easily create correct clock times from arbitrary date values simplifies date sorting routines. In addition, `mktime` will take 'out-of-spec' values for fields in the `tm` structure and normalise the structure to form a correctly specified date. For example, to find a date 30 days from now you can do the following

```
time_t ti, ti30;
struct tm tm;
time(&ti);
tm = *localtime(&ti);
tm->tm_mday += 30;
ti30 = mktime(&tm);
```

The `tm` structure will contain a corrected set of values for the date that is thirty days from now. Notice that I've not used the static `tm` structure buried in the `localtime` routine for this purpose. In addition, if `mktime` fails it will return a `-1` value as an error – real code should check this.

Many old utilities did not have these new routines and adopted other strategies to determine aspects of a computed date. I have some complicated code in my accounting package that does date computations that will be simplified considerably by using the `mktime` routine. For example, how many days have elapsed since I sent out an invoice and not been paid? Without `mktime` to generate correct clock times, the routine to compute this from day/month/year data is very complicated. I have usually resorted to going back a day at a time carefully computing the date of the previous day and counting the number of times that I do this. Now I can construct correct clock times for the dates with one function call and simply subtract the two values.

Abbreviating the year

If code is currently pulling in two digit year values from the user, then it's probably reasonable to assume that two digit years that are less than, say, 70 should have 2000 added to them, and values greater than that should have 1900 added to them. This seems arbitrary, but should work.

It's also interesting to speculate if people will shorten dates further between 2000 and 2009. Will you write:

```
1/1/0
1/1/5
```

or

```
1/1/00
1/1/05
```

or

```
1/1/2000
1/1/2005
```

I'll bet the first and the last will be used. So should any code that scans for abbreviated dates also permit single digit year values? I would think so.

Unix tends to make considerable use of abbreviated year values. There are many programs that use the date to create temporary (or logging) files in the familiar `yymmdd` or `yymmddhhmmss` format, usually to give each file a unique human-recognisable name. One of the reasons for using the year, the month, the day and so on is that the files that are created will be sorted into correctly ascending order by the `ls` command. Once Y2K strikes, then there is a conflict in these requirements. If the code is left untouched, it will very probably create files starting with the 100 value and this may be somewhat counter-intuitive.

The curtain goes down

I've reached the conclusion that there must be a small number of programs on your Unix system that may be broken and could need fixing before the end of 1999. Some of these programs will probably form an essential part of daily life. The set of broken programs could even include the `date` program itself, which takes a string of the form *yymmddhhmmss* from the user to set the kernel's date value.

The best way to tell whether code is fixed is to look at it. I've scanned the BSD/OS source tree as an experiment, and only found one possible problem that had been overlooked. Unfortunately, I don't have the luxury of source scanning on my Windows or Sun systems, which I find more worrying. On my Sun, I've spent a little time scanning the binaries on the system for the string `19%`. I've used a script like:

```
for n in /bin
do
  res=`strzings $n | grep '19%'`
  if [ "$res" != "" ]
  then
  echo $n
  echo $res
  fi
done
```

The key here is the `strings` command that Sun has on its system because it was originally a UCB application – it's not necessarily available on all Unix systems. Of course, looking for `19%` only points to the easy cases. It has yielded one program on my Solaris 2.5 system that is not year 2000 safe. I will inform Sun.

Many thanks to Bob Stammers for help with this article. ■

Resources

Bob Stammers' Web page and the DOSCHK program is available at *http://ourworld.compuserve.com/homepages/saphena/year2000.htm.* The program is free for personal use, but Bob wants a fee from commercial establishments.

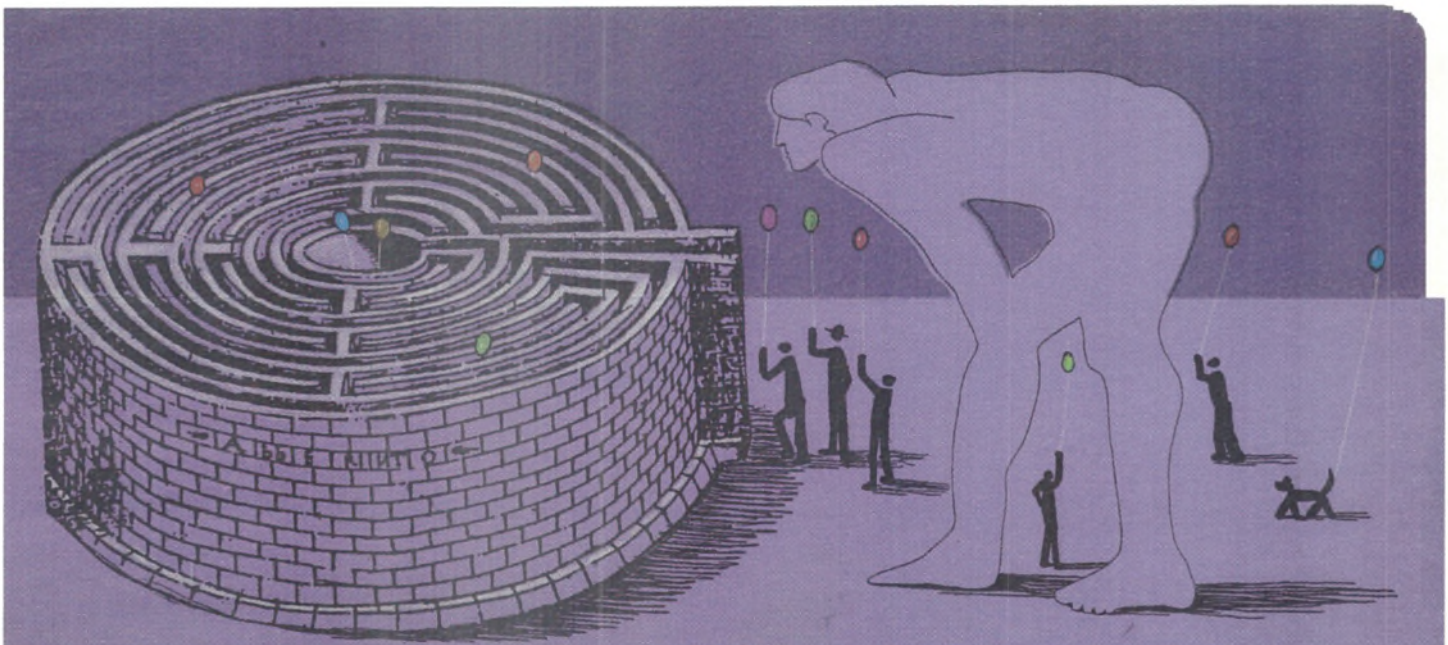
The RighTime Company Web pages also have a description of the problem: *http://www.rightime.com*, click on the *Year2000.txt* link. The company supplies a TSR for non-NT Windows machines, *Year2000.com* that can live in your machine providing a roll-over at the right moment.

http://web.idirect.com/~mbsprog is the Year 2000 Network. It comes with a zillion flashing red 'NEW' signs (when will people learn?). This site seems largely infoware and salesware. *http://www.year2000.com* is another Year 2000 site, there's perhaps more hard information here, and no flashing red signs.

Peter Collinson is a freelance consultant specialising in Unix. He can be reached electronically as pc@hillside.co.uk, by phone on 01227 761824 or on the Web at http://www.hillside.co.uk.

Sooner or later, you'll want to get this application

The mating game



Frazer Hudson @ Monster

Hatching a plan to escape from a maze, Mark Harman and Chris Kopec explore how different evolutionary strategies can affect the development of genetic algorithms.

Genetic algorithms are an approach to solving programming problems based on the laws of natural selection introduced by Charles Darwin in his famous book *Origin of species by means of natural selection*, in 1859. Darwin's theory of evolution has been astonishingly successful and all-embracing in its application in biology and many biologists would say that nothing in nature makes sense without it. More recently, programmers are finding that Darwin's theory, coupled with a modern understanding of population genetics, can be harnessed in the search for solutions to analytically hard problems (see *EXE*, April '97). The laws of how the fittest individuals survive in a biological system can be applied to 'breed' good solutions to computing problems, from a population of candidate solutions.

We'll look at a program which uses such a genetic algorithm to find the best path through a user-defined maze. The program performs no analysis of the maze, but instead simply breeds paths through it and allows them to compete, mate and mutate, until the best path is found. The completed program (along with an example maze) is available online from http://www.unl.ac.uk/~mark/students/kopec/exearticle.html. and from EXE's ftp site.

At the top level, most genetic algorithms follow the same steps:

1. Initialise a random population of individuals (candidate solutions to the problem).
2. Select two individuals to mate.
3. Produce a child by combining the two parents.
4. Select a place in the population to put the child (perhaps killing off a 'less fit' rival).
5. Perhaps mutate an individual.
6. Repeat steps 2 to 5 until a suitably 'fit' individual is found.

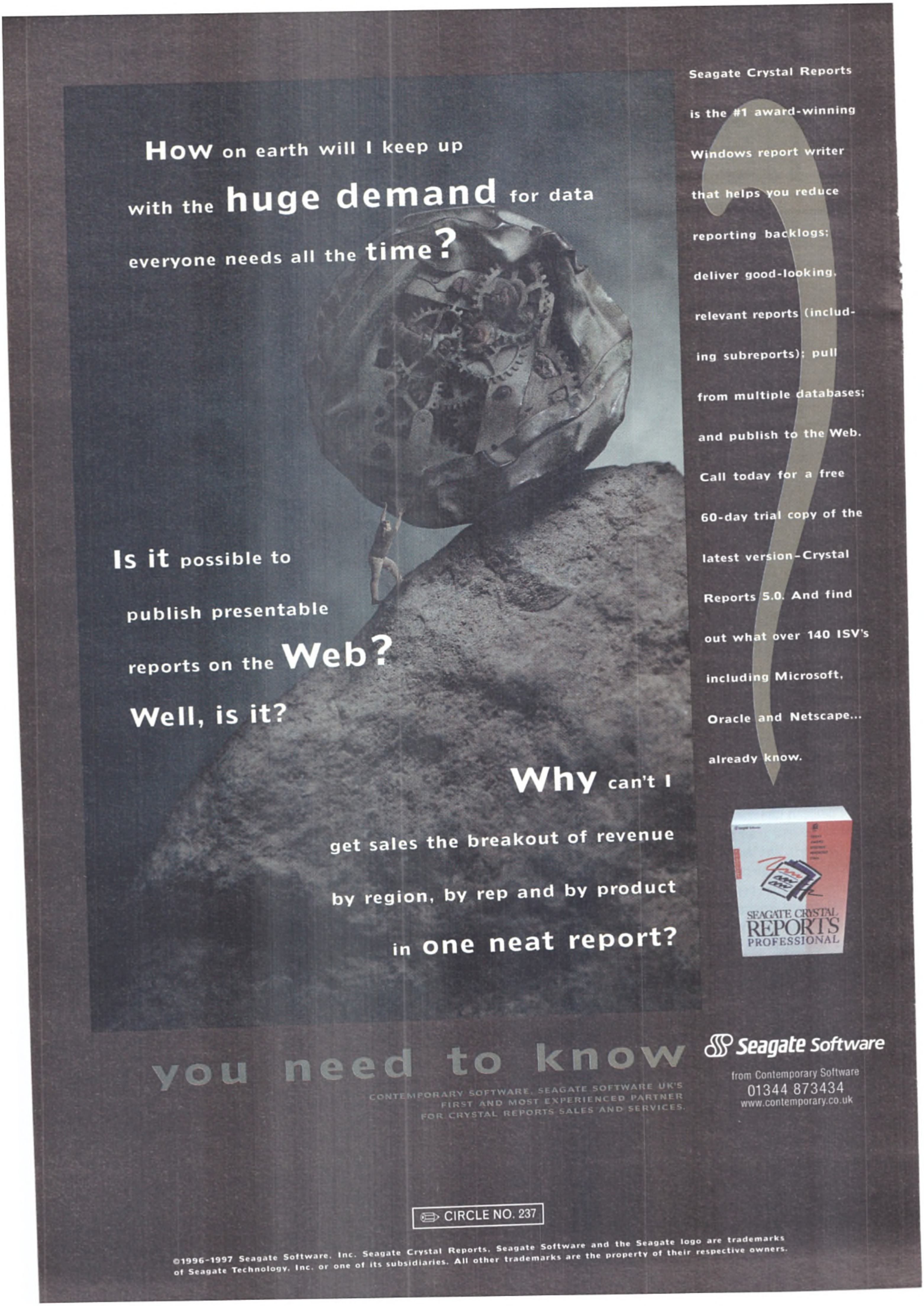
The idea is to create an environment for the candidate solutions in which adapting and surviving means becoming more like the kind of solution the algorithm is looking for. For any particular problem, certain ways of representing individuals and performing mating and mutation will be better than others. Solving problems with genetic algorithms is inherently an *experimental science*: improving an algorithm is a matter of adjusting these 'evolutionary' parameters, rather than analytically optimising a concrete solution. To facilitate this, the maze program has customisable options controlling the mating, mutation, and fitness-measurement methods, as well as the size of the population and the number of genes each individual has.

Escaping from the maze

The maze is represented by a 15 by 15 grid of characters, stored in a file called MAZE.$$$. Paths through the maze begin at the bottom left corner, and the exit (marked by the character e) can be placed anywhere within the 15 by 15 character grid. Walls are denoted by a w character and corridors by a dot. Figure 1 depicts an example maze.

The first step to take in defining a genetic algorithm is to decide how to represent the possible solutions to the problem as gene sequences. In biology, a gene codes for some property of the individual. In a genetic algorithm, a gene codes for some property of the solution we seek. (Indeed the correspondence between the biological world and the computational one is close enough that we often find ourselves mixing biological and computational terminology).

In our maze program, an individual will represent a path through the maze, and a gene will code for a simple step along that path. The problem is to decide what a gene should represent. Several natural choices suggest themselves; we could have four genes – north, south,


How on earth will I keep up
with the huge demand for data
everyone needs all the time?
Is it possible to
publish presentable
reports on the Web?
Well, is it?
Why can't I
get sales the breakout of revenue
by region, by rep and by product
in one neat report?
Seagate Crystal Reports is the #1 award-winning Windows report writer that helps you reduce reporting backlogs; deliver good-looking, relevant reports (including subreports); pull from multiple databases; and publish to the Web. Call today for a free 60-day trial copy of the latest version–Crystal Reports 5.0. And find out what over 140 ISV's including Microsoft, Oracle and Netscape... already know.
SEAGATE CRYSTAL
REPORTS
PROFESSIONAL
you need to know
CONTEMPORARY SOFTWARE, SEAGATE SOFTWARE UK'S
FIRST AND MOST EXPERIENCED PARTNER
FOR CRYSTAL REPORTS SALES AND SERVICES.
Seagate Software
from Contemporary Software
01344 873434
www.contemporary.co.uk
CIRCLE NO. 237
©1996–1997 Seagate Software, Inc. Seagate Crystal Reports, Seagate Software and the Seagate logo are trademarks of Seagate Technology, Inc. or one of its subsidiaries. All other trademarks are the property of their respective owners.

east and west, which we could code with a character (N, S, E or W). Each gene would then represent a step of 'one unit' in the direction indicated by the gene. For example, for the maze in Figure 1, one path from the bottom left to the exit (top right) is

NNNNEEEEEEEEEEEENNNWWWWWWWWWWNNNEEEEEENNNNEEEEEE.

(Notice how we think of this sequence of characters as both a path through the maze and as the genetic code of some virtual creature in an artificial world).

Another possibility, which might take up less space for many types of maze, would be to have a more sophisticated set of genes. We could encode the number of steps in a particular direction as well as the direction itself. Using this approach the individual above would be represented as N4E12N3W10N3E6N4E6.

Alternatively, we could use a gene to code for changes in direction. We could call this gene R and it could code for rotation through 90 degrees anticlockwise. Add to this a number *n* to code for *n* steps forward in the direction we are facing, and the representation is complete. Assuming that we start off facing north, the above would be represented by the genetic code: 4RRR12R3R10RRR2RRR6R6RRR6.

There will always be some speed-efficiency considerations to be taken into account, and the way we represent an individual will have a profound impact upon the effectiveness of the mating and mutation that takes place in our genetic algorithm, as we shall see later. We have chosen the second representation, because it is simple to code and quite efficient in terms of storage. These are important features, since one of the drawbacks of the genetic approach is its intensive processing in both space and time.

In order for our algorithm to move beyond the original randomised population of candidate solutions, we need to define how new individuals will be evolved. The simplest way we can produce a new solution is by mutating an existing one.

Mutating solutions

Mutation simply means selecting some individuals and altering their genetic code in some random way. It is an important process in all genetic systems: without it, no new 'ideas' would enter the system, and we would be stuck with the characteristics of the random starting population.

There are numerous ways to mutate an individual, depending on the type of solution we are trying to find. The most simple type of mutation is to pick an arbitrary bit in an individual's gene sequence and flip it. An alternative technique, that works very well with problems like our maze solver, is *gene shuffling*. Instead of introducing 'new' genes to an individual, we permute the order of the existing genes, or delete a gene and shuffle the remaining genes up to take its place. This is ideal for the maze problem, because we will frequently have redundant genes that need to be removed – for example, an instruction to go north when we are already at the top of the maze. By shuffling such a gene out, we remove it without disrupting any beneficial gene sequences that may precede or follow it.

The mutation process we come up with is of course closely related to how we choose to represent individuals. Suppose we have managed to breed a path which fails to reach the exit by only a single step. In this situation we are very close to a good solution, so we want very few mutations to be required to reach a better solution. Mutating N7 into N8 will take four bit flips, whereas mutating NNNNNNNE into NNNNNNNN will take at most two. So combining bit flipping with the first gene representation may not be as effective as with the second.

Turning to the third representation, we can see that we have three gene sub-sequences which are each really coding for a single gene, namely R, RR, and RRR. In this case, gene shuffling will tend to behave more like bit flipping, since it may disturb clusters of R genes.

Once we've chosen a technique for producing a mutation, the next step is to decide how many individuals will be mutated from one generation to the next, and to what extent each individual will be mutated. If we have too much mutation, then our genetic algorithm will become dominated by random changes and no species will have time to develop. In the most extreme cases, the algorithm will simply degenerate into a random search of the solution space – we will be trying to produce Hamlet with a few hundred monkeys and typewriters. On the other hand, if we set the mutation rate too low, our algorithm may take much longer than necessary to find an appropriate solution, and may miss out on better solutions simply because they are not close to the individuals in the original population.

The fitness function

The 'understanding' embodied in a genetic algorithm is entirely bound up in the fitness function, which takes an individual as a parameter and calculates a numerical measurement of how good a solution that individual represents. By themselves, the processes of mating and mutation take no account of the quality of the solutions they produce. By defining a fitness function, the algorithm can evaluate new solutions against the original population, and choose which solutions to preserve.

For our maze problem, it will in fact be easier to measure the *unfitness* of an individual, rather than its fitness, because it is easier to quantify the measure by imposing 'penalty scores' (or fines) on bad features of solutions. These penalties reduce a solution's reproduction potential, so that heavily penalised (ie less fit) solutions will not survive.

