Get the message?
Working with SMS

More tools for Windows
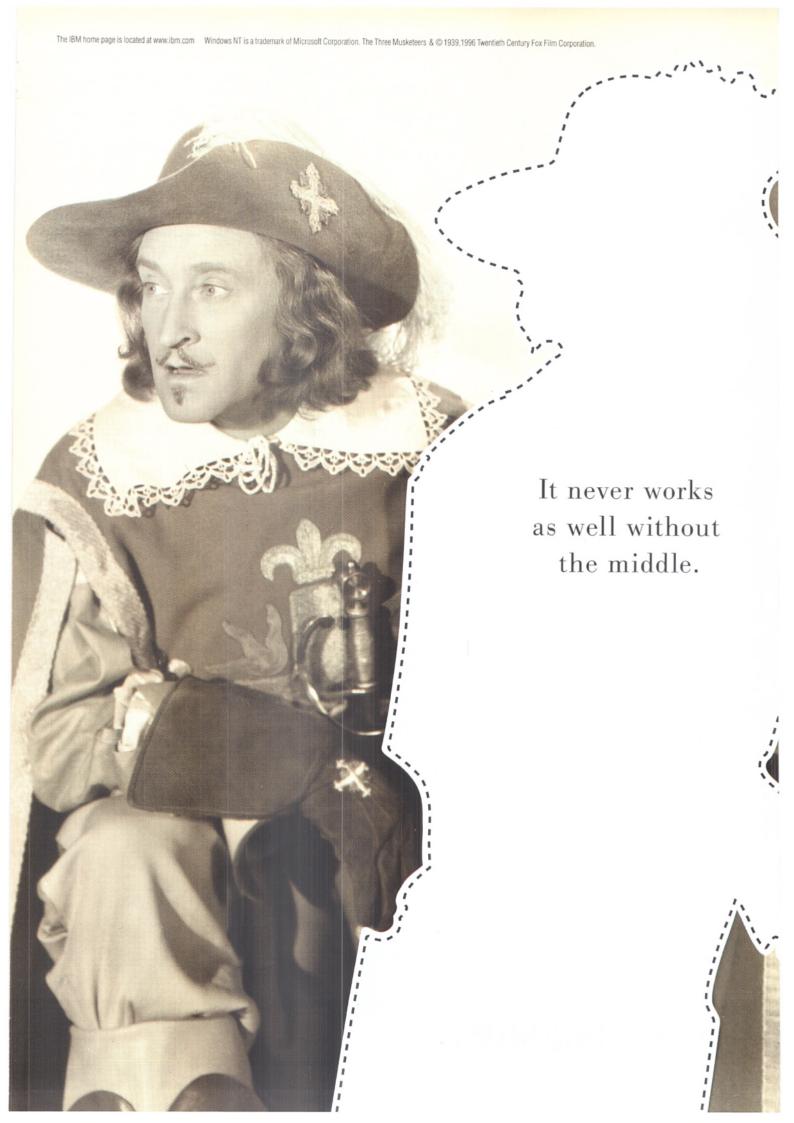
Recycling Internet protocols

A GUI builder for Java Workshop

Protect your files

C++ templates: The truce

It never works
as well without
the middle.

Presenting IBM Software Servers. Now the client/server picture is complete.

It's amazing how well client/server computing can work. Especially if you put something between your clients and servers to help them communicate better. That's why we created the seven new IBM Software Servers.

They support all major clients and run on the platform of your choice: OS/2, Windows NT or AIX. And best of all, they're designed to work together seamlessly, so your business runs more smoothly. All for one and one for all.

For your complimentary Preview kit, please call us on 0800 96 00 31 or visit www.software.ibm.com/info/AD201/

· · ·

Lotus Notes
The leader in messaging and groupware.

Database Server
Helps workgroups and companies share important data.

Internet Connection Server
A solid foundation for building Web-based business.

Communications Server
A powerful gateway for connecting diverse systems.

Transaction Server
A flexible infrastructure for mission-critical applications.

Directory and Security Server
Manages companywide users and resources from one desk.

Systems Management Server
Automates management of multi-vendor distributed systems.

For further information telephone us at 0800 96 00 31 or fax: +(45) 48 10 19 99, or send this coupon to IBM United Kingdom Limited, Freepost ANG 5023, Winterhill, Milton Keynes MK6 1YL.

☐ Yes, I would like to receive additional information.