As a result of this strategy, you will see that the system first finds one or more routes through the maze, and then refines them to find the shortest sequence of steps an individual must take to reach the exit. We refer to these two stages as *exploration* and *refinement*.

```
mutate_individual = random(no_of_organisms);   /*select an individual to mutate*/
mutate_gene = random(no_of_genes-1);           /*select a gene to shuffle out*/
PointToCopy(individual,child);                 /*copy an individual, refer to it as a child*/
CopyGeneToMove(child);                         /*copies the gene to be shuffled out*/
ShuffleGenesAlong(child);                      /*moves the other genes along, filling the gap*/
PlaceGeneCopy(child);                          /*places the copied gene at the end of child*/
uf = assess_unfitness(child);                  /*calculates the unfitness of the child*/
if (uf<=unfitness[individual])                 /*if the child is fitter than the original*/
  {PlaceChild(child);                          /*updates old individual with new child*/
   unfitness[individual] = uf;}                /*updates the unfitness of the new individual*/
```

Listing 1 – Mutating a sequence of genes via gene shuffling.

```
.....W..........e
.W..W......WWWW.
.W..W......W....
.W..WWW..WW.....
.W.........W....
.W..WWWWWWW.....
.W.............WW
.W............W.
.WWWWWWWWWW..W.
..W...........W.
................
..WWWWWWWWWW....
..W..........WW.
..WWWWWWW.......
```

Figure 1 – A simple maze.

During the exploration stage, each solution is penalised a default of 100000 points for not finding the exit (if it had we would be in the refining stage). Solutions are rewarded for the number of locations they visit, encouraging them to spread throughout the maze, so that one of them will (eventually) find the exit.

When the exit has been found, we switch to the refining stage, and a different set of penalties: solutions that have not found the exit by this stage are fined a default of 30000 points. In addition, all solutions are charged 1 point for bumping into walls, and 50 points for every instruction (gene) that they use. All of these values are adjustable, so you can experiment with our preset solution.

We now have the means to introduce new solutions to our population, and assess their fitness. But the most important process in evolutionary systems (both natural and artificial) is, of course, mating.

Let's do it

Birds do it, bees do it, educated fleas might consider it and genetic algorithms do it (virtually) all the time. Mating merges two individuals into a new individual, which shares some of the characteristics of each of its parents. The idea is that fitter mergings (children) will replace less fit solutions, and then breed even fitter children. Or so it should be, if we get the conditions for mating right.

At the lowest level, mating two individuals involves combining some genetic information from each to produce the child. In our maze program, we could choose to perform mating by copying sub-sequences of each parent's genes to the child, switching between parents at random. Of course there is no reason to conform to rather dull and conventional forms of mating. Our solutions have no gender: they are *hermaphrodites*, able to mate with as many individuals as they please. In general, we have found that mating more than two parents does not appear to have any advantages over the basic model, but this may be a property of the kinds of problem which we have been solving with genetic algorithms (see 'The electric ant', *EXE*, December 1996).

Whichever mating strategy we decide to adopt, we will have to combine it with a method of child placement. The question is: 'which individual in the population should a child replace?' The strategy we adopt for child placement will have a critical impact upon the *diversity* of the population we breed, and it is important to allow more than one species of solution to evolve.

Diversity of species

For our purposes, a species is a set of individuals who each share a similar genetic code. The dividing line between different species might seem a little fuzzy in principle, but in practice things are usually clearer. This is because the circumstances will make certain individuals noticeably more fit than others, with the inevitable consequence that species will cluster around these 'better' solutions.

It is important to encourage species diversity, since it often happens that a species which is evolving towards the very best overall solution takes longer to arrive at the 'global maximum' than other species which are initially more successful but destined to become stuck in an evolutionary rut. In our example, we may evolve a route to the exit early on in the development of the population, with the consequence that a species will grow up around it. We should not allow this species to completely take over the population, as there may be another route to the exit which is shorter, but which the population has yet to discover. What

we have to do is to allow less fit species to survive long enough to achieve their full potential, while tilting the odds in favour of fitter individuals within each species, thereby ensuring that the species moves forward towards ever greater fitness.

A lot of research work has been undertaken to address the problem of promoting species diversity (both in the natural world and the computer laboratory). Fortunately for our maze program, we can maintain diversity by choosing a replacement strategy which ensures that a new individual replaces a similar one in the population as a whole.

As we explained last month, a simple way of achieving this is to make a child replace a less fit parent. This works because parents are guaranteed (by the mating process) to be genetically similar to their children, and are therefore of the same species.

In our maze program you will see species emerging when there are two or more different paths around some obstacle: for each path, there will be an optimal route around which a species will evolve. This is particularly evident when you define mazes with more than one route to the exit.

Generic genetic algorithms

The principles of the genetic approach to problem solving and optimisation have been extended to address a wide variety of applications, all sharing some form of mating, mutation and fitness assessment. The similarities between different applications are in fact so great that it might lead us to suppose that we could design a generic system for use with many different applications. Using such a generic genetic algorithm, the programmer need only be interested in the issues of data representation, mating, mutation and fitness assessment, leaving a generic genetic engine to take care of the underlying search for solutions. The UNL computing research group is working on just such an engine, with a range of handles for altering the representation and mutation of solutions, the rate and type of reproduction, and the strategy for placing children into a population.

Currently, such systems exist in only the most primitive form, but our ultimate hope is that, in the future, we will not need to tell a computer how to achieve a certain result, but instead simply tell it what properties we want the result to have. Perhaps the next great leap in the evolution of computing will consist of neither hardware nor software, but geneware – evolved solutions that we may not even understand, but nonetheless are perfectly suited to our problems. ■

```
int fitness(individual i) =
if (exploration_mode)   return 100000 - locations_visited(i);
else return !(found_exit(i))*30000 + bumps(i) + length(i)*50;
```

Listing 2 – A penalty-imposing fitness function.

The maze program and a sample maze are available for downlaod from http://www.unl.ac.uk/~mark/students/kopec/exearticle.html, and from EXE's ftp and site.

Mark Harman is director of research at the School of Computing, University of North London. He is currently working with Chris Kopec to see if genetic algorithms can be applied to the problem of program slicing (See EXE, October 1996), and to build the generic genetic engine described in this article. You can email him at m.harman@unl.ac.uk.

Chris Kopec is a lecturer in computing and mathematics. He has recently completed an MSc in Computing at the University of North London. His thesis, 'Optimisation: a generic genetic approach' is available on-line at http://www.unl.ac.uk/~mark/students/kopec/msc.html. He can be contacted at 100556.2714@compuserve.com

NEW!

ROBOHELP 4.0

Help Authoring Tools

ROBOHELP 4.0 is the only Help Authoring Tool for Windows 95, Windows 3.1 and Windows NT which allows Help Authors to simultaneously create Windows Help, HTML Based Help, Professional Printed Documentation, and Intranet/Internet Web sites from a single source.

* You can automatically generate all the HTML files needed for an Intranet/Internet Web site in seconds.
* Instantly create a user manual or any other report complete with Table of Contents, an Index, and Automatic Page Numbering.
* With free technical support and our no risk, 30 days money back guarantee there's no reason not to get WinHelp Office 4.0 and ROBOHELP 4.0 today.

ROBOHELP 4.0 £295

WinHelp Office 4.0 (inc. ROBOHELP 4.0) £425

ForeHelp 2.1 & 2.95 — Doc-to-Help 1.7 & 2.0

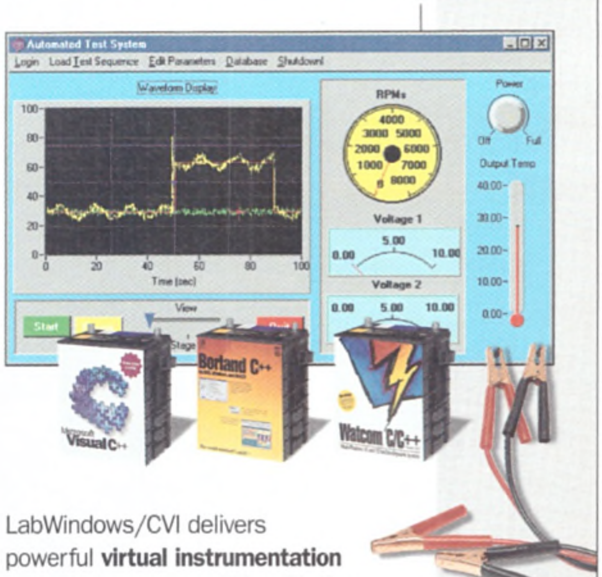
£295

£295

Oxford Computer Consultants
Littlegate House
St Ebbes
Oxford OX1 1PS
Tel: 01865 793 077
Fax: 01865 793 124
E-mail: Sales@oxfordcc.co.uk
Web: http://www.oxfordcc.co.uk

CIRCLE NO. 239

Supercharge Your C/C++ Development



LabWindows/CVI delivers powerful **virtual instrumentation** tools that are compatible with the leading 32-bit Windows C/C++ compilers. With its easy-to-use interactive development environment, you can:

- Quickly design virtual instrument front panels
- Perform extensive waveform analysis, FFTs, filters, and statistics
- Choose from more than 600 GPIB, VXI, and serial instrument drivers
- Acquire data with plug-in DAQ boards

Call today and supercharge your next test system with the new LabWindows/CVI virtual instrumentation tools for C/C++!

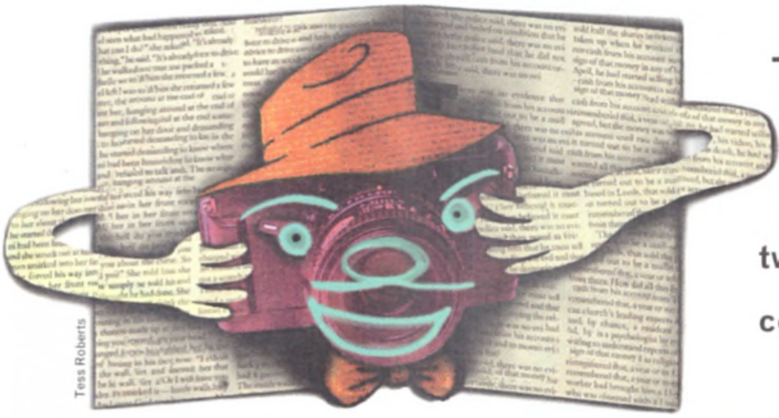
Call for your FREE evaluation software today.
01635 523545

NATIONAL INSTRUMENTS
The Software is the Instrument

National Instruments Corporation (U.K.) Ltd.
21 Kingfisher Court • Hambridge Road
Newbury, Berkshire • RG14 5SJ
Tel: 01635 523545 • Fax: 01635 523154
info.uk@natinst.com • www.natinst.com/uk

CIRCLE NO. 240

The private life of the Delphi component library



Tess Roberts

The Delphi IDE and component library may seem like an inseparable pair, but behind closed doors they are more like two rallying tennis players than wedded companions, as Dave Jewell explains.

When Delphi first appeared, a number of developers (myself included) spent some time musing about the possibility of putting the VCL library – components and all – into a DLL. The advantages of this were clear – it would make compiled Delphi applications very much smaller, and mean that a group of different applications could share the same component DLL. I knew that in principle it must be possible to do this, because within five minutes of getting my first Delphi 1 beta, I'd discovered that the component library, COMPLIB.DCL was nothing more than an ordinary Windows DLL. Every time you add or remove components from the library, it gets rebuilt by the IDE and renamed with a DCL extension. Whenever you place a component onto a form at design time, you're actually using a real live VCL component – which just happens to live inside a DLL. I reasoned that if the IDE could work this magic, it must be possible for an ordinary application to do the same. After all, there's nothing fundamentally special about the Delphi IDE – it's an ordinary Delphi program which happens to know how to 'drive' COMPLIB DLL.

In the end, I gave up on my quest to get the VCL library into a DLL because of the large amount of work involved. In retrospect, I'm glad that I did, because Delphi 3 now sports this capability for free through 'packages'. Nevertheless, I remained fascinated by the implementation details of the Delphi component library and this article is intended to bring together much of what I discovered on my travels.

Introducing Grandpa TInterface

The Delphi 1 and 2 IDE's both communicate with the component library (and with experts, version control systems and the like) using a number of documented and undocumented interfaces. What all these interfaces have in common is that they all inherit from `TInterface`. So what's special about the `TInterface` class? To answer that question, we need to take a look at how this interface is defined. The following class declaration is taken from VIRTINTF.PAS, part of the VCL source code.

```
TInterface = class
public
  procedure Free;
  procedure Release; virtual; export;
  function GetVersion: Integer; virtual; export;
end;
```

Doesn't look too exciting, does it? This interface simply defines three methods, `Free`, `Release` and `GetVersion`. Notice that the `Release` and `GetVersion` methods are defined with the `export` specifier, meaning that they can safely be used across DLL boundaries (the compiler automatically generates prologue/epilogue code to ensure that the DS register is set correctly when calling an exported routine). In addition, the `TInterface` class provides its own version of the `Free` method, which is called in preference to the usual `TObject.Destroy` routine.

Things get a bit clearer when we see how `TInterface` is used. Below, you can see the definition for the `TIStream` class from VIRTINTF.PAS. (By convention, many Borland classes which inherit from `TInterface` begin with 'TI' rather than just 'T', to indicate that they are interface classes and 'DLL-safe'). The `TIStream` class serves as a virtual, abstract interface to some sort of physical byte stream.

```
TIStream = class(TInterface)
public
  function Read(var Buffer; Count: Longint):
        Longint; virtual; export; abstract;
  function Write(const Buffer; Count: Longint):
        Longint; virtual; export; abstract;
  function Seek(Offset: Longint; Origin: Word):
        Longint; virtual; export; abstract;
  function GetModifyTime: Longint; virtual; export;
        abstract;
end;
```

If you examine each of the four defined methods, you'll see that they all have the attributes `virtual`, `export` and `abstract`. Most Delphi programmers will be familiar with the `virtual` specifier, and I've already explained what `export` does, but what about `abstract`? Well, it tells the compiler to point the corresponding VMT function pointer at an internal library routine called `Abstract`. It turns out that this routine displays a run-time error message and terminates the application!

The point, of course, is that you'd never want to instantiate an object of a class which defines abstract methods. Rather, the `TIStream` class is a sort of template which defines the desired interface, but leaves the implementation to some derived class. This idea of an abstract interface definition will be familiar to C++ and Java developers, and Delphi 3 even overrides the interface keyword so you can use it to work with COM objects.

There is an important point to note: once a 'virtual export abstract' method has been declared, any subsequent implementation of that method in a derived class will automatically inherit the export attribute, thus making it suitable to use as an interface method across a DLL interface.

Fun for all the family...

Having met the grandparent(s), it's now time to introduce ourselves the rest of the family. Firstly, let's look at the documented `TInterface` classes. The ISTREAMS.PAS file contains two stream classes that inherit from `TIStream`, `TIMemoryStream` and `TIFileStream`, which provide memory-based and file based streaming facilities respectively. The Delphi IDE itself makes extensive use of the `TIStream` class when communicating with the component library.

The file VCSINTF.PAS defines another member of the family, `TIVCSClient`. This class encapsulates all the functionality needed by the version control system. In order to get a reference to a VCS client object (remember, in Delphi an object reference is just a far pointer to the object), the Delphi IDE calls a known initialisation routine in the VCS DLL and returns the client object reference. You'll see that a very similar approach is used when communicating with the component library.

Delphi's add-on experts are handled by the `TIExpert` class defined in EXPTINTF.PAS. As with the VCS sub-system, expert DLLs are initialised by calling a known entry point in the DLL which returns a reference to an expert object.

This raises an interesting question: if these returned object references are what the IDE uses to 'pull the DLL's strings', how does the DLL code get to pull the IDE's strings as and when needed? For example, a VCS DLL might want to determine which files are associated with a given project, or save it. In order to achieve this, the IDE passes an object reference to the DLL's initialisation routine. This is a reference to an object of type `TIToolServices`, defined in TOOLINTF.PAS, and it's what the IDE provides to allow add-on tools to make use of its services. (You need to be clear about this. `TIToolServices` is all about providing IDE services to tool DLLs – it has nothing directly to do with the services provided by the tools themselves).

You can see from this that you've got a sort of symbiotic two-way contract: the IDE passes it's own control interface to the add-on DLL, and it gets the DLL's control interface back in return.

The undocumented bits...TAppBuilder and TLibrary

So much for the documented stuff. Now let's take a look at the bits you might not have come across before. As with the add-on DLLs, the Delphi IDE establishes a two-way contract with COMPLIB.DCL. Delphi 1.0 calls a special DLL routine called COMPLIB0027, passing an object reference to it, and expecting another one in return.

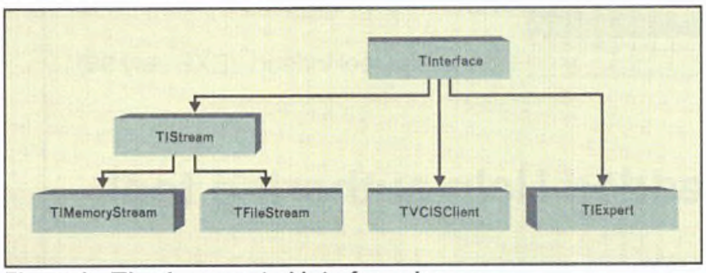


Figure 1 – The documented interface classes.

Note: Maybe you're wondering why Borland chose all these weird and wonderful DLL initialisation names such as COMPLIB0027, INITEXPERT0012, and INITVCS0011. The reason is simple – the numeric part of the function name was originally used to encode the number of VMT entries in the object returned from the function call. As the Borland development team worked in parallel on the IDE and on subsidiary DLL's, they changed the function name each time a new method was added to the returned class type. In this way, they implemented a crude form of version checking which helped ensure that the IDE and those other DLLs had similar views about the objects involved!

When it calls COMPLIB0027, the IDE passes an object of type `TIAppBuilder` and gets back a reference to an object of type `TLibrary`. This class inherits from another class called `TILibrary` which (surprise, surprise!) inherits from `TInterface`. Thus, as with the documented add-on DLLs, the IDE gets a `TLibrary` interface with which it can manipulate the library and the library gets a `TIAppBuilder` interface which it can use to interact with the IDE.

Listing 1 shows Delphi 1.0's `TILibrary` interface, which acts as the foundation stone for `TLibrary`. Understandably, the actual `TLibrary` implementation adds a number of private internal methods, but these can't be accessed from the Delphi IDE because they're not present in the `TILibrary` interface. (Incidentally, for the sake of brevity, I haven't included the `virtual, export, abstract` specifiers in Listing 1, but bear in mind that they are there).

The `TILibrary` interface introduces a number of new data types with which you may not be familiar with. For example, `TIModule` is a DLL-safe representation of a module, `TLibForm` represents a design-time form, and so on. Much of the time, the component library acts as a 'server', with the IDE as its 'client'. When you're sat in front of the IDE and you're adding components to a form, it's natural to think that both the form and its components are maintained by the IDE. In fact this isn't so; when you create a form in your project, it's the component library which actually owns the form, and adds components to it. The whole business of manipulating a design-time grid and its components is performed within the COMPLIB.DLL.

Borrowing from the library

You might be wondering where all this extra functionality comes from. If you look at your Delphi directory layout, you'll find that there's a directory called LIB. This contains not only compiled versions of the standard VCL units, but also DCU files for a number of proprietary units such as LIBINTF, LIBMAIN, and so on. Whenever the component library gets rebuilt, these files are included into the DLL along with whatever components might be installed. It's these mystery units which provide the design-time functionality I've been discussing here.

So, let's take a quick peek at the `TILibrary` interface. I won't try to describe what each call does; rather, we'll just look at some of the more interesting highlights. As I'm sure you'll appreciate, when you install components into the library, it's possible to implement more than one component within the same Pascal source file. From the viewpoint of the component library, each installed source file is referred to as a *module*. When you call `GetModuleCount`, the return value indicates the number of modules (units) currently installed into the component library. If you're familiar with the documented `TIToolServices` class, you'll know that it too has a method called `GetModuleCount`, which is implemented by calling the `GetModuleCount` routine in `CompLib`, the IDE's active `TLibrary` object. In just the same way, the `GetModuleName` method in `TIToolServices` ends up calling the method of the same name in the `TLibrary` class. This returns the name of a given module, given it's index number.

```
type TILibrary = class (TInterface)
    function CreateForm (AModule: TIModule; const AFileName:
String;
    Existing: Boolean; const AFormName: String): TLibForm;
    function CreateFromStream (AModule: TIModule; const
             AFileName: String;
               IStream: TIStream): TLibForm;
    procedure DesignerOptionsChanged;
    procedure EditAction (Action: TEditAction);
    funtion FindFile (const FileName: String): TLibForm;
    function GetActiveForm: TLibForm;
    procedure GetClassUnits (const ClassName: String;
                    Proc: TGetStrProc);
    function GetCompClass (Index: Integer): TCompClass;
    function GetCompClassCount: Integer;
    procedure GetCompHierarchy (Proc: TGetStrProc);
    function GetComponentBitmap: HBitmap;
    function GetEditState: TEditState;
    function GetModuleCount: Integer;
    function GetModuleName (Index: Integer): String;
    procedure GetProperties (PropKind: TPropKind;
                    Proc: TGetPropProc);
    function GetSelectionName: String;
    function GetSelectionType: String;
    procedure HideWindows;
    procedure IsDesignMsg (var Msg: TMsg;
                    var Handled: Boolean);
    function IsTopmostForm (Wnd: hWnd): Boolean;
    procedure ModalEditDone (ReturnWindow: TWinControl);
    procedure RaiseException (const Message: String);
    procedure ShowWindows;
    procedure SetLockState (State: Boolean);
    procedure SetProjectName (Name: String);
```

Listing 1 – The TLibrary interface from Delphi 1.0.

In order to get access to the actual components themselves, you use the `GetCompClassCount` and `GetCompClass` methods. The first returns the total number of installed components in the library, and the latter returns an instance of the undocumented interface type `TCompClass` for a specific component. This instance shouldn't be confused with an instance of the component itself. Rather, it's an encapsulation of component-related information which is used by the IDE and the component library to manipulate the component. One of the methods of `TCompClass` is `GetModuleIndex`, which can tell us which host unit contains the implementation of the component in question. Another method gives us the default palette page for this component and a third, `GetBitmapPos`, is used to index into a communal bitmap which is maintained by the component library. This bitmap can be accessed through the `GetComponentBitmap` method. You might be forgiven for thinking that this routine returns a small bitmap – the bitmap used by Delphi to identify a particular component on the Component Palette. In fact, it returns a rather large bitmap which contains the component images for *all* the currently installed components. The IDE uses the aforementioned `GetBitmapPos` method to determine the position of a particular component's image within this bitmap.

The `HideWindows` and `ShowWindows` routines are principally used when the application under development is executed, or when it stops executing. In order to reduce screen clutter, the Delphi IDE hides the Object Inspector, the Alignment palette and the Component List window (if visible) and then calls `CompLib`'s `HideWindows` method to hide any design-time forms that are visible. The process is reversed when an application stops.

As I mentioned earlier, the design-time forms are owned by the component library and not by the IDE. Consequently, the IDE has to keep track of the active form through the `GetActiveForm` method.

The interactions between the IDE and the component library are sometimes reminiscent of a lengthy volley at Wimbledon! For example, imagine that you right-click on a form (owned by the library). The library code will pop-up a speed menu for the form. This menu has been created by (and is managed by) the library, but when you click on the Creation Order item (for example), the library code calls a method of `TIAppBuilder` (specifically `ExecDesignDialog`), passing it a verb which tells it which design dialog to display. To cut a long story short, this results in the IDE creating a new `TForm` (of type `TCreationOrderDlg`), but not before it has first called the library's `GetActiveForm` to verify that there's actually an active form out there.

This ping-pong behaviour is carried to extremes in the case of run-time exceptions. If an exception occurs while the IDE is executing a `TIAppBuilder` method call from the library, then it releases the exception locally (ie within the IDE) and calls another method of `TLibrary`, `RaiseException`, which causes the component library to raise the exception again, back where it belongs in the component library. The same situation pertains if the library is executing a `TLibrary` method on behalf of the IDE.

Returning to Listing 1, the `GetSelectionName` and `GetSelectionType` methods are extensively used by the Object Inspector. As you make selections on the design-time form, the Object Inspector updates to show the name and type of the selected item(s), using the two aforementioned methods. In the same way, the `GetProperties` method is used to return a list of properties relating to the currently selected methods, which is used to fill the Object Inspector window. Because the number of properties associated with a given Delphi control is arbitrary, this method uses a call-back procedure which is called once for each property. Internally, the component library code uses the `GetComponentProperties` routine (see the DSGNINTF.PAS VCL source file) to enumerate the available properties and invoke the call-back routine for each encountered property.

Through the keyhole

There's a lot more that could be said about the interaction between the IDE and the component library, but by now you should be getting the idea. My aim in writing this article was to give you a feel for the intriguing double-act which is continually going on between these two major components of the Delphi environment. At the same time, I wanted to give you a deeper appreciation for the sort of complex systems that can be built using `TInterface` as a basis. In many ways, Delphi's `TInterface` mechanism is much superior to traditional, C/C++ based DLL calls. If you include the abstract interface class definition unit in both the application and the DLL, then you get automatic type-safe linking without the need for name mangling. Also, because of the inherently object-oriented approach, the DLL only ever needs to export a single, named initialisation routine irrespective of the degree of complexity between the calling application and the DLL. Do be careful, though, to follow Borland's example and ensure that run-time exceptions are correctly 'fielded' in the appropriate module. ■

Dave Jewell is a freelance consultant, programmer and technical author specialising in low-level systems programming, development systems and compiler design. He is the author of 'Instant Delphi' published by Wrox Press. You can contact Dave as djewell@cix.compulink.co.uk.

System Architect Brings Your Company to the Future of Application Development, without Abandoning Its Past.

System Architect the Price/Performance Leader... and Your Gateway to O-O.

It single-handedly solves your company's traditional application development challenges quickly and accurately and makes the transition to Object-Oriented smooth and easy.

Using System Architect's unique Shared Repository, you can:

- use all of today's most popular modeling techniques
- share and re-use data without duplication of effort
- quickly integrate object-oriented analysis into your development projects

System Architect can be easily modified and extended to support your unique modeling and code generation requirements. System Architect deftly supports your projects, taking you to a new plateau of increased productivity and cost-effectiveness.

With System Architect, you eliminate the need for multiple products to accomplish your tasks. It's the ***off-the-shelf solution*** you can customize to fit your precise requirements.

Just Beginning Your Development Project?

SA/Process Manager sketches the ultimate blueprint. It walks you through each stage of application development — from planning to implementation. SA/Process Manager guides you through structured analysis and design, data modeling, and object-oriented projects with clear, step-by-step advice on what to do, how to get it done, and when to do it. Customize SA/Process Manager to suit your needs, and tailor it to your unique development scenario.

Get Your Hands on the Best Family of Enterprise Development Tools Available Today.

Since 1986, Popkin Software has engineered high performance, low-cost software tools for the public and private sectors. Our award-winning software supports today's most popular methodologies.

Popkin Software supports your enterprise development needs by employing all major BPR, structured analysis and design, data modeling, and object-oriented methods in one extendable multi-user repository. And we provide comprehensive training to ensure your team builds effective, lasting solutions ***right out of the box***.

Call Popkin Software today and receive our free information pack. For more information, or to place an order, call 01926 450858. Visit our site on the World Wide Web at http://www.popkin.com.

POPKIN SOFTWARE & SYSTEMS, LTD.

Real Tools For The Real World.

Now with Internet Model Generation.

CIRCLE NO. 244

Bowling a googly

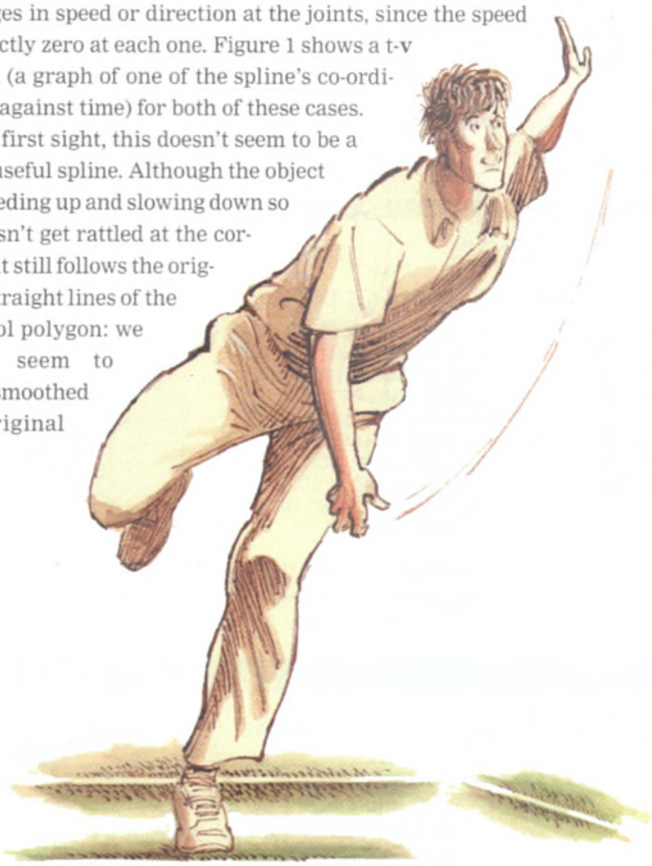
In the second part of his series, Jules May introduces kinks with Hermite splines, points out the most user friendly spline of all, the Bézier, and redefines time to render spline surfaces.

Last month we investigated splines in general terms, and looked at a specific type of spline, with Aitken's algorithm. Aitken's is good for smoothing a set of points 'automatically', or in other words with little or no user interaction. In many cases, though, we want to provide as much user interaction as possible. When planning car bodies, for example, the designer may want to apply very small tweaks to a shape without having to change the basic shape he's already made. A font designer, on the other hand, will want some way to mix sharp corners with smooth curves. For both of these applications, Aitken's will not be enough, and we need a different kind of spline.

Hermite splines

Imagine we have a set of control points (each at a certain position and point in time, as described last month), joined in sequence by straight lines to form a *control polygon*. If an object travels along the control polygon at a constant rate, it will make very sudden changes in speed and direction every time it moves from one segment to the next. Instead, it could accelerate along an edge and brake to a halt as it reaches the end of a segment, then turn to line up on the new segment and continue in the same way. This way, there will be no sudden changes in speed or direction at the joints, since the speed is exactly zero at each one. Figure 1 shows a t-v graph (a graph of one of the spline's co-ordinates against time) for both of these cases.

At first sight, this doesn't seem to be a very useful spline. Although the object is speeding up and slowing down so it doesn't get rattled at the corners, it still follows the original straight lines of the control polygon: we don't seem to have smoothed the original polygon at all. But now comes the clever bit: in addition to the position and time at each control point, we record the direction and speed of the object as well, as in Figure 2.



John Jensen

The beauty of this system is that (if we wish) we can record a different speed or velocity depending on whether we're approaching the control point or leaving it. When the velocities are the same, the joint is smooth, and when they are different, the joint is kinked. Figure 3 shows a familiar application of kinks.

Basis functions

How would you evaluate such a spline? First of all, it is clear that we can evaluate each segment in isolation: we know the joints are smooth (when we want them to be smooth), because the velocity of the spline at each joint is defined by the shared control point.

As we saw last month, any cubic curve needs four pieces of information to define it fully. With Hermites, we have two positions and two derivatives (indicated in Figure 2 by the green dots and blue lines respectively). One of our points has position p_1 and derivative d_1, and occurs at time t_1, and the other position p_2, derivative d_2, and time t_2.

We'll take the same approach as we did last month, and find the position s of the spline at time t, where $t_1 \leq t \leq t_2$. At that moment, each of the four pieces of information makes its own contribution to the position of the spline. We find four functions which tell us exactly what contribution each makes, and multiply them by the information itself. In more concrete terms, $s(t) = P_1(t).p_1 + P_2(t).p_2 + D_1(t).d_1 + D_2(t).d_2$. The four functions P_1, P_2, D_1, and D_2 are called the *basis functions* of the spline.

Consider, first, P_1. We know that at time t_1, s should be equal to p_1, ie p_1 makes all the contribution to s, and p_2 makes none. That means that $P_1(t_1) = 1$. At time t_2, p_2 makes all the contribution to s, and p_1 makes none, so $P_1(t_2)=0$. Finally, we know that unless d_1 is non-zero, s will be stationary at t_1, so the derivative of P_1 at t_1 is zero, as it is at t_2.

Proceeding by similar reasoning, we can form the following list of requirements (see Equation 1 overleaf) and, from these, sketch the graphs in Figure 4 and derive the solutions in Figure 5 (which are not half as fierce as they look).

To find several points in a segment, you compute the four basis functions for the segment, with appropriate values of t_1 and t_2. In the case of uniform knot spacing, you can assume $t_1 = 0$ and $t_1 = 1$, and calculate the functions just once – you'll see they become very much simpler. Either way, you plug the functions into the cubic shown above, and evaluate it for different values of t to calculate the position of the spline at different points. You might want to create a forward difference evaluator, or a midpoint evaluator, as shown last month.

Cracking a Hermite is not much harder. You want to find a single new point at $t_c = ½ (t_1 + t_2)$. You can see by inspection that $P_1(t_c) = P_2(t_c)$

dBsee++ automatically generates source code in MFC C++ and C++ Multiplatform.

100% Object oriented code !

Generates C++ Multiplatform app's (Windows, Windows 95, NT, OS/2, Mac, Power Mac, Unix-Motif) !

No Royalties !

Transparent access to DBF and SQL database (CodeBase specific version) !

CodeBase 6.2

DBMS Power for C, C++, VB, Delphi and Java developers

Full xBase compatibility (including .NTX, .CDX, .IDX, .FPT, MDX, .DBT, .CGP & .DBF format's).

Full source code and royalty free distribution !

Unlimited Client/Server engine included.

Very Fast (query 1,000,000 records in under 1 second) and equally compact.

Free with CodeBase - Data Aware Custom Controls and an xBase compatible Report Writer.

30 Day Money back guarantee

Purify for NT

Purify at a Glance

- Identifies execution errors and memory leaks within your application everywhere they occur.
- Enables developers to eliminate run-time problems before they reach the end-user.
- Improves productivity and reduces training time by presenting all information in an easy to use and understandable interface.

For today's complex software development projects, run-time or memory access errors are some of the most difficult problems to solve.

Reading or writing past the boundary of an array, leaking memory unpredictably, using uninitialised memory with random contents; these potentially disastrous errors may take weeks to find and fix.

Purify solves these problems by combining the most comprehensive run-time error checking available with an easy-to-use graphical user interface.

Forget your memory problems with Purify, call 0181 316 5001 for a detailed info pack !

InstallShield Express

The Worldwide Standard for Enabling Software Distribution to All Windows Platforms.

InstallShield Express comes with a selection of pre-built objects to automatically handle installation issues for MS Visual Basic, MS Visual C++, Borland Visual dBase, Borland Delphi, Borland Paradox, Optima ++, Borland C++ and MS Access!

The fast and easy way to create your installation when working to deadline's ! For more details call 0181 316 5001.

MKS Source Integrity

For a Free demo call:
0181 316 5001

MKS Source Integrity is a complete software configuration management system designed to maximise your team-based development.

What's more it involves virtually no learning, allowing you to quickly increase productivity, potect software assets, and guarantee overall source code integrity !

Accusoft ImageGear

AccuSoft
High Performance Imaging

ImageGear - add comprehensive imaging support directly into your application !

Over 45 Raster formats supported.
Scanning support including 32-bit drivers.
Full support for display and printing
Sub-degree rotation
Sub-second decompress
over 200 different functions etc..

TIFF, JPEG, PCX, TGA, PNG, DCX, JFIF, Group 3, Group 4, Group 3 2D, CALS, IFF, IOCA, ASCII, PCD, IGF, ICO, MO:DCA, WMF, PCT, EPS, GIF, ATT, BMP, BRK, CLP, LV, GX2, CUT, DIB, IMG (GEM), IMG (XEROX), IMT, KFX, RLE, MAC, MSP, NCR, PBM, PGM, SUN, PNM, PPM, PSD, RAS, SGI, WPG, XBM, XPM, XWD, DICOM, JBIG, ABIC, CIF

The Pro Gold version of ImageGear gives developers the best functionality and speed on the market today !

For comprehensive details on ImageGear, contact Highlander Software Ltd the official European Distributors on 0181 316 5001

HIGHLANDER
Software Ltd

Tel: 0181 316 5001
Fax: 0181 316 6001
e-mail:sales@highlander.co.uk

CIRCLE NO. 246

VISA Access MasterCard AMERICAN EXPRESS

Govt and Large PLC's - official PO's welcome.
All Trademarks are acknowledged !
All prices specifications and offers remain subject to change without notice.

Barycentric combinations and basis functions

Some basic vector maths is taught in every secondary school, and these articles are assuming that you know some simple rules, such as what Cartesian co-ordinates are, and the parallelogram rule for vector addition. What is less well-known, though, is that there are in fact two different kinds of vector, and we must be careful not to get them confused.

Cartesian co-ordinates are usually introduced as identifying points on a surface or in a space: we describe a tree as being at a point {3,2}, for example, or say that New York is at 74°, 41°. Then we can add directions to them (for example by walking south from the tree or west from New York). These directions are vectors as well, but of a different type: we can add two directions, but we can't add London's position to New York's.

One property splines must have is *frame invariance* – if we move all the control points by the same amount **w**, the result should move by the same amount. That means that if **a+b=c** where **a** and **b** are vectors, then **(a+w) + (b+w) = (c+w)**. Clearly, that's wrong, but that's because we're trying to add London to New York's.

Positions are defined by position vectors (indicated here by upper case letters), and can not be added. We can, however, add direction vectors – indicated by lower case letters – to each other (resulting in another direction), scale them, and add them to position vectors (to produce another position). So, **A+b=C**, and **(A+w)+b=(C+w)**, which is right. Note that, since **A+b=C**, then **b=C-A**: we can subtract two positions to get a direction (as in 'start from the tree, and walk in the direction of the cave').

There is something else we can do with positions: make weighted averages of them, in effect determining their 'centre of gravity'. Normally we write this as $\alpha\mathbf{A} + \beta\mathbf{B}$, where $\alpha+\beta=1$. This looks like we're multiplying the positions by real numbers (which is forbidden), but we're not. The reason this is legal is that $\alpha\mathbf{A}+\beta\mathbf{B} = (1-\beta)\mathbf{A}+\beta\mathbf{B} = \mathbf{A}+\beta(\mathbf{B}-\mathbf{A})$, and $(\mathbf{B}-\mathbf{A})$ is of course a direction. We usually write $\alpha\mathbf{A}+\beta\mathbf{B}$ because it is symmetrical, and doesn't put undue emphasis on A.

In general, $\sum\alpha_i\mathbf{A}_i$ where $\sum\alpha_i = 1$ is a well-defined position, and if all the α_i are between 0 and 1, the position lies within the convex hull of the $\mathbf{A}_i$. It is called a *barycentric combination* of the $\mathbf{A}_i$.