Name:_____ Surname: _____

Job Title:_____

Company:_____

Address:_____

_____

City:_____ Postcode: _____

CIE2A800

Country:_____

Telephone:_____

Fax:_____

e-mail:_____

IBM

Solutions for a small planet

EXE

The Software Developers' Magazine

EXE online

http://www.exe.co.uk

SoapFlakes

J-Word

Of the 32 news stories printed in last month's *EXE* magazine, eight, including the lead story, concerned products for Java – Latté, J++, Visual Age Java, WebXpresso, WinGen for Java, Eleven, Amzi Logic Server's Java Classes and the O_2 Java Suite – and another two – Cactus, Intra – contrived to mention Java even though they were really about something else. This month, the fever has spread to the comment section: over the page you will find Jules May questioning Java's panacea status, and of course you now know about me. This is a hell of a lot of coverage for a programming language which in real life is actually mostly used to implement Tetris, and to display animations rather slowly.

I'm not alone in marvelling. Gordon Eubanks is head of Symantec, a company which pioneered Java development environments with Café ('pioneer' in this context means to have done something more than three weeks ago). Here he is, quoted in the September edition of *PCW*: 'Do I think [Java] is over-hyped? Absolutely. Do we over-hype it? Probably.' If Eubanks, a distinctly interested party, is prepared to admit that things have got out of hand, then one wonders just how out of control things are.

Another indicator that our bandwagon has lost its brake fluid is the behaviour of Microsoft. OS independence and machine portability are as welcome at Microsoft as British gelatine in a German blancmange. The companies one associates with the phrase 'Microsoft's most bitter rivals' is surely the same set as those who have backed Java the hardest: Netscape, Sun, and, Oracle. Microsoft would have surely killed Java if it could, and may do so yet if it can. But right now it has been compelled to climb aboard to avoid being knocked over, and is merrily creating IDEs, embedding JVMs in Windows and so on.

What especially baffles me about this fever is that the majority of people who are getting feverish don't seem to understand what Java is. To cite a couple of personal experiences from the last fortnight: someone rang me up the other day and demanded 'a Java solution'. (I should explain that I work for a concern which sells compilers and add-ons.) I offered him Café. 'I want to do television pictures across the Internet, with sound and everything', he said. 'Will this help me to that?' I opined that it probably wouldn't. 'So don't bother me with this rubbish', he said angrily, 'just sell me a **solution.**' The conversation ended five minutes later in mutual frustration – I swear that the caller believed I was holding out on him.

The second incident concerned a consultant who came in to ask a series of impressively fatuous questions about a certain Visual Development Environment he was evaluating. At the end of three hours, during which I had explained in detail about the tool's language, compiler, targeted platforms and so on, this man asks 'And does it support Java?' He didn't know what Java did, or why it was relevant (or not) to the development his company was contemplating, or why it didn't make sense to ask about Java when it had just been explained at length that this was a Pascal tool which generated native Intel code: none of this concerned him. He was a man who used fashion as his development methodology, so he needed Java.

I've played with Java, and I have considerable admiration for a number of aspects of its design. But the current climate is ridiculous. Technical issues are being completely obscured by clouds of hot air. One of Bjarne Stroustrup's underrated skills, during the first days of his fame, was the way he always kept C++'s advantages in perspective. Would that Java had some cooler heads at the helm.

Will Watts

Merlin: RIP OS/2?

The OS/2 newsgroups seem to have gone rather quiet after the release of Merlin, the beta programme for the much-hyped OS/2 Warp 4.0. Team OS/2-ers who had quivered in anticipation for so long appear to have been disappointed. Having used Merlin, I don't entirely blame them.

Let's face it: OS/2 needs a miracle to save it. Among Merlin's 'killer' features are a revamped user-interface, cute coloured tabs in dialog boxes, and – this is the biggie – voice navigation and dictation. As long as you've got 32 MB+ of RAM and a Pentium 133, that is. So far, it steadfastly refuses to understand me unless I speak in an American accent. As for the install program – 'unfriendly', 'inhospitable', and 'please may I abort now' are phrases which come to mind. It's all so frustating because, once you set it up properly, OS/2 is a tremendously good operating system.

IBM's answer to the dearth of native OS/2 software has been to get into Java in a big way: Merlin will run Java from the desktop. Will the promised deluge of Java applications appear? Not if Microsoft has anything to do with it – and don't forget that it was the cold shoulder from Microsoft, in whose labs OS/2 originated, which murdered the native OS/2 software market.