Normally, you have to keep an eye on which vectors are positions, and which are directions, but you have to pay particular attention when handling basis functions. Consider the Bézier basis; all four functions multiply positions, so in order for the combination of points to be barycentric $P_1(t) + P_2(t) + B_1(t) + B_2(t) = 1$ for any value of t. But the Hermite basis is different; two of the functions multiply the positions of the points, and two multiply the derivatives (which are direction vectors). In order for the combination of points to be barycentric, $P_1(t) + P_2(t) = 1$ for any t, but of course there's no such restriction on D_1 and D_2, because they're multiplying directions. You can see the difference in the shapes of the functions.

$P_1(t_1)=1$	$P_1(t_2)=0$	$\frac{\partial P_1}{\partial t}(t_1)=0$	$\frac{\partial P_1}{\partial t}(t_2)=0$
$P_2(t_1)=0$	$P_2(t_2)=1$	$\frac{\partial P_2}{\partial t}(t_1)=0$	$\frac{\partial P_2}{\partial t}(t_2)=0$
$D_1(t_1)=0$	$D_1(t_2)=0$	$\frac{\partial D_1}{\partial t}(t_1)=1$	$\frac{\partial D_1}{\partial t}(t_2)=0$
$D_2(t_1)=0$	$D_2(t_2)=0$	$\frac{\partial D_2}{\partial t}(t_1)=0$	$\frac{\partial D_2}{\partial t}(t_2)=1$

Equation 1 - Referred to on the first page of this article.

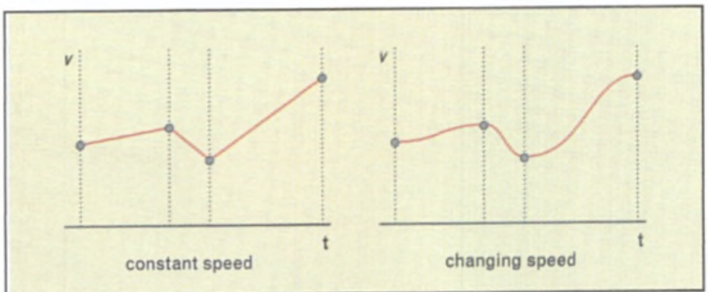


Figure 1 – Moving along a spline at different speeds.

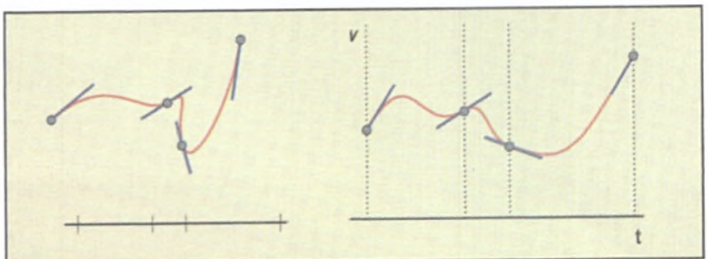


Figure 2 – The effect of velocity vectors at each control point.

= ½, and that $D_1(t_c) = -D_2(t_c)$ (which, for a knot spacing of 1, works out to be 1/8). It's only slightly more difficult to work out the derivative at the point: it's

$$d = \frac{\partial P_1(t)}{\partial t}(t_c).p_1 + \frac{\partial P_2(t)}{\partial t}(t_c).p_2 + D_1(t_c).d_1 + D_2(t_c).d_2$$

Again, you can see from the graph in Figure 4 that

$$\frac{\partial P_1(t)}{\partial t}(t_c) = -\frac{\partial P_2(t)}{\partial t}(t_c)$$

For a knot spacing of 1, that works out at -1½.

With all these powers of two, you'd expect that cracking uniform Hermites would be admirably suited to integer methods. In fact, one of the big advantages of uniform Hermites is that integer evaluators for drawing the splines are easy to construct.

For many purposes, Hermites are all you need; they're intuitive and easy to work with, and they're excellent for modelling animation (where each control point would represent a keyframe, defining the position, velocity, and timing at that point). However, for modelling shapes, Hermites have a big drawback which can catch the unwary user and the harried programmer alike: they are sensitive to the spacing of the knots. In the case of the Aitken spline, you can change every knot spacing in equal proportion, and the spline won't change shape. If you do that to a Hermite spline, the spline will certainly change! The reason is that we've recorded a velocity at each control point, and the knots are spaced out in time.

Imagine two control points, each with a certain velocity, and imagine that their knot spacing is such that the spline can easily cross the distance between them, as in the centre graph in Figure 6. If we push the knots closer together, the spline will have to rush through the intervening space from one joint to another, and if we move the knots further apart, it will have to slow down, and dawdle between the joints (see the other two graphs). In effect, a narrower knot spacing causes

Spline surfaces

When we travel along a line, even a bendy line like a spline, we have only two choices which direction to move – forward and back. We have one degree of freedom, which we call 'time'. When we're moving over a surface, we have two degrees of freedom – two different kinds of time, if you will – which we'll call *t* and *u*. Every point must have a knot, as usual, but every knot must have both a *t* and a *u* defined for it. Because of that, we can't use a control polygon as we did before: we need to form a control polyhedron, or a *control mesh*.

Figure A shows a perspective view of a polygon mesh drawn over the Bézier surface it defines, along with the 'time line', which has turned into a grid. Each line on the grid has four points along it, so each individual line is cubic. The overall surface is described as *bicubic*, because there are two degrees of freedom.

Calculating the shape of this surface proceeds in exactly the same way as before: we can write an expression in *t* and *u*, calculate a point at a particular time $\{t,u\}$ where $t_1 \le t \le t_2$ and $u_1 \le u \le u_2$, or crack the patch into four smaller patches.

Because the principles are the most generally applicable, we will explore finding a specific point at time $\{t,u\}$. Any line on the time grid with constant *t* (a vertical line) or constant *u* (a horizontal line) will be a cubic. Not only that, if the surface is a Bézier surface, the cubics will be Bézier curves. We can evaluate the four curves of constant *t* for the desired value of *u*, giving us four points defining a curve which we can then evaluate for the desired value of *t*. See Figure B.

You can see that, since we're only evaluating lines, the surface could have a different order in *u* to the order in *t*, for example a quintic in *t* and a cubic in *u*. Incidentally, you should be able to satisfy yourself that it makes no difference whether we evaluate first in *t* or in *u* – the result is the same.

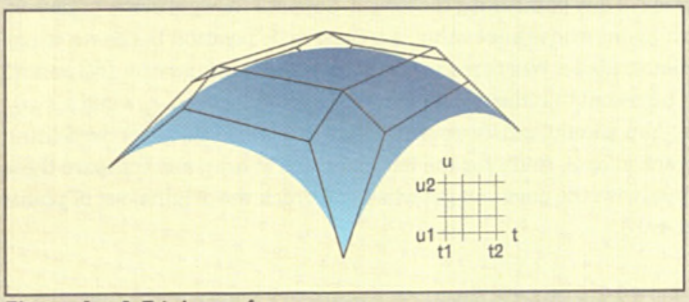


Figure A – A Bézier surface.

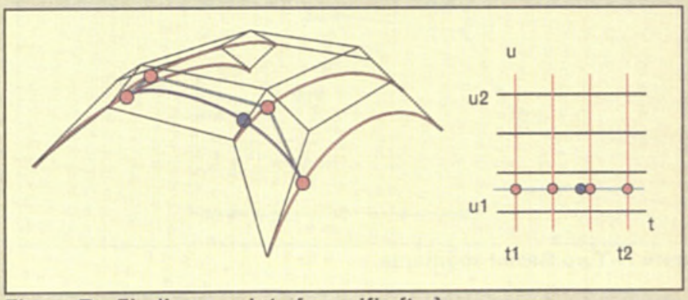


Figure B – Finding a point of specific {t,u}.

the effect of the derivatives to be amplified. This reaches its worst expression when traversing the spline backwards (which is sometimes a worthwhile thing to do). The spline will completely change shape, looping back around itself on every segment!

Bézier curves

Because they largely overcome the difficulties explained above, Bézier curves have become the more popular form. With Béziers we start just as with Hermites, defining two points to represent the limits of a curve segment. However, instead of adding two derivatives, we add two new subsidiary points. Unlike the case for Aitken splines, these two points are approximated, rather than interpolated (ie we smooth the control polygon by filing off its corners): see Figure 7. Usually, the control points are spaced equally in time (though we'll see next month what happens when the knot spacings are unequal).

We can convert between Béziers and Hermites very easily:

$$b_1 = p_1 + \frac{1}{3}d_1 \quad b_2 = p_2 - \frac{1}{3}d_2$$

Given that Béziers are so similar to Hermites, what are the advantages of using them? Firstly, many users find it easier to work with positions rather than vectors, and for some applications the arithmetic is easier (we'll see later how to draw Béziers of any degree). But, there's something more important: Béziers have a property called a *convex hull*.

Figure 8 shows the same curve segments as Figure 7. Imagine a rubber band is stretched around the points describing each segment. The boundary of the rubber band contains an area, called the *convex hull* of the points. Having a convex hull property means the entire length of the spline curve is guaranteed to lie inside this hull.

The convex hull is important because it enables you to keep separate splines separate. Imagine you're drawing splines in a Windows program: when an area of the screen needs repainting, you can ignore any splines whose convex hulls are entirely outside the invalid area. If you're calculating collisions or intersections between two splines, you know that the splines can't possibly overlap unless their convex

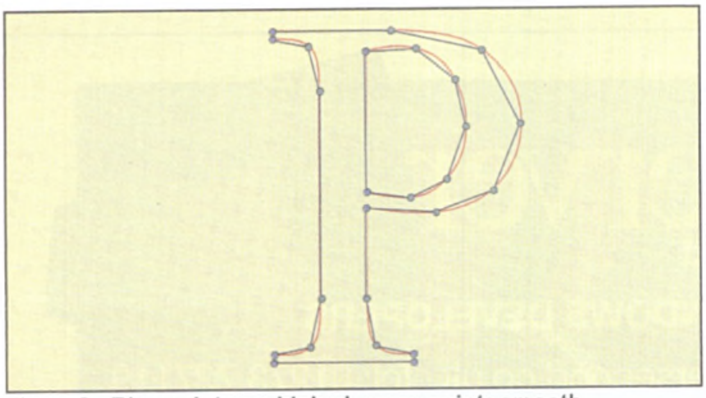


Figure 3 – Blue points are kinked, green points smooth.

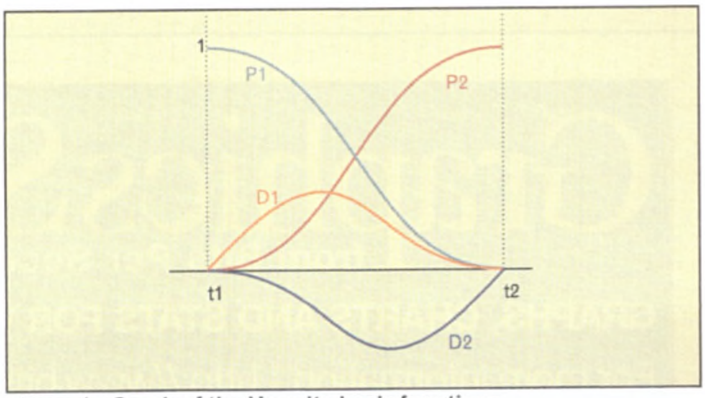


Figure 4 – Graph of the Hermite basis functions.

$$P_1(t) = \frac{\frac{t^3}{3} - t^2\frac{t_1+t_2}{2} + t(t_1t_2) + \frac{t_2^3}{6} - \frac{t_2^2t_1}{2}}{\frac{t_1^3+t_2^3}{6} - \frac{t_2^2t_1 - t_1^2t_2}{2}} \qquad D_1(t) = \frac{t^3 - t^2(t_1+2t_2) + t\left(2t_1t_2+t_2^2\right) - t_1t_2^2}{t_1^2 - 2t_1t_2 + t_2^2}$$

$$P_2(t) = \frac{\frac{t^3}{3} - t^2\frac{t_1+t_2}{2} + t(t_1t_2) + \frac{t_1^3}{6} - \frac{t_1^2t_2}{2}}{\frac{t_1^3-t_2^3}{6} + \frac{t_2^2t_1 - t_1^2t_2}{2}} \qquad D_2(t) = -\frac{t^3 - t^2(2t_1+t_2) + t\left(2t_1t_2+t_1^2\right) - t_1^2t_2}{t_1^2 - 2t_1t_2 + t_2^2}$$

Figure 5 – The Hermite basis functions.

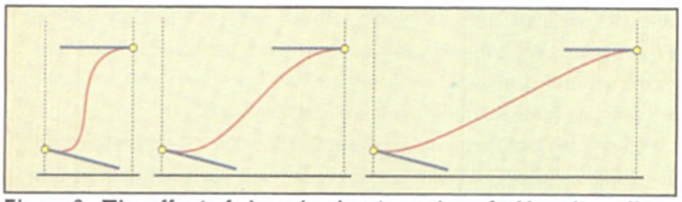


Figure 6 – The effect of changing knot spacing of a Hermite spline.

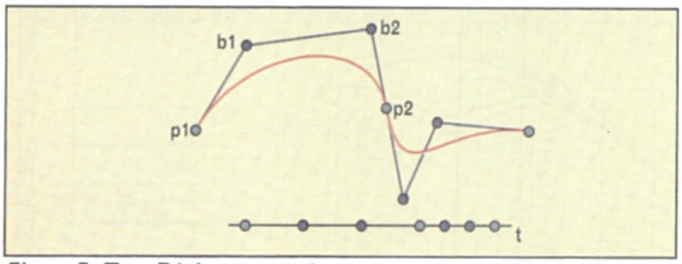


Figure 7 – Two Bézier segments.

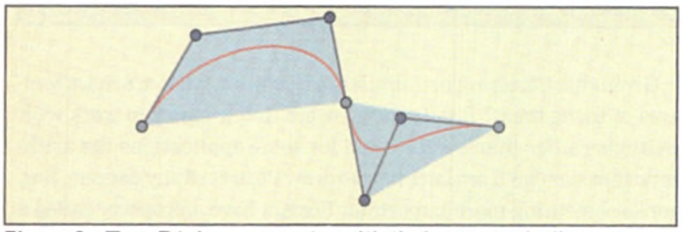


Figure 8 – Two Bézier segments, with their convex hulls.

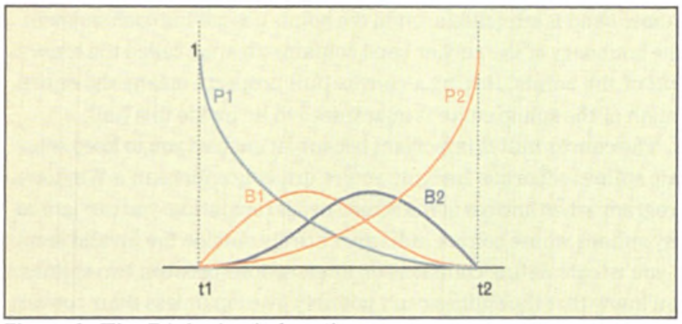


Figure 9 – The Bézier basis functions.

hulls do. In short, convex hulls give you a cheap and easy method of pruning curves without having to calculate them in their entirety.

Calculating Béziers

The basis functions for Béziers are shown in Figure 9. They are defined by groups of equations called Bernstein polynomials – you can read the theory if you want; there's no space to explain it in detail here. Instead, we'll look at a very simple geometrical construction – the de Casteljau algorithm – from which you can calculate the basis functions if you need them.

Figure 10 shows the construction. To evaluate the position of the curve at a time t where $t_1 \le t \le t_2$, divide each line in the ratio α where $\alpha = (t - t_1) / (t_2 - t_1)$ (α will be between 0 and 1). Join these points with lines, and divide these lines in the same proportion. Keep going until you've only got one point; that's the point you're looking for.

Cracking a Bézier uses the same kind of approach, and we can crack at any point along the length. Suppose we want a new main control point whose knot value is t. Clearly, its position is s as we've calculated above. What are the positions of the subsidiary control points? It turns out that the two segments are given by p_1, q_0, r_0, s and s, r_1, q_2, p_2. You should satisfy yourself that this works as advertised (hint: crack at $\alpha=½$, find s for the left hand side at $\alpha=½$, and compare these lines with the ones you get when you crack the original set of points at $\alpha=¼$).

Béziers of other orders

You should have no difficulty generalising both of these constructions to other orders. In fact, one of the nice properties of Béziers is that you can construct splines of very high orders without too much work. As you increase the order of the curve, you'll need more points to define it (five points for a quartic, six points for a quintic, and so on), and the more points you have in each segment, the less pull each point has, so the smoother the curve becomes. You can offer this behaviour to a user in order to enable him to make very slight adjustments to a curve. Suppose the user is pushing a cubic Bézier around to make a shape, and he likes the shape but he wants a gentle bulge in the middle of it. You can replace the cubic with a quintic which is the same shape, but which has different (and more) control points, then the user can carry on adjusting the new points to make his bulge. This is called *degree-raising* of the spline.

It is easiest to raise the degree of a Bézier one step at a time, since this only requires linear interpolation. In Figure 12, we're moving from a cubic (which has four equally spaced knots) to a quartic (with five equally spaced knots). We calculate the new knot values (the shorter dotted lines in the t-v graph), then insert a new point at the intersection of each edge with the new knot values.

Soft Options

FROM THE LEADER IN COMPONENT SOFTWARE

'The Suite is excellent value'... PERSONAL COMPUTER WORLD
'Add Spreadsheet & Charting to programmes with ease'... BYTE. FEB 95'

If you develop using VisualBasic 4.0, Visual C++ 4.0, Delphi 2.0, Powerbuilder 5.0 or Optima C++ then the best way to enhance your application and reduce development time is to add one or more components from Visual Components.

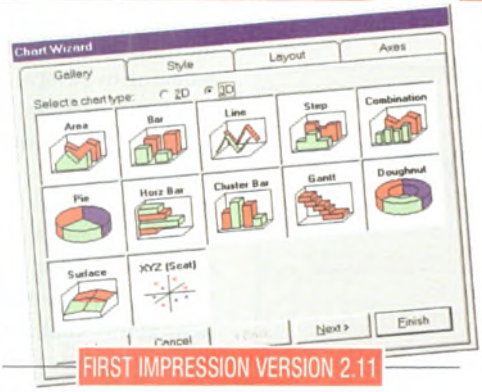
CALL FOR YOUR FREE ActiveX CD

FORMULA ONE VERSION 4.0

This 16 and 32-Bit OLE/ActiveX Controls, footprint is only 772kb, and now has German, French, Spanish and Italian user resources. With Print Preview and 23 built-in dialog boxes,159 methods, 144 properties, 27 events, Clear and format multiple cell selections, including conditional formatting, automatic recognition of data, time, fraction, currency, percent and scientific data entries 130 worksheet functions,19 Operators, Microsoft Excel 4.0, 5.0 and 7.0 compatible, Support for Excel-style workbooks (workbooks can contain as many as 256 worksheets), External and three-dimensional cell references, Relative and absolute cell references, Auto fillcells with both preset and custom lists, ODBC Database SQLQuery and Update, Print scaling, fit to page printing, support for 56 colours Data validation rules for cell entries, Royalty free runtime distribution.

VISUAL SPELLER

This 16 and 32-Bit OLE/ActiveX Control with a 100,000 word English dictionary also supports all major European language dictionaries. Import Microsoft Word custom dictionaries. Load multiple dictionaries simultaneously. Includes dictionary builder, seamlessly works with VisualWriter or any text based application including databases. Automatic or Manual correction. Royalty free runtime distribution.

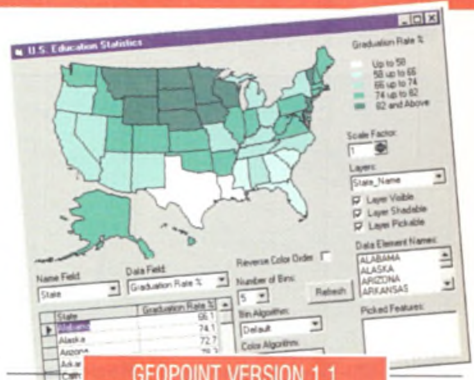


FIRST IMPRESSION VERSION 2.11

A Charting Control both 16 and 32-Bit OLE/ActiveX with built in data grid and editor now has German, French, Spanish and Italian user resources, 26 built-in dialog boxes, More than 300 methods and properties and 25 events, User interface available at design and runtime, Automatic connection to Formula One worksheets/books, Chart Wizard with chart gallery, Linear and logarithmic axis scaling, secondary y-axis, Stackable series, 2D and 3D line styles, Photo-realistic attributes; dynamic elevation and rotation, 3D light sources, ambient and edge lighting, true 3D textures, true 3D perspective, oblique and orthogonal projection, Export charts in BMP or WMF formats, Royalty free runtime distribution.

VISUAL WRITER

With this Rich Text Format(RTF) compatible 16 and 32-Bit OLE/ActiveX you can create, import and export documents. ASCII text (TXT) compatible; create, import and export documents. Native TXF format, WYSIWYG display and output. Built in search and replace functionality. Format characters with multiple fonts, styles and colours. Support for all Window fonts, including TrueType and Adobe Type 1 fonts. Edit multiple documents simultaneously. Easy connection to VisualSpeller, our spell checking component. Three built-in dialog boxes. Create status bars, button bars and rulers. Fully supports embedded images; TIFF, BMP and WMF formats. Fully supports embedded controls. Embedded images can be static or move with text. Database support allows for text controls to be bound. Royalty free runtime distribution.



GEOPOINT VERSION 1.1

Add thematic shading to your applications with this spatial data components. Geopoint is a 32-bit ActiveX control for applications built under Windows95 or Windows NT 4.0. Data binding and Uniform Data Transfer support make this GIS control a great fit for anyone working with databases or spreadsheets. Geopoint binds directly to Microsoft data source controls and Powerbuilder DataWindows to present data in the context of positional or geographic boundaries. An object-based API supports multiple layers and map features for each layer of the map. Native support for MapInfo, Atlas and AutoCAD, Royalty free runtime distribution.

VISUAL COMPONENTS™
The leaders in component software

Call today +44 1892 834343

Lenexa House, 11 Eldon Way, Paddock Wood, Kent. UK TN12 6BE
FAX: +44 1892 835843 BBS: +44 1892 835579 • CIS: GO VISTOOLS
www.visualcomp.com sales@viscomp.demon.co.uk

CIRCLE NO. 249

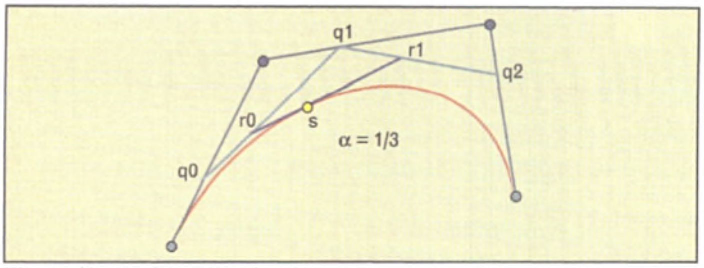


Figure 10 - de Casteljau in pictures.

$$q_0 = (1-\alpha)p_1 + \alpha b_1$$
$$r_0 = (1-\alpha)q_0 + \alpha q_1$$
$$q_1 = (1-\alpha)b_1 + \alpha b_2 \qquad s = (1-\alpha)r_0 + \alpha r_1$$
$$r_1 = (1-\alpha)q_1 + \alpha q_2$$
$$q_2 = (1-\alpha)b_2 + \alpha p_2$$

Figure 11 – de Casteljau as math (cubic case).

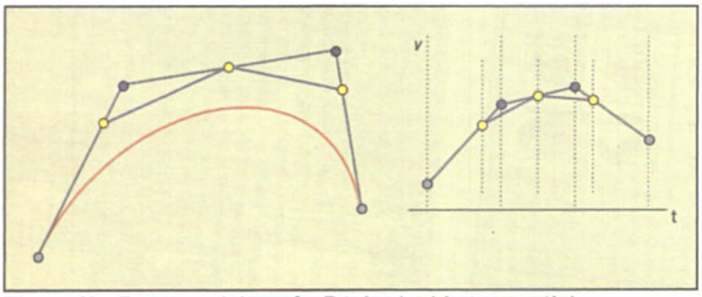


Figure 12 – Degree raising of a Bézier (cubic to quartic).

The opposite process, degree reduction, cannot in general be done without losing accuracy, and so is inevitably a messier process. Say we are moving from a quartic down to a cubic: the problem is to find the closest cubic to the given quartic. Instead of interpolating the edges we're given, we have to extrapolate, and extrapolation will get progressively less accurate the further along the sequence we travel. The trick, then, is to calculate one set of points travelling from left to right, then calculate another set moving from right to left, and then merge the two sets of points together. In Figure 13, the yellow dots are the subsidiary points of the original quartic, the blue line represents the extrapolation from left to right, and the green line represents the extrapolation from right to left. Notice how we've found the right-hand red dot (which is a cubic subsidiary point) – we expect the green line to make a more accurate prediction than the blue line, so we position the red dot closer to the green line than the blue, using a linear interpolation.

In general, you'll achieve a more satisfactory result if you crack the original spline and then reduce the two halves. However, if you're reducing by several orders, and you crack at each stage, you'll end up with an awful lot of segments! As a rule of thumb, consider cracking the original once for every halving of the order you intend to carry out.

Continuity conditions for Béziers

So far, we've been concentrating on the individual curve segments. However, just like Hermites, we will want to join segments end-to-end to get more complex curves, and need to ensure that the curves remain smooth. Assume that we have two segments, c_1 and c_2, each with the

properties described earlier (eg $c_1\{p_1\}$ is the first control point of the segment c_1).

C(0) continuity is obvious: in order for the two curves to meet at all, the last control point of the first segment must be the same as the first control point of the second segment, ie $c_1\{p_2\} = c_2\{p_1\}$.

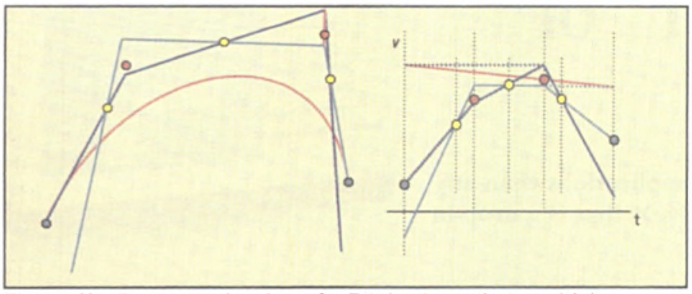


Figure 13 – Degree reduction of a Bézier (quartic to cubic).

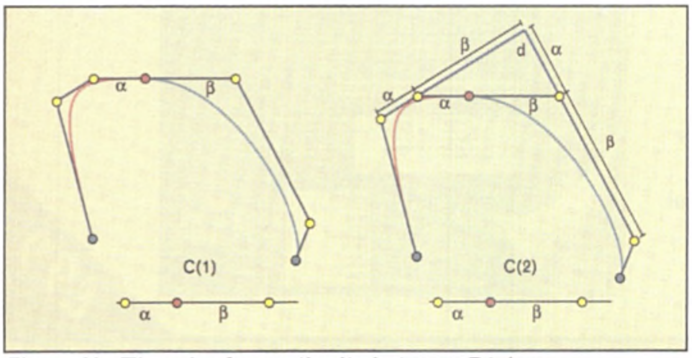
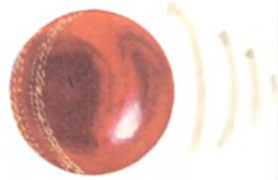


Figure 14 – The rules for continuity between Béziers.

If you want the directions of the curves to be the same at the joint, then you have to ensure that $c_1\{b_2\}$, $c_1\{p_2\}$, and $c_2\{b_1\}$ lie on a straight line, but C(1) continuity requires a little more than this: the speeds have to be the same as well. In order to achieve this, you must ensure that the ratio *length* $(c_1\{b_2\}, c_1\{p_2\})$ *: length* $(c_2\{p_1\}, c_2\{b_1\})$ is the same as the ratio of their knot spacings. See Figure 14a.

C(2) continuity requires that the accelerations of the curves are equal at the joint, which is a little more complex to check. Extrapolate the line $c_1\{b_1\}, c_1\{b_2\}$ and the line $c_2\{b_2\}, c_2\{b_1\}$, and mark the point where they cross as *d*. If

length $(c_1\{b_1\}, c_1\{b_2\})$ *: length* $(c_1\{b_2\}, d)$
$=$ *length* $(c_1\{b_2\}, c_1\{p_2\})$ *: length* $(c_2\{p_1\}, c_2\{b_1\})$
and
length $(d, c_2\{b_1\})$ *: length* $(c_2\{b_1\}, c_2\{b_2\})$
$=$ *length* $(c_1\{b_2\}, c_1\{p_2\})$ *: length* $(c_2\{p_1\}, c_2\{b_1\})$



then the joint is C(2) continuous. Figure 14b makes this clearer.

These rules are true no matter what the degree of the two splines. Cubics, of course, can't do any better than C(2) continuity, but you can match a cubic to a quintic using the same rules, and achieve higher continuities by the same principle. On the other hand, enforcing even C(2) continuity between Béziers is tricky, and makes it hard to provide users with sensible editors. The rule is more useful when working with other kinds of spline, as we'll see next month when we'll meet the B-spline, his buddies, and balls. ■

Jules May is a freelance consultant and programmer. For advice on how to take bends at high speeds, you can contact him at jules@cix.compulink.co.uk.

FasTrack™ Microsoft Training

VB5 Courses Available NOW!

The Very Best Training for Leading Edge, Microsoft Developer Technologies

Microsoft has kicked off its 'Year of the Developer' with the launch of its latest visual tools and its series of world-wide developer briefings. Now ICS Solutions, a Microsoft Regional Director and leading Solution Provider Partner, are offering an up-to-the minute range of courses for Microsoft Developers and for Microsoft Certified Professional qualifications.

FREE Psion Offer

Most Microsoft Certification courses and ICS FasTrack™ Training courses include a FREE* Psion palmtop organiser with every place booked - ask ICS for full details.

Up to Speed in Just a Week

The three ICS FasTrack™ Condensed Courses pack ten days traditional training into five. Tutors include well-known database textbook authors and former Microsoft product managers.

Microsoft Visual Basic 5 ICS FasTrack™ Condensed - £1,395*

Become productive quickly and effectively on the most important of Microsoft's new generation of developer tools.

Microsoft SQL Server ICS FasTrack™ Condensed - £1,395*

Simply the best course around for SQL Server taught by the leaders in Microsoft database computing.

SQL Server/VB Enterprise Development - £1,395*

Practical development tips from the experts in Microsoft Enterprise level development for Windows NT.

Get Certified Now

Microsoft Certified Professional Qualifications are the industry badge of quality for the software professional. Now there is no better high quality route to qualification than through ICS Solutions.

Microsoft Certified Systems Engineer

MCSE - £2,995*

This NT4 focussed package includes 20 days classroom training as well as student study materials aimed at exam success.

Microsoft Certified Solutions Developer

MCSD - £1,995*

8 days of classroom tuition, including VB and SQL Server, with technical documentation. Ideal for the developer with some VB knowledge.

Microsoft Certified Product Specialist

MCPS - £1,995*

Aimed at those wishing to specialise in SQL Server and Windows NT 4.0. 10 days of training with exam-oriented study materials are included.

ICS SOLUTIONS LIMITED

Microsoft REGIONAL 97 DIRECTOR

Microsoft SOLUTION PROVIDER
Authorised by Microsoft

Tempus Business Centre, Kingsclere Road
Basingstoke, Hampshire. RG21 6XG
Tel: 01256 469460. Fax: 01256 842362
http://www.ics-solutions.co.uk/ics/

CIRCLE NO. 255

for further details or to book your place
call 01256 469460
or email: training@ics-solutions.co.uk

All trademarks acknowledged as the property of their respective owners.
* All prices exclude carriage and VAT. FREE Psion offer subject to ICS terms and conditions, may vary from product illustrated - ask for details.

One last thing...

As the deadline for the ISO C++ standard approaches, it has come to light that some parts of the standard library require a feature that the core language doesn't support. Francis Glassborow explains.

Illustration: Paul Phippen

The recent meeting of the C++ standardisation committees was unusual because, under the ISO rules, we were not allowed to make any substantive decisions. While Committee Draft 2 (CD2) is being voted on by the national standards bodies, no issues relating to the content can be altered by WG21. On the other hand, the co-located ANSI X3J16 committee needed to discuss the comments (read 'requested changes') that it was going to ask ANSI to submit to the supervising ISO committee, SC22. As we had had our knuckles rapped for meeting while CD1 was being voted on, we were at pains to keep things politically correct, while at the same time getting some work done.

We actually got quite a lot of work done, and finished on Thursday. Since we could not take any formal votes – other than thanking our host – we gave those travelling on the Friday a better chance to dodge the ice-storms and so forth that add to the joys of New England travel in March.

We agreed that it would make sense to allow `return void` in functions that have a `void` return type but that that was as far as we thought we should go on developing `void` as an incomplete type. I should explain at this point that the keyword `void` has two distinct meanings in C/C++: the first is as an incomplete data type, which is why you can use `void *`, and why (through special dispensation) you can have a `void` return type. The use of `void` to mean 'has no parameters' is redundant in C++, and could become so in C if prototypes were to be made obligatory and K&R-style function declarations removed.

There has always been a certain attraction to allowing `return void`. Consider these two examples of forwarding (wrapper) functions:

```
int work();
void morework();
int wrap_work() { return work(); }
void wrap_morework() { morework(); }
```

Leaving aside nitpicking – in the real world there would be more substance to the above functions – under the current rules the syntax of the bodies of the two forwarding functions must be different. This is just a minor wart when writing simple code but it becomes a major irritant when writing some types of template code. X3J16 decided to recommend the maximum change that could clearly be seen to have no ripple effects on the rest of the language. The change would allow:

```
void wrap_morework() { return morework(); }
```

but it will not allow a call such as:

```
work(morework());
```

In other words, however attractive it may be to some, we do not propose to allow a function returning `void` to be called to provide an 'argument' to a function that takes no arguments.

The second issue that raised quite a lot of discussion was the degree to which we could provide exception safety within the STL containers. Some post-meeting work by David Abrahams (Mark of the Unicorn, Inc) looks promising, so we may yet fix this problem in a reasonably general way before shipping a standard.

The earlier issue of generalised allocators (which provide memory for STL containers) seems to be reaching an acceptable compromise among those concerned. Such a resolution will matter to many of us, not immediately but at some time in the future when distributed and multi-threaded programs become common.

Default template parameters

The crunch issue for the meeting was that of default template parameters. We had realised at the November meeting that there was a problem that needed fixing, had slapped a temporary patch in place and largely assumed that fixing it would be easy and non-contentious.

The problem is that the standard library has been designed with the assumption that the core language would define the meaning of a default template parameter. For example, the following should work:

```
template <typename T, typename allocate = allocator< T > >
    list { … }
```

Now, close to the last minute, we have realised that there is nothing in the language that supports such syntax – a case of experts being too focused on their own corner.

We can fix the problem in one of three ways: remove all defaults from the library and make the user provide the template argument (effectively what Visual C++ does today); remove the defaults but overload the definitions so that the user gets the same result but at the cost of substantial extra visible complexity in the library (effectively what Borland C++ does); or fix the core language to support default template parameters.

There is no doubt that the last of these options is what any sane user would want. Unfortunately, the implementers are very concerned that such a fix would have repercussions elsewhere in the language, and there is little (read 'no') time to investigate. I was surprised to encounter some implementers who believed that the original decision to provide default arguments for normal functions was a mistake. I have always found these useful. However they drew my attention to the difference between the way in which these are traditionally used (which is safe and helpful), and what the syntax inevitably supports.

For example, at what stage is a default parameter checked? It does not matter for the simple scalar type defaults that many of us use. But when we start calling functions to provide default values life can get rather less certain. This is just another example of features that are perfectly reasonable in most code but potentially lethal in mission/safety critical situations. It is often unclear to the average working programmer that some attractive features can have nasty stings to them.

Before you decide that this is a damning problem with C++, you should realise that similar problems arise with increasing frequency as languages become higher level. There is a sort of Heisenberg uncertainty principle in operation. It is easy say what you do not mean in low- level languages, while high-level constructs can result in you not meaning what you say.

In this instance I guess the eventual decision will rely on one person's call. Erwin Unruh (of Siemens Nixdorf) has an extraordinary record in correctly identifying side effects of language changes. If he can identify a way to support template default arguments that he is certain will not ripple outwards, most will accept it. If, however, he is unwilling to say more than that he thinks that the best proposal will work, we will have to resort to a library fix.

New compiler releases

Borland has joined the rush towards visual programming with C++Builder, by and large a version of the Delphi environment for C++ (see the review in last month's *EXE*). The product joins the ranks of tools like Sybase's Optima++ and IBM's VisualAge for C++, which are intended to give programmers one more level of abstraction, making it easy for even naïve programmers to write applications. That's all well and good, but visual programming has a problem: it rests heavily on pre-built components, and if these do not support exactly what you want to do, you still have to cut your own code. More significantly, even if appropriate components exist, your application is relying on their quality. If an application destroys your business because of a fault in a vendor-supplied component, who is going to be held responsible.

On another front, Microsoft has released Visual Studio 97, and my first reaction is one of extreme anxiety, since the product brings together almost all of its development tools under one roof. Are they about to do to PC development what Microsoft Office did (an aside: the grammar checker in Word 97 wanted to replace 'did' with 'done') to office applications? Depending on your viewpoint you may or may not think this a good thing.

Part of the package is Visual C++ 5.0, which from the language lawyer's perspective is an excellent product – it supports almost the whole C++ feature set. If only they would have the humility to put MFC into a namespace I'd be more than happy. I'll leave it to others to comment on the quality of the product (compilation speed, bugginess etc).

Last month's problem

Can you provide two distinct meanings for `x * y;`? The key to this problem is that `*` can mean several things, depending on context. The one that probably first springs to mind is the binary operator for multiplication. If `x` and `y` are variables of types that support the `*` operator either explicitly (via a user definition of `operator *`) or implicitly (because they are of suitable arithmetic types), then the statement will evaluate the expression and discard the result. As some users overload operators by functions that do not have the expected semantics, even the fact that the result is apparently discarded is OK. For example, whether you approve or not, the following is correct C++:

```
#include <iostream.h>
struct Mad {
        int num;
        void operator * (int val) { num *= val; }
} example;

int main () {
        example.num = 2;
        example * 4;
        cout << example.num;
        return 0;
}
```

(I used a `struct` to indicate that I consciously did not want data hiding).

The other possibility is that `x * y;` is simply a declaration of `y` as a pointer to type `x`.

In normal code, the context clearly disambiguates these two meanings for the statement: if `x` is a type then it is a declaration, otherwise it is an expression. However, in a template context it may not be at all clear to a compiler whether `x` will be a type provided by the context of the argument passed to the template parameter or whether it will be some other kind of identifier such as a variable, enumeration value etc. Unless the compiler knows whether `x` represents a type it cannot parse your code.

The `typename` keyword was introduced into C++ to resolve exactly this kind of ambiguity. In the context of a template, any identifier that might or might not be a type name will only be treated as a type if it is pre-declared as such. So:

```
typename x;
x * y;
```

makes `x * y;` a declaration. Without the `typename` declaration it would be treated as an arithmetic expression.

Having brought `typename` into the language it became a clear candidate to distinguish type parameters in template definitions. This is why we can now write:

```
template <typename T> swap (T t1, T t2);
```

instead of the earlier (and still commonly used)

```
template <class T> swap (T t1, T t2);
```

This month's problem

What is wrong with the following code? (It is not the apparently obvious typos).

```
#include <iostream>
int main () {
        int i=4;
        cout << i << endl;
}
```

ACCU conference 1997

Have you booked your place at the C & C++ European Developers Forum 1997? If not, act today and contact Parkway Research, Westwood House, Elmhurst Road, Goring-on-Thames, Berkshire, RG8 9BN, tel: 01491 875 386, email: parkway@rmplc.co.uk. ■

Extended subscriptions till August 1998: individual £22-50, student £11.25, corporate £120, Overload & C++ SIG £52.50 (including ACCU membership). For further information and application forms write to Francis Glassborow, 64 Southfield Road, Oxford, OX4 1PA, ring 01865 246490 or email francis@robinton.demon.co.uk

Portability everyone can agree on

"Zinc's portability is transparent. And only Zinc has a full set of UI objects and real extensibility to develop commercial-grade applications. Full source code is a big plus, too."

DEVELOPMENT MANAGER

"We're now shipping our application on every major platform and in every major world market. Our sales force loves it! With Zinc we're beating the competition."

MARKETING MANAGER

"Zinc cost us very little up front and saved us years of development expense. Revenue is up, thanks to the new markets [Zinc enabled us to enter]. I'm very satisfied."

FINANCE MANAGER

When was the last time Development, Marketing, and Finance all agreed on anything? Now's your chance to make it happen. With Zinc you'll build better applications, on more platforms, in less time, and with less money—and that's a promise.

Only Zinc offers complete portability.

Since Zinc Application Framework is the only cross-platform tool that delivers 100% portability, you'll have your applications on other platforms as fast as you can recompile. And no one supports as many platforms as Zinc does. It's all part of what makes Zinc the most productive—and affordable—tool you can own.

Productivity that leads to opportunity.

Zinc zips through tedious tasks with C++ object orientation and a unique visual development tool. And, enabling your application for international markets is already done—just translate your text. Plus, Zinc is the only tool that supplies 100% of the source code.

It all adds up to productivity. Which means more profitability. Which means everybody's happy—especially you.

PCWEEKLABS ANALYST'S CHOICE JAN. 30, 1995

"Developers seeking easy delivery of GUI applications... will find Zinc their best option by far."

"This product is absolutely the best development environment I have personally seen for the international engineer."

Multilingual Computing

EDITOR'S CHOICE '95 MLC

"Best Portability"

INFOWORLD February 6, 1995

"Zinc came closest of all the products we tested to our ideal of portability... In short, Zinc did a great job."

For free demonstration software and an information packet, please call:

+44 (0) 181 855 9918

USA: +1 801 785 8900 or fax +1 801 785 8996
Europe: +44 (0) 181 855 9918 or fax +44 (0) 181 316 2211
Asia: +81 (052) 733 4301 or fax +81 (052) 733 4328
Electronically: Info@zinc.com or GO ZINC on CompuServe.
Web: http://www.zinc.com/ Ftp: ftp://ftp.zinc.com/

z i n c

NO LIMITS

CIRCLE NO. 257

Flash!

Asymetrix dubs SuperCede the first 'Flash Development Environment' for Java, but how close is it to achieving the increases in programmer productivity it promises? Neil Hewitt listens for the thunder.

The party is over. The ticker-tape cleared away. The music switched off. After eighteen months of solid rolling, the Java bandwagon has finally come to a halt. Now that it's almost mainstream, developers can come to the Java melee confident that the tools are there to support them – or so the vendors say. Although the real situation is rather more complicated, at least there is now a wide selection of RAD and 'trad' tools for the aspiring guru to choose from.

The latest offering to broaden the wide church that is Java development is Asymetrix SuperCede for Java, from the company that brought you SuperCede for C++. Any new Java tool hoping to carve a slice of an already-crowded market has to offer something the competition doesn't have, and in the case of SuperCede this is the much-touted 'flash development' environment with Interactive Compilation.

Interactive Compilation, for the uninitiated, is a technique which allows for changes to be made to a program's source code during the debugging cycle, and have those changes take effect immediately, without having to stop and recompile. This claim alone was enough to pique my curiosity.

Unfortunately, SuperCede will have to contend with tools from Microsoft and Symantec both released in March. Visual Café Pro includes all the goodies found in Visual Café plus extensive database integration features. Visual J++ 1.1 is fully integrated into Microsoft's Developer Studio, and offers a high degree of support for ActiveX. Both products are looking very sharp indeed, so I was keen to see how SuperCede stacks up against them.

Free floating weirdness

Asymetrix has plumped for the 'free-form' float-on-the-desktop environment found in versions of Visual Basic prior to 5.0. This is all well and good if you have an uncluttered desktop and no existing windows open. If, like me, you tend to leave applications running in the background rather than religiously closing them when you move away, you could quickly find everything becoming rather confusing. On the other hand, you can always maximise the SuperCede Project window and simulate the full-screen effect that way.

On start-up, the Project window is always displayed first (see Figure 1). SuperCede remembers which project you were working on last and automatically reloads it, so there's no need to keep selecting the same files manually – a tedious 'feature' of several other environments. Build settings and compiler options live in the Project properties box, which is invoked from here.

In SuperCede, projects consist of *components*. Common sense would suggest that these would include such things as forms, controls and database connectors, but in SuperCede, a component is an applet or application, complete with class files. Multiple applets (or applications) can be grouped into one project, but only one project can be open at any time. Confused? I was.

In order to create new forms and controls, you have to delve deeper, into the component editor which is in fact the guts of the actual development environment (Figure 2). The component editor takes a traditional split-pane approach, with a tree of source files, classes or forms displayed down the left, and the source code or form designer on the right. The form designer is a fairly standard effort, with none of the customisable toolbars or component palettes of the Microsoft or Symantec offerings. In fact there appears to be no easy way of incorporating user-written controls other than by using the canvas control, which defines an area of dialog real-estate as given over to a Java class specified in the control's properties. Frankly, in a modern development environment this just isn't good enough.

One nifty feature of the form editor is the tooltip which pops up when you drag a control, showing the current x/y co-ordinates. Asymetrix doesn't seem to be very keen on keyboard shortcuts, however: remember how, in almost every form editor under the sun, pressing 'delete' removes the highlighted control? Well, not in SuperCede, it doesn't. You have to use the intuitive 'right-click and choose delete' method instead.

This 'take it or leave it' approach pervades SuperCede. Everything you want to do has to be done from scratch. There are no wizards, experts, or whatever you want to call them to make common tasks easier. Neither is there any straightforward way of plugging in other edi-

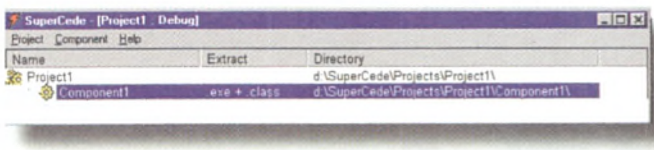


Figure 1 – The project window is the first thing you see.

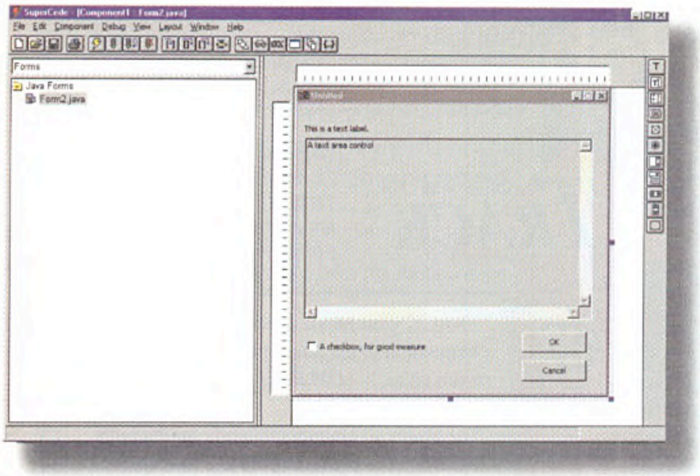


Figure 2 – The form designer in action.

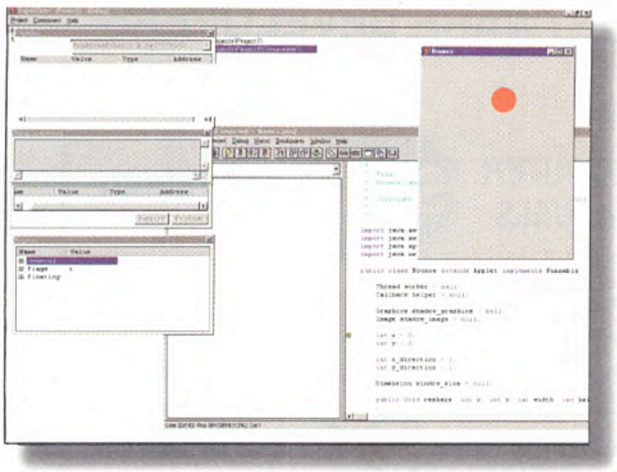


Figure 3 – Debugging the Bounce applet.

tors or tools into the development environment. The customisability potential of SuperCede is a big round zero.

Getting close to the source

The source view is equally conventional, with basic syntax highlighting. There's no method or parameter prompting, and no code completion (a sad loss for fans of Visual Basic 5.0). Source code which is generated by the system is displayed by default as a dark grey block, which is hardly attractive but does serve to make it rather obvious. This code has to be left alone, as any manual alterations will be erased next time the forms are edited.

Implementing the actual event handlers can be tedious, as SuperCede does little of the work for you. The function definitions are generated automatically, and the 'events' page of a control's properties makes it simple to choose which events a control will handle, and jump straight to the appropriate code. The remainder of the coding, however, has to be done by hand.

The underlying SuperCede compiler is quick. But not that quick. While I didn't find the time it took to compile some of the sample programs excessive, there are definitely faster compilers around. The debugger has the features you'd expect on any decent development environment – step-into, step-over, trace, breakpoints, multiple watches (register, memory, call stack, etc) – and functions well enough at that (Figure 3). Where it really comes into its own, however, is in conjunction with SuperCede's star attraction, the Flash Compiler.

I looked in vain for the interactive compilation features until I realised that they were already enabled. In fact, what happens is that when the program source code is changed and a debugging session invoked, the update process begins automatically. It examines the differences between the original and modified source, and generates a binary patch which is grafted onto the executable image, avoiding a full re-compile. The engine used to do this is referred to as the Flash Compiler, and is integrated into the main compiler.

My first attempts to alter the source code of a running applet were unsuccessful, to the point where I began to doubt that interactive compilation actually worked at all. Then I realised that I had been altering sections of the code which were only called at start-up! I then tried altering the 'bouncing ball' sample applet by changing the algorithm which updates the ball position. To my surprise (and amazement) it simply carried on running with the altered algorithm.

More extensive testing revealed that interactive compilation often works without any user intervention other than clicking on the 'update' button. In some cases, however, SuperCede detects that a source code change will affect a critical part of the application, and then asks whether it should restart the whole application, or just the affected routine. The debugger includes a 'pop' function which is used to roll back the execution order from a breakpoint, so that, for example, you could pop back out of an infinite loop to a point before the loop began, alter the loop code to fix the bug, and go on running as if nothing had happened.

The more I thought about it, the more useful this feature seemed. The ability to try quick fixes in situ, without needing to restart or recompile, is a godsend. It's one of those features, like VB 5.0's method prompting, which once you have you don't want to give up.

Users of SuperCede for C++ will be pleased to hear that the two systems can be integrated to form a single-interface environment (à la Developer Studio). C++ class libraries – whether built with SuperCede or not – can be integrated with and called from Java code built in SuperCede, which is a handy feature if you already have C++ code written for your target platform, but is of little use if you're trying to build an entirely cross-platform application.

When the time comes to actually distribute your application, you might be surprised to discover that there are no *.class* files in your project directories at all. In fact, all the object code for your projects is stored in an internal repository from which the dynamic debugging environment fetches it as needed. To produce the class files you need for a distributable applet, the data has to be extracted from the database. SuperCede can transform classes into traditional .class files, or into .exe or .dll files – in the latter case, no visual interface is supported, but instead the Java code can run in the VM behind the scenes. Java DLLs can even be called from native executables (on Windows at least) via the VM. Packaging classes in this form, however, requires the Asymetrix VM run-time libraries to be distributed along with the code.

Thanks a bundle

Netscape's Internet Foundation Classes (reviewed in the March 1997 issue) are bundled with SuperCede, as is ObjectSpace's Java Generic Library, a grab-bag of useful extensions to the JDK classes (based in part on the STL – see *The rebirth of C++* in this issue). ODI's Persistence Class Library completes the tool bundle. All these add-ons are useful products in their own right, and certainly extend the range of what you can do with SuperCede. However – you knew there had to be a however – all of these functions still have to be added in at a source-code level; there is no support for IFC's control classes in the SuperCede form editor, for example.

The product's learning curve is eased by the inclusion of a multimedia tutorial from MindQ (Figure 4). This CD-based course covers most

Figure 4 – The MindQ Interactive tutorial (with gratuitous Paul Allen shot)

Figure 5 – Que's Java reference is provided as online help.

aspects of the SuperCede environment in an instructive fashion, with plenty of step-by-step help for beginners, and lots of friendly diagrams and spoken instructions. A few video clips are scattered around the tutorial too, including one starring Paul Allen himself, in which he professes to be very excited about SuperCede for Java. Lucky old him. The American advertorial-style voiceover gets to be thoroughly tedious after a while, so it's a relief to discover that it can be turned off. All in all, though, this is a useful addition to the package which, amazingly enough, actually does help you to learn something.

Que's *Java Quick Reference* is supplied in an online version (Figure 5), which makes looking up class and method references a doddle. This is context-sensitive and hooked into the source code editor so it takes no time at all to find what you need. Not as good as the parameter prompting features of VB 5.0 and Delphi97, but better than many online help systems I have used.

The final part of the SuperCede line-up is the Asymetrix SuperCede Java Virtual Machine itself. The Asymetrix VM features download-time just-in-time compilation which – in English – means that Java bytecodes are compiled to native code by the VM as the code downloads. The native code translations are stored in a cache for immediate retrieval if the applet is invoked again. This is still a step ahead of other companies' VM technology. It is also – as I understand it – the mechanism used to implement the interactive compilation function. To get the Asymetrix VM working in the Netscape environment requires a plug-in which contains the VM code and replaces the in-built Netscape JVM. All relevant plug-ins and documentation for the VM are included in the package.

Despite the quality of the bundled software, there's still quite a bit missing from SuperCede compared to rival tools. There's no out-of-the-box database integration, for example. Both Microsoft's Visual J++ and Symantec's Visual Café Pro come with full ODBC connectivity out of the box, as will Borland's JBuilder. There were other annoying omissions as well: none of the supplied sample programs came with a SuperCede project file, which meant that to load and compile them I had to build a new project and import the files by hand.

A flash in the pan?

It's hard to see precisely which market Asymetrix is targeting with this product. RAD Java tools would probably appeal more to the Web and database developer, which is no doubt why Microsoft *et al* have placed such a focus on integration and connectivity, but the base SuperCede seems ill-prepared for either market.

Had SuperCede been released twelve months ago, I suspect I would have been impressed. But in 1997 with the likes of Visual J++ and Visual Café Pro already available, SuperCede looks hopelessly dated in all but one respect: interactive compilation.

I found the Flash Compiler system to be a great time-saver and a true aid to the RAD process. If Asymetrix can get the interface right in a future release, this might well be the 'killer' Java tool we've been looking for. But unless your Java needs are suited to the limited abilities of the SuperCede interface, I'd suggest you might want to look elsewhere for now. ■

SuperCede costs £119 from Asymetrix on 01923 208433. Asymetrix has announced two new editions of SuperCede, SuperCede ActiveX edition and SuperCede Database edition, priced at $199 and $399 respectively. Availability dates have not yet been announced. More information can be found on the Asymetrix Web site: http://www.asymetrix.com.

The C++ video course – foundation reviewed by **Kevlin Henney**

So you know C, and now you want to know C++ (or perhaps, more cynically, 'CV++'). There are various options open to you: books, instructor-led training (ILT), computer-based training (CBT), video courses, or good old fashioned 'hack it and see'. You currently have *EXE* in your hands, so I presume that you believe in the written word, and that you also have enough experience to discount hack it and see as the sole method by which your company should secure its future.

In C++ all use must be preceded by declaration, and here's mine: among other roles, I manage and deliver C++ courses for QA Training, so clearly I have some vested interest in ILT. However, I believe that a training course (or a video set) is only one part of the whole learning picture. In one week you will not turn an average C programmer into a top flight C++ programmer, but what you can do is provide the necessary foundation, with a path mapped out to the future in the form of pointers to other sources of information, lab sessions and so forth.

So what about videos? The Silicon River C++ video course aims to introduce C programmers to C++ over the passage of 60 hours, comprising 12 hours of video, with the remainder theoretically spent working with the supporting exercises and source code.

The course is broken over 6 videos, with 22 sections, each covering a general topic such as special member functions and operator overloading. Sections are further broken into detailed units, such as *the copy constructor* and *calling virtual functions*, of which there are 46 plus 9 workshops. The non-video materials follow and supplement the video presentation, but are quite basic.

Technically the most rewarding section for C programmers is *C++ as a safer C*, which comprises the first section and the whole of the first video. After that, unfortunately, the order in which topics are covered and linked together becomes less clear. For instance, the second video spends a great deal of time on conversion operators, assignment operators and copy construction (an area I would group together as 'language mechanics') – and all this before classes and encapsulation have been introduced. It is difficult to appreciate why controlling copy construction is important when you are only dealing with unencapsulated types.

Overall, the course flow gives the wrong weighting to such language mechanical issues. When structuring a program, understanding how to represent object and class relationships in C++ is the first major obstacle, not special operators, friends, default behaviour and so forth (which are all to do with housekeeping). The video and work book provide a useful resource in this respect, but it is likely that most people will want to cut to the chase.

Some time is spent on object and class relationships, but disappointingly many of the examples are quite weak: a janitor class is defined as containing a cupboard object by value rather than by reference or via a pointer. It strikes me that this would make life as a janitor both socially challenging and more than a little uncomfortable.

Many of the concepts of inheritance and polymorphism are made inaccessible by a deep discussion of the I/O streams library. While it is true that I/O streams often don't get the coverage they deserve, I suspect that their presence is more a result of being one of the presenter's pet areas of interest, rather than their value as an example – there is even a totally bizarre implementation of a string class in terms of `streambuf`. Many of the individual examples used to illustrate inheritance and polymorphism are pretty grim, too, and abstract classes are relegated to pretty much the end, receiving a lot less than the Warhol-minimum fifteen minutes of fame they deserve. On the bright side, though, there are good guidelines on why protected data members are not such a great idea.

The structure of the course is at once both linear and depth-first: each topic is entered and explored exhaustively (with respect to both the subject and the viewer). Many programmers will feel bombarded by information, not all of which is relevant to them when they first see it – it is difficult to evaluate how important a fact is when you are new to the area. This is a danger inherent to video based training: with CBT, users can work at their own pace, taking shortcuts and detours as necessary, but with a video the only interaction available to you is through the rewind and fast forward buttons, and they just don't cut it.

An alternative approach would have been to split the course into two, with the first half being the foundation proper, leaving the second half to deal with more advanced topics such as language mechanics, trivia, implementation details and techniques such as copy-on-write. This approach would build more confidence in the course users by allowing them more time to try out concepts, revisiting them later to recap and develop their understanding.

The whole show is run by Nigel Evans, who stands and gestures in front of a black background where the code and diagrams appear. He is enthusiastic and technically well grounded in C++, but with a video you are in a non-interactive environment and it is important to realise that as such only three things matter: presentation, presentation, and presentation.

Over the course of the 12 hours of video, I failed to establish a rapport with Nigel, which I think underlines the difficulty of video-based training: there is no dialogue, and no feedback.

Overall I feel that the technical accuracy of the course is better than that of a number of the books crowding the C++ shelves in bookshops, with up to date references made to the draft ISO standard and library features, but as a training product it is let down heavily by presentation, course structure and C++ usage issues.

✓ **Verdict:** *Hmm, not great, but could prove useful to a few C programmers when supplemented with additional training or reading material.*

Personal Edition (6 videos a total of 12 hours in length, workbook, answer and source disk): **£199.95**

Five Pack (6 videos, 5 workbooks, answer and source disk): **£299.95**

Company Pack (2 sets of 6 videos, 5 workbooks, answer and source disk, plus video interview tape on C++ and OOP): **£499.95**

C++ Advanced Video Module – STL: ...**£49.95**

C++ Advanced Video Module – Exception Handling: **£49.95**

Silicon River: 0181 317 7777 fax: 0181 316 7778
URL: http://www.siliconriver.co.uk

SUBSCRIBERS CLUB

Discounted books – exclusively to you

SAVE 20 PER CENT

C++ Interactive Course

By Robert Lafore

Price: £46.95 **Price to you: £37.60**

This course teaches object-oriented programming in C++ from the ground up, using a completely interactive approach. Readers learn everything they need to write full-featured C++ programs at their own pace. An on-line 'mentor' is available to answer the reader's specific C++ questions. On-line testing offers immediate feedback on the reader's progress.

SAVE 20 PER CENT

Teach Yourself C++ Programming in 21 Days

By Jesse Liberty

Price: £23.00 **Price to you: £18.40**

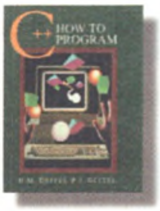
This book is a timesaving guide to learning C++. Using the proven techniques of the Teach Yourself series, this essential guide teaches users how to write programs in a matter of days. Covers all the key topics and features of C++, as well as the basics of object-oriented programming. Teaches C++ without the need to know C or any other programming language.

SAVE 20 PER CENT

C++ How to Program

By HM Deitel and PJ Deitel

Price: £24.95 **Price to you: £19.95**



This authoritative introduction to C++ enables inexperienced or intermediate programmers to grasp the concepts of this popular programming language. It is unique among books on this subject in that it starts from the very beginning and explains the complexities in a clear, concise and understandable language abundantly illustrated with numerous working C++ program examples.

SAVE 20 PER CENT

Windows NT 4.0 Server Unleashed

By Jason Garms, et al.

Price: £54.95 **Price to you: £43.95**

The Windows NT server has been gaining tremendous market share over Novell and the new upgrade – which includes a Windows 95 interface – is sure to add momentum to its market drive. To that end, *Windows NT 4.0 Server Unleashed* is written to meet that growing market. It provides information on disk and file management, integrated networking, Backoffice integration and TCP/IP protocols.

Selection	RRP	Your Price
C++ Interactive Course	£46.95	£37.60
Teach Yourself C++ Programming	£23.00	£18.40
C++ - How to Program	£24.95	£19.95
Windows NT 4.0 Server Unleashed	£54.95	£43.95

Details of all books below can be found in EXE Magazine, August 96 - April 97

Selection	RRP	Your Price	Month
Requirements Engineering	£24.99	£19.99	Apr 97
TCP/IP Unleashed	£49.95	£39.95	Mar 97
Netware Unleashed	£35.50	£28.40	Mar 97
ActiveX from the Ground Up	£22.95	£18.35	Feb 97
Visual J++ Handbook	£22.95	£18.35	Feb 97
The Cyberia Guide to Smart Web Publishing	£49.99	£39.99	Feb 97
The Java Language Specification	£30.95	£24.75	Jan 97
The Java Tutorial	£26.60	£21.25	Jan 97
The Java Class Libraries	£36.95	£29.55	Jan 97
Client/Server Unleashed	£41.50	£33.20	Dec 96
JavaScript Essentials	£26.95	£21.55	Dec 96
The Java Virtual Machine Specification	£30.95	£24.75	Dec 96
The Essential Client/Server Survival Guide	£24.95	£19.95	Dec 96
Teach Yourself Java in Café in 21 days	£36.50	£29.20	Nov 96
Graphic Java – Mastering the AWT	£25.50	£20.40	Nov 96
Java Developer's Guide	£46.95	£37.55	Nov 96
Presenting ActiveX	£27.95	£22.35	Nov 96
The Java Handbook	£20.95	£16.75	Oct 96
Delphi in Depth	£32.95	£26.40	Oct 96
Expert C++	£26.95	£21.55	Oct 96
Pattern-Oriented Software Architecture	£24.95	£19.95	Oct 96
The Late Night Guide to C++	£24.95	£19.95	Oct 96
Writing Compilers and Interpreters 2e	£40.00	£32.00	Oct 96
The Java App Programming Interface Vol 1	£36.95	£29.56	Sep 96
The Java App Programming Interface Vol 2	£36.95	£29.56	Sep 96
Java Essentials for C and C++ Programmers	£24.95	£19.95	Aug 96

BOOK ORDERS

Title **QTY** **PRICE**

____

____

____

____

____

____

____

Shipping at £3,50 per order **TOTAL**

My Subscribers Club Number: ________________

Payment Options Cheques or purchase orders only

❐ 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.**

Name ________________

Address ________________

____

Postcode ________________

EXE DIRECTORY

Call Mark Parker on 0171 287 5000 for details

TRAINING

Computer Training & Education
34-36 Rose Street
North Lane
Edinburgh EH2 2PL
Tel. 01345 697611

Informix Software
Informix House
Littleton Road
Ashford
Middlesex TW15 1TZ
Tel. 0181 818 1010

Kingston University
Faculty of Technology
Penrhyn Road
Kingston Upon Thames
Surrey KT1 2EE
Contact Linda David
Tel. 0181 547 7054
Fax 0181 547 7971
Kingston University is a charity dedicated to the development of individuals and organisations through education.

Learning Tree
International Ltd
Mole Business Park
Leatherhead
Surrey KT22 7AD
Contact Jan Mott
Tel. 01372 364600
Fax 01372 364611
uksales@learningtree.com
WWW .learningtree.com

Merlin Training and Development Ltd
The Derwent Business Centre
Clarke Street
Derby DE1 2BU
Tel. 01332 201911
Fax 01332 201912
Specialist tailor made courses for groups of people on your premises, or public courses in Derby.

Network Consultants
7 West Bar
Banbury Oxon OX16 9SD
Tel. 01295 253689
Fax 01295 271218

Peritas Ltd
Burfield Road
Old Windsor
Berks SL4 2JP
Contact: Mari Shepherd
Tel. 01753 604138
Fax 01753 604160
http://www.peritas.com
Peritas is the largest UK IT Training Company. A Microsoft Authorised Technical Education Centre supporting the MCP Programme including Microsoft Certified Solution Developer Certification.

QA Training Ltd
Cecily Hill Castle
Cirencester
Gloucestershire GL7 2EF
Contact Amanda Whitehead
Tel. 01285 655888
Fax 01285 644828
www.qatraining.com
Leaders in IT technical training and consultancy.

Rhino Training
PO Box 1087
Bristol
BS12 2XY
Contact Cristine Shakespeare
Tel. 01454 417057
Fax 01454 417067
100525.2054@compuserve.com
WWW .rhino-software.co.uk
Providers of training in; Visual Basic, Visual C++, Windows NT, SQL Server, Delphi, Java, Internet/Intranet

Richfords Computer Services
South Bank Technopark
90 London Road
London SE1 6LN
Tel. 0171 922 8819
Fax 0171 922 8839

Skilladvance Training
707 High Road
Finchley
London N12 0BT
Contact Bill Cosgrave
Tel. 0181 446 6481
Fax 0181 446 9143
training@skilladvance.co.uk
WWW.skilladvance.com
High quality Training in Unix, Informix, Oracle, Windows NT, Access, Visual Basic, Internet and Networks.

SECURITY PRODUCTS

Aladdin Knowledge Systems UK Ltd
Tel. 01753 622266
Fax 01753 622262
sales@aldn.co.uk
http://www.aks.com
Aladdin is a leading supplier of advanced software security (HASP) and smart card development tools (ASE) for software developers.

BL Computer Security Ltd
101 Hendon Lane
Finchley
London N3 3SH
Tel. 0181 343 0734
Fax 0181 346 2672
bl@blcs.co.uk
www.blcs.co.uk
We specialise in design and manufacture of computer security products. Anchol, Lure Booster, Deadlock(Dongles) and C.L.A.M.P Alarms

Data Encryption
Systems Ltd
Silver Street House
Silver Street
Taunton
Somerset
TA1 3DL
Contact Roy Davidson(Sales)
Tel. 01823 352357
Fax 01823 352358
www.des.co.uk
deskey@silver.cityscape.co.uk
DES manufactures software security products developed as a solution to software piracy and theft

Glyn Williams & Associates
Ladywood House
Ladywood
Near Droitwich Spa
Worcestershire WR9 0AJ
Tel. 01905 757700
Fax 01905 757800
gwa@gwassoc.demon.co.uk
Software/Hardware copy protection systems - The professional choice for security and features. Worldwide support.

Rainbow Technologies Ltd
4 The Forum
Hanworth Lane
Chertsey
Surrey KT19 9JX
Tel. 01932 570066
Fax 01932 570743
sales@uk.rnbo.com
Only Rainbow delivers leading edge technology and ISO certified quality for software protection and license management.

Softlok International Ltd
Softlok House
2 Bark Street East
Bolton
Lancashire Bl2 3FR
Tel. 01204 436000
Fax 01204 436025
sales@softlok.com
Protect your future, protect your software. Established in 1987, Softlok specialises in software piracy protection.

DEVELOPMENT TOOLS

Bits Per Second Ltd
14 Regent Hill
Brighton BN1 3ED
Tel. 01273 727119
Fax 01273 731925
rflowers@bitspersecond.co.uk
http://www.bitspersecond.co.uk
Graphing/GIS tools. Centura support of training. Client/Server design & consultancy. Network storage consultancy. Multi-platform device driver development.

Borland International Ltd
8 Pavilions
Ruscombe Business Park
Twyford
Berkshire RG10 9NN
Tel. 01734 320022

Citadel Software Ltd
Coombe
Trewen
Launceston
Cornwall PL15 8QF
Tel. 01566-86037
Fax 01566-86147

ComponentSource
27-37 Vachel Road
Reading
Berks RG1 1NY
Tel. +44 (0)118 958 1111
Fax +44 (0)118 958 9999
Request FREE CD or product
sales 101320.2624@-compuserve.com
www.componentsource.co.uk
TRY freely available demonstations of hundreds of software components and BUY and unlock full versions instantly, from a FREE regular CD.

Exepos
Acorn House
Straight Bit
Flackwell Heath HP10 9LS
Tel. 01628 533143

Highlander Software
112 Powis Street
London SE18 6LU
Contact Justin Robinson
Tel. 0181 316 5001
Fax 0181 316 6001
sales@highlander.co.uk
Suppliers of high quality development tools for C, C++, Visual Basic, Delphi and Java.

IBM United Kingdom
Freepost ACG 5022
Wintermill
Milton Keynes
Tel. 0800 969045

Intasoft Ltd
Tresco House
153 Sweetbrier Lane
Exeter EX1 3DG
Tel. 01392 217670
Fax 01392 437877
sales@intasoft.co.uk

MKS UK Ltd
239 Kilburn Park Road
London NW6 5LG
Tel. 0171 6240100
Fax 0171 624 9404

Popkin Software & Systems
St Albans House
Portland St
Leamington
Warwickshire CV32 5EZ
Tel. 01926 450858
Fax 01926 422165

Powersoft Europe Ltd
Windsor Court
Kingsmead Business Park
High Wycombe
Bucks HP11 1JU
Tel. 01628 34500
Fax 01628 38660

Programming Research Ltd
Glenbrook House
1/11 Molesey Road
Hersham,
Surrey KT12 4RH
Tel. 01932 888080
Fax 01932 888081
Contact Mr John Heathcote

Pure Atria
Wyvlos Court
Swallowfield
Nr Reading
Berks RG7 1PY
Tel. 01344 462500

QBS Software Ltd
11 Barley Mow Passage
Chiswick London W4 4PH
Tel. 0181 956 8000
Fax 0181 956 8010
orders@qbss.co.uk
www.qbss.com
Vast range of development products: Next Day delivery; 90 days free support; account customers welcome.

Quadron
209 East Victoria Street
Santa Barbara
CA 93101 USA
Tel. +1 805 966 7630

Quite Software
105 Ridley Road
Forestgate
London E7 OLX
Tel. 0181 522 1726
Fax 0181 2571044

Select Software Tools Ltd
West Moreland House
82-86 Bath Road
Chelenham GL53 7JT
Tel. 01242 229700
Fax 01242 229701
http://www.selectst.com
Select Software Tools provide OO modelling tools for large scale Client/Server development.

Silicon River
58-60 Beresford Street
London SE18 6BG
Tel. 0181 316 7777
Fax 0181 316 4138

System Science
1-6 Bradley's Close
White Lion Street
London N1 9PN
Tel. 0171 833 1022
Fax 0171 837 6411

Zinc Software UK Ltd
106-108 Powis Street
London SE18 6LU
Tel. 0181 855 9918
Fax 0181 316 7778

PUBLISHING

AP Professional
24/28 Oval Road
London NW1 7DX
Contact: Rachel Bridgman
Tel. 0171 482 2893
Fax. 0171 267 0362
app@apuk.co.uk
www.europe.apnet.com/approfessional
Book publisher on internet development,programming, PDAs, software agents and more. Specialists in Macintosh books.

IDG
61/63 Uxbridge Road
Ealing
London W5 5SA
Tel. 0181 579 2652

HELP SYSTEMS

Peterborough Technical Communication
8 Whitewater
Peterborough PE2 6FB
Tel. 01733 237037
Lo-*call* 0345 419470
Fax 01733 239933
petecom@bcs.org.uk
www.gold.net/petecom/
Creation of manuals and on-screen help systems. Please call or e-mail for our free information pack.

Please arrange______yr(s) entry into the EXE Directory for me/my company.

❑ YES I want a one year entry at £240 (+ VAT) ❑ YES I want a two year entry at £400 (+ VAT) + an extra 5 words free*

Name: .... Job Title: .... Company Name............................

Address: .... Postcode............................

Tel: .... Signature: .... Date: ....

Directory Heading: .... Fax No: .... Contact Name: ....

Contact Tel: .... Extra words: ....

❑ I enclose a cheque for £_______made payable to Centaur Communications. (Please write 'EXE' on the back of all cheques)

❑ Please invoice my company

**For extra words on a one year booking the cost is £10 per word*

REAL-TIME WINDOWS

This small successful Designer and Manufacturer of Software and Hardware design tools for the Automobile, Telecoms and Related Industries seeks a broad based Software Engineer to join their team.

You will be working on New products designed to complement New Technology coming from the Microprocessor Industry 16/32 and 64 bit technology. You will be designing GUI applications for Windows 95, NT and Unix platforms, using C, C++, Visual C++, XVT etc.

The company will be interested in hearing from Junior and Senior Software Engineers from 1 year to 10 years experience of software design using any of the above. The ideal candidate will be Degree qualified looking for their next move within the Industry.

The company is also interested in Real Time embedded Software Engineers possibly seeking a move into Windows design, you will need a good understanding of Hardware, preferably Motorola and in depth knowledge of time critical applications development.

All positions are based in Berkshire with Salary and benefits dependent upon experience, guide c£18K to £30K+

We are also very keen to hear from Software Engineers with a Real Time C and UNIX background for positions involving applications development from Control Systems to Digital Mapping. Positions based in London and Sussex. Salaries £20K to £35K. Call for more details.

We have similar positions based in the USA San Francisco developing security systems for the Worlds Airlines. Salaries from $50,000 to $80,000. Call for details.

Email:- ashassoc@globalnet.co.uk
URL: http://www.globalnet.co.uk/~ashassoc

ASH Associates

COMPUTER RECRUITMENT CONSULTANTS
First Floor, 39 to 41 High Street
Ringwood, Hants, BH24 1AD
Email: ashassoc@globalnet.co.uk
TEL:(01425) 475480 FAX:(01425) 480807

Call James Hunt, Ron Cook or Kaye Chambers Now! Telephone 01425 475480

In and Around West Yorkshire

We have clients currently seeking the following:

Windows Programmers or Visual BASIC Programmers to retrain in DELPHI working in a systems house environment — £18,000 to £23,000

Real Time Programmers for 'C' & 'C++' development (supervisory control & data acquisition systems), involves some work for whisky distilleries! — circa £20,000

Real Time Programmers 'C' & 'C++' under Windows to develop in car products, working with CAN (Control Area Network) technology — £15,000 to £30,000

Software Team Leader with Visual BASIC & SQL experience for new client server development working on industry leading re-insurance systems — circa £22,000

Analyst Programmer to undertake a long term project to completely re-develop all the systems for a very successful medium sized manufacturing company. This job starts with a clean sheet of paper — to £25,000 + car

For your next career move around West Yorkshire telephone Vincent Atherton on Leeds (0113) 250 4560 or write to:

Airedale Recruitment

Realtex Hs., Micklefield Lane, Rawdon, Leeds, LS19 6 AX

EXE

Sub editor

EXE magazine is looking for a sub editor to work on both the magazine and its Web site. We are looking for a person with the ability to transform drab copy into sparkling prose, while subbing out cliches like that one.

You will be comfortable rewriting technical features and news stories, and able to produce headlines and standfirsts to tight deadlines. An organised mind is essential, and experience of HTML and using Quark XPress a definite plus.

Opportunity to write for the magazine and the Web site.

Competitive salary depending on experience and company benefits.

Please apply with a full CV, examples of your work and a 300-word critique of any UK computing title to David Mery at the address below:

EXE Magazine, Centaur Communications Ltd, St Giles House, 50 Poland Street, London W1V 4AX (or email dmery@dotexe.demon.co.uk)

meet the people you need to meet at Recru

RecruIT 97 is a vital source of IT contacts in your search for lucrative employment.
Admission is free. For further details and priority registration visit the Rexlive Web Site today at:
http://www.rex.co.uk
e-mail: rex@rex.co.uk or
telephone +44 (0)1344 27333.

RecruIT 97 is organised by

RECRUITMENT XCHANGE

experience the cyber cafe at all our events

EXE3

Recruitment Exchange Ltd, Garden House, London Road, Sunningdale, Berks SL5 0LE.
Tel: +44 (0)1344 23293 Fax: +44 (0)1344 872938

FREE THE SOFTWARE!

At Cygnus we have some unique views about letting users have access to our products. That is why more developers use the GNUPro toolchain across more host-target platforms than any other product. But it's not just with our tools. At Cygnus, we're developing new technologies and products that will reach more customers through our innovative business model. Find out more about these and other opportunities at the Cambridge UK offices of Cygnus.

Current Vacancies
GUI Design Architect £40-45K PA, GUI Engineers £30-35K PA

In addition, we are always on the lookout for gifted compiler and interpreter engineers, and engineers with experience in embedded/real-time development.

Essential Experience
The GUI Architect must demonstrate in-depth knowledge and experience of user interface design and implementation and the GUI engineers must have proven track record of GUI implementation.

The jobs entail specification, design and implementation of user interfaces for innovative embedded system development tools. These tools will be developed for a range of platforms including WindowsNT/95 and Unix variants. Part of the work required will include the development of techniques to enable dynamic run-time construction of sections of the user interface.

Relevant Technical Knowledge
C, C++, TCL/TK. NT/W95, Win32, Unix, X/Motif, 2D graphics, Cygnus GNU toolchain and other IDE's, IDE implementation, embedded and real-time systems, and operating systems.

Education
Knowledge to Degree level is expected. Proven long term experience and expertise are however more important than strictly academic qualifications.

Personal Skills
We are looking for enthusiastic, motivated individuals who have a positive approach to quality. Good communication and the ability to work well in a team are essential.

About Cygnus Solutions UK
Cygnus is the pre-eminent developer and supplier of leading edge programming tools for the embedded/real-time developer marketplace. Most of the software that we produce is completely free, even the source. Our revenues come from custom development for silicon manufacturers and OEM's, and from customer support contracts. We have offices in Silicon Valley, Boston, Japan, Germany and Cambridge UK and employ around 100 professionals worldwide.

You will work in an informal environment with much flexibility as regards working hours and dress. Management is by results, not appearances. We treat engineers for what they are - our most important asset. You will be required to travel to the US for engineering meetings at least twice a year. Other travel may be required from time to time.

Check out http://www.cygnus.com for more information.

Contact
Paul Beskeen, Director of Engineering, Embedded Components
Cygnus Solutions, 36 Cambridge Place, Cambridge, CB2 1NS.
paulb@cygnus.co.uk
Phone: 01223 323 423 Fax: 01223 323 723
Please send your resume by email, fax or post to the above address.
Or simply phone for a chat to learn more about the positions.

CYGNUS SOLUTIONS

RDBMS

UNIX/SYBASE
C.London – £32k + Benefits
We require an Analyst Programmer for the financial sector, based in Piccadilly. Working as part of a small team on development and maintenance of the in-house trading system, using Sybase and ideally Powerbuilder, under UNIX. You will assist Fund Managers in identifying system requirements and you will act as project manager for minor projects.
Ref: RC/1

UNIX/INGRES
C.London – To £28k
Our client, a prestigious software house is searching for Senior Analyst Programmers with good Ingres and UNIX skills. The ideal candidate will also have a good understanding of a recognised methodology. Formal training and the opportunity to develop you own career path, together with an excellent benefits package are some of the attractions!
Ref: PP/2

UNIX/SYBASE/POWERBUILDER
West London – To £35k
A world leader in state of the art technology, requires a number of senior development professionals with excellent user facing, interpersonal and presentation skills. You will be responsible for the development, implementation and support on a range of Client Server Systems. Experience of Sybase and/or Powerbuilder under UNIX would be ideal, although cross training from major relational databases will be provided.
Ref: KB/3

UNIX DEVELOPMENT

UNIX KERNEL
Herts – To £35K + excellent bens
Our client is seeking a highly skilled UNIX engineer with strong kernel/device driver and 'C' programming experience. The successful candidate will join a highly skilled team involved in technically varied and demanding work. This position will particularly appeal to versatile candidates who enjoy problem solving and challenge.
Ref: LC/4

C++/REAL-TIME
Herts – To £30k
Our client is predominantly a developer of software and a supplier of computer systems and associated support services for retail applications. Candidates should have experience of real-time applications and either two years of 'C'/C++ under UNIX or Visual/Borland C++ in a Windows environment, preferably using the class libraries. You must be able to work well within a team.
Ref: CP/5

C++/UNIX
Herts – To £35k
These positions will appeal to candidates interested in developing systems for the on-line video market. You must have excellent academic qualifications and at least two years C++ under UNIX development experience. Additional experience of the Apple Macintosh, Graphics or network design, whilst not essential, will be of interest. Good communications skills, both written and verbal are essential, as is the ability to keep learning the latest technologies. Excellent career opportunities on offer.
Ref: PP/6

DEVICE DRIVERS
Berks – To £35k
Exceptional projects with a leading worldwide software company. Opportunities for software engineers with a first class background in systems level and UNIX device driver development. Working within the telecommunications sector, any previous experience of this area will be of great interest.
Ref: JK/7

'C'/UNIX DEVELOPER
Berks – £22k
Rapidly expanding international company supplying state of the art Collection Systems, requires an experienced individual to program, install and support their systems - applications training will be provided. You will have a minimum of two years programming experience using 'C' in a UNIX environment. Windows NT experience would be advantageous.
Ref: KB/8

EMBEDDED C/C++
Berks/Avon – £15k - £30k
A range of Software Engineers are required with a minimum of one year's embedded 'C' programming skills to develop software for a variety of projects. Development will be under a motorola 68000 operating system on a PC host. Any UNIX skills would be advantageous. The successful applicants will be working for one of the UK's leading Software Houses which currently is expanding into the European markets.
Ref: DE/9

C++/NT/UNIX

C++/NT/UNIX
Surrey – to £40k
Our client is a leading developer of a renowned trading platform for financial institutions. The platform is widely used within the City of London and Wall Street. The continued success of the company in both the UK and US has led to a need for experienced software engineers.

WINDOWS NT
Software Engineers
Experienced developers are required with a minimum of one year's Visual C++/MFC experience. Candidates should demonstrate first rate technical knowledge and a good academic background. Any experience of developing real-time systems or using OLE would be advantageous.

UNIX
Software Engineers
Candidates must possess excellent C++ programming skills on any UNIX platform. Any communications or networking skills would be considered a plus, along with any previous experience of real-time development.
A financial background is not necessarily required as full training will be provided.
Ref:JK/10

Logistix Recruitment Limited, Lamb House, Church Street, Chiswick Mall, London W4 2PD

Logistix

Fax: 0181 742 3061 E-mail: logistix@atlas.co.uk
Telephone: 0181 742 3060

RECRUITMENT

Internet meme

Drug dealers

Refer to their clients as 'users'.

'The first one's free!'

Have important South-East Asian connections (to help move the stuff).

Strange jargon: 'Stick', 'Rock', 'Dime bag', 'E'.

Realise that there's tons of cash in the 14 to 25-year-old market.

Job is assisted by the industry's producing newer, more potent mixes.

Often seen in the company of pimps and hustlers.

Their product causes unhealthy addictions.

Do your job well, and you can sleep with sexy movie stars who depend on you.

Software developers

Refer to their clients as 'users'.

'Download a free trial version...'

Have important South-East Asian connections (to help debug the code).

Strange jargon: 'SCSI', 'RTFM', 'Java', 'ISDN'.

Realise that there's tons of cash in the 14 to 25-year-old market.

Job is assisted by industry's producing newer, faster machines.

Often seen in the company of marketing people and venture capitalists.

DOOM. Quake. SimCity. Duke Nukem. 'Nuff said.

Damn! Damn! DAMN!!!

Stob ProPak 3

Ms Stob has got into the component publishing business.

It's here! It's kool! It's Stob ProPak 3!

Stob ProPak 3 is much more than an upgrade to Stob ProPak 2! Traditional yet innovative, proven but leading edge, and definitely not cobbled together from a few shareware bits and pieces that we found on CompuServe, Stob ProPak 3 contains over twenty[1] components that will make writing your programs so easy that you'll wonder that it is worth paying you to do anything. And for just a few pounds extra, Stob ProPak Pro 3 contains many[2] more!

- **Visual Basic programmers - Boost your productivity and efficiency by producing applications that would make a fruit machine blush!**
- **Internet Explorer users - since all our controls are based on Microsoft's ActiveX® technology, they can be downloaded across the Internet or your corporate intranet and used to enhance and improve your HTML pages. At least, that's what John read on Microsoft's Web site.**
- **C++ and Delphi programmers!**

Just some of our controls:

StobCal calendar component – With 12 attractive full color views of San Francisco and the Bay Area, entering dates becomes a pleasure, not a chore!

StobUDP UDP component – Enhance the power of the Internet (not forgetting your corporate intranet) with this powerful UDP component. Simply drag and drop this component onto a form to add more UDP to applications suffering from UDP deficiency – it's as simple as that! You need never buy another UDP component in your life!

StobIt progress bar – StobIt progress bar does away with those tricky calculations that are hard to code and waste your valuable time. Simply set the Duration property to the length of time you think you need, and that's it! Features an 'installation package' mode, when the bar races to 81% in one second, then stops dead, occasionally running the hard disk to prevent the user from killing the program.

StobWrite component – We've taken the Microsoft Word concept, and stripped it of all those bulky extras that you don't need, to produce a word processor component that will fit into any application. Based around a file format supported by literally more word processors than any other, StobWrite can read NotePad files *directly*.

StobRedLED Red LED display component – This realistic-looking LED control can be switched 'on' or 'off' just as required. Great for emulating modem status lights. This control alone justifies the cost of the kit.

StobBlueLED Blue LED display component – Less useful for emulating modem status lights.

StobClue 'Tip o' the day' dialog – Comes free with two hundred practical statements that you will find really make sense in the real world. 'Place you chair near your PC where you can reach the keyboard' 'The function keys are the ones at the top with 'F' and a number on them' 'Don't pull out the mouse cord while you are using the mouse as it will stop working and you may damage your equipment'!

StobTFTP TFTP component – Need some TFTP for your Internet (not forgetting your corporate intranet) solution? We have plenty! TFTP convenience at the click of a mouse, whenever and wherever you want it.

StobGothicListBox list box – At last! A list box with a Gothic font *built in*. Now you can include as many Gothic list boxes as you like in your applications, without worrying about shipping separate Gothic fonts. Stob ProPak Pro only.

StobMultiMediaMegaMovieMaker multimedia component – Enhance and enliven database applications with simple animations, or little postage-stamp sized films of Bill Gates delivering speeches about the future of Windows.

StobEazyPeazyScreenSave component – Write you own screen savers in seconds! Because, God knows, there can never be too many screen savers in the world. Especially not of the kind that have been written in a few seconds.

StobTelnetServe Telnet Serving component – Have you ever been caught short when you needed a Telnet Server? Well from today that will never happen again because, because,.. oh stuff it. I give up. This is yet another incomprehensible TCP/IP thing. Ok?

StobGrid Grid component – Although superficially just like all the other grid components you have knocking around your hard disk, StobGrid is actually *very different*.

StobBlagl component – Comes out sort of like a grey car radio panel when dropped on a form. Its huge property list includes integers *Xmush*, *Ymush* and *Zmush*, strings *StatelyM* and *PortlyM* and an enumerated type property *Rank* taking values 'dog', 'feet' and 'Chelmsford'.

And many more.

Complete with documentation in multiple formats: README.TXT, README.WRI and README.DOC.

All this for just £299 + VAT + Shipping + more VAT. Stob ProPak Pro 3 costs £199 extra + more VAT + more Shipping.

Don't forget to download our demo from the broken link at http://www.stob.personal.demon.co.uk/download/. Or fill in the form on page 120 of this magazine. We won't send you a demo; it's just that we have found that we can make money by selling programmers' names to direct marketing organisations.

1: 21 components.
2: 2.

MORE DEVELOPERS PROTECT.

HASP Packs More Into Less.

Based on a full-custom ASIC utilizing 2500-gate, 1.5-micron E^2 technology, HASP packs the most advanced protection into the smallest key in the world.

HASP® PROTECTS MORE.

Stand 718
The Essential Development Tool
software development 97
24-26 JUNE 1997 NEC BIRMINGHAM

NSTL Study Rates HASP No.1!

A recent test conducted by the National Software Testing Labs, the world's foremost independent lab, compared the flagship products of leading software protection vendors.* The result? HASP was rated the clear overall winner - and number one in all the major comparison categories.

NSTL TEST RESULTS, OCTOBER 1995†

Scoring Category	Aladdin HASP	Rainbow Sentinel
Security	9.3	6.3
Ease of Learning	9.1	7.1
Ease of Use	8.3	7.2
Versatility/Features	10	8.7
Compatibility	6.7	6.5
Speed of API Calls	0.9	1.2
Final Score	**8.5**	6.5

*For a full copy of the NSTL report, contact your local HASP distributor.

These days, more and more developers are choosing to protect their software against piracy. They're protecting more products, on more platforms, with better protection – and selling more as a result.

And more of these developers are protecting with HASP. Why? Because HASP offers more security, more reliability and more features than any other product on the market.

HASP supports the most advanced platforms, including all Windows 32/16-bit environments, OS/2, DOS, Mac, Power Mac, NEC, UNIX and LANs.

To learn more about how you can protect better – and sell more – call now to order your HASP Developer's Kit.

Grow With Aladdin!

The fastest growing company in the industry, with over 4 million keys sold to 20 thousand developers worldwide, Aladdin is setting the standard for software security today.

yes It runs with NetWare
Microsoft Windows Compatible
MacOS

01753 622266
http://www.aks.com

ALADDIN
The Professional's Choice

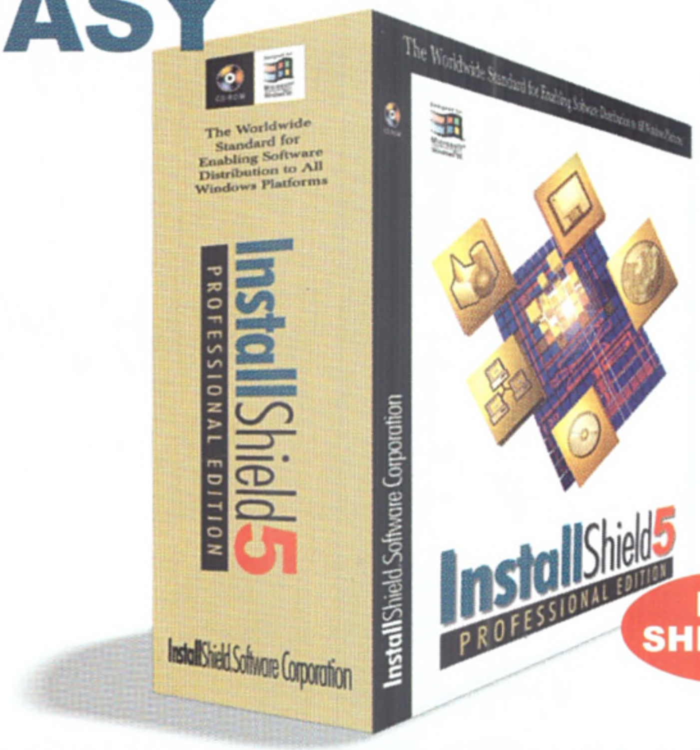
United Kingdom **Aladdin Knowledge Systems UK Ltd.** Tel: +44 1753-622266, Fax: +44 1753-622262, E-mail: sales@aldn.co.uk
North America **Aladdin Knowledge Systems Inc.** Tel: (800) 223 4277, 212-564 5678, Fax: 212-564 3377, E-mail: hasp.sales@us.aks.com
Int'l Office **Aladdin Knowledge Systems Ltd.** Tel: +972-3-636 2222, Fax: +972-3-537 5796, E-mail: hasp.sales@aks.com
Germany **FAST Security AG** Tel: +49 89 89 42 21-37, Fax: +49 89 89 42 21-40, E-mail: info@fast-ag.de
Japan **Aladdin Japan Co., Ltd.** Tel: +81 426-60 7191, Fax: +81 426-60 7194, E-mail: aladdinj@po.iijnet.or.jp

■ **Aladdin Benelux** 024 641 9777 ■ **Aladdin Russia** 095 923 0588 ■ **Australia** Conlab 3 98985685 ■ **Chile** Micrologica 2 222 1388 ■ **China** Shanghai LRI 021 6437 7828 ■ **Czech** Atlas 2 766085 ■ **Denmark** Berendsen 39 577316 ■ **Egypt** Zeineldein 2 3604632 ■ **Finland** ID-Systems 0 870 3520 ■ **France** 1 4085 9885 ■ **Greece** Unibrain 1 6756320 ■ **Hong Kong** Hastings 02 5484629 ■ **India** Solution 11 2148254 ■ **Italy** Partner Data 2 26147380 ■ **Korea** Dae-A 2 848 4481 ■ **Mexico** SiSoft 5 2087472 ■ **New Zealand** Training 4 5666014 ■ **Poland** Systherm 61 48027 ■ **Portugal** Futurmatica 1 4116269 ■ **Romania** Interactiv 64 153112 ■ **Singapore** ITR 65 5666788 ■ **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 0635

CIRCLE NO. 268

maximum **POWER**...

has never been so **EASY**



NOW SHIPPING!

NEW features

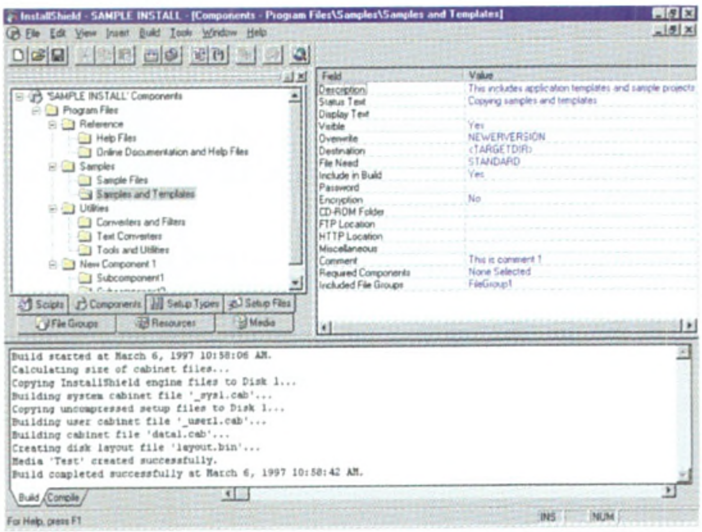
- Completely new integrated 32-bit visual development
- Powerful, fully integrated and automated media builder
- Drag-and –drop visual file layout allows for easy modification and updates
- Full power of InstallScript for customized installation; color syntax editing
- Visual editors for File properties, Groups, Components and Setup Types
- Support for integrating .AVI video, WAVE/MIDI sound, 256-color images
- Support for integrated global software distribution

Introducing InstallShield5 Professional

THE COMPLETELY *NEW* EDITION OF AN INDUSTRY-STANDARD.

InstalShield5 Professional combines an easy-to-use visual development environment with the reliability and horsepower of proven InstallShield technology. InstallShield5 Wizards speed you through some of the tasks of installation development, while its visual scripting capabilities ensure you have complete control over the details of your application's setup. Producing multiple builds and multiple installations is simpler than ever before.

- Project Wizard gets you up and running and establishes a framework for an installation project
- Function Wizard guides function use and auto-inserts complete function statements



- MediaBuild Wizard handles everything needed to create builds for various media on the fly
- Color Script Editor visually organizes the elements of installation scripts by color-coding syntax

QBS Software Ltd.
Tel: 0181 9568000
Fax: 0181 9568010
WWW: www.qbss.com
E-Mail: orders@qbss.co.uk

Soft/Export, Ltd.
UK Free Fone: 0800 973098
Fax: +353 1 294 2197
E-Mail: installshield5@softexport.com

InstallShield Software Corporation

Download an evaluation today! www.installshield-europe.com • www.installshield.com

CIRCLE NO. 269