On a recent visit to the Microsoft campus at Redmond, I came across a plaza paved with plaques commemorating all the software that Bill and Co. have ever released (and very extensive it is, too). There, hidden away in a corner with other embarrassments such as Microsoft Delta (anyone remember that?) and the disastrous Bob, was the plaque for Microsoft OS/2. Sad to say, this, and the apparently immortal stalwarts of Team OS/2, may soon be the last memorials to a once-great OS.

Neil Hewitt (no longer Team OS/2)

Mayhem!

The sound is faint at first – a distant, rolling thunder on the horizon. Almost imperceptibly at first, the volume grows, but now it's growing louder by the second. The ground begins shaking, and suddenly, a massive wall of movement swings into view, and passes within a few feet of me. It's a stampede, diverting occasionally to draw new members out of groups it passes, swelling the already crowded ranks, and crushing all in its path with its headlong inevitability. I reach in and pull out one of the sweating, panting bodies. I can see him clearly, now; it's a programmer.

'Let me go!' he wails, 'I'm being left behind'. I look where the crowd is heading. The Golden City on the opposite horizon, the land of milk and honey, where nobody is poor, the nerds are kings, coffee is free, and the National Health system isn't completely bankrupt. I let him go, and as he rejoins the herd, I can hear behind the rhythmic sound of a million sneaker-shod feet the constant chant 'Gotta go, gotta go, gotta go to Java.'

Actually, it's not really a shining city. It's not much of a city at all. It's not even much of a town. If truth be told, it's not much more than

a planning application, but that's where they're all going.

I eventually catch up with the herd where it has stopped at a watering hole. Java is all anybody talks about. 'Platform independent' repeats one, endlessly, while another replies 'Standardised language'. 'Applets, applets for all!' screams yet another. One programmer is sitting in a corner, talking to himself: 'I've been to C, and I learned my way about. I've been to C++, and it was just like home, but different, and I still saw most of it – I think. Java will be just like those places, won't it?' I decide to collar him.
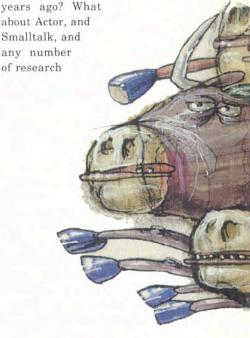
'How about viruses?' I ask. 'Oh, I never got sick in C++, only a little tired. Nobody will get sick in Java.'

Gentlemen! Listen to me! These mantras you're repeating; don't they strike you as a teensy bit familiar? I mean, we all knew C wasn't working, but that's why they invented C++. Platform independent it certainly was – it was just as intractable whatever the platform. Ada, and other more ignored solutions had the same problem. C++ sold because it looked just like C (one city map

looks much like another, until you try to use the wrong one to get somewhere). What about VBXs and OCXs; pre-packaged functions that anyone could use, only nobody did because the packaging was too bulky and they wouldn't fit through the doorways. What about p-code, which was precisely a mechanism for moving bits of code in a platform-independent manner?

But Java won't be like those, honest, Java is going to *work*.

Well, so it should; it's interpreted, and it straps onto a display manager so the code runs on the display machine. Forgive me if I'm remembering wrong, but only three years ago, weren't you telling me that display and core code should be separated? Yet here you are trying to put them back together again. Oh yes, I remember, separating the parts made porting easier, and the porting is done by the language this time. So what about all the languages which were trying to do exactly this ten years ago? What about Actor, and Smalltalk, and any number of research

projects? Oh yes, I remember again; back then you had to be Microsoft to achieve anything worthwhile. Now you have to be a consortium comprising Microsoft, Apple, Novell, IBM, and Uncle Tom Cobbeley and all. That's the only way that everybody can agree on what they want to achieve. Just like Corba. Anybody remember Corba?

Java is different. Java is not a big consortium. Java is Sun, and Sun is just doing what it wants. Whatever Sun says is right, is right.

But, back in the bad old days, when every machine was different, and ten copies was good circulation for a program, everybody did what they wanted, and nothing worked. When everybody started working on similar things, like Basic and Pascal, everybody did it slightly differently, and things were terrible. Standardisation is a good thing. You can't have comms without standardisation.

Java doesn't have those problems. Java has mechanisms to tell you what kind of platform you're running on.

So did the old printer drivers. So does C, with `#if`. Are you seriously suggesting that all the problems of portability can be solved with `#if`?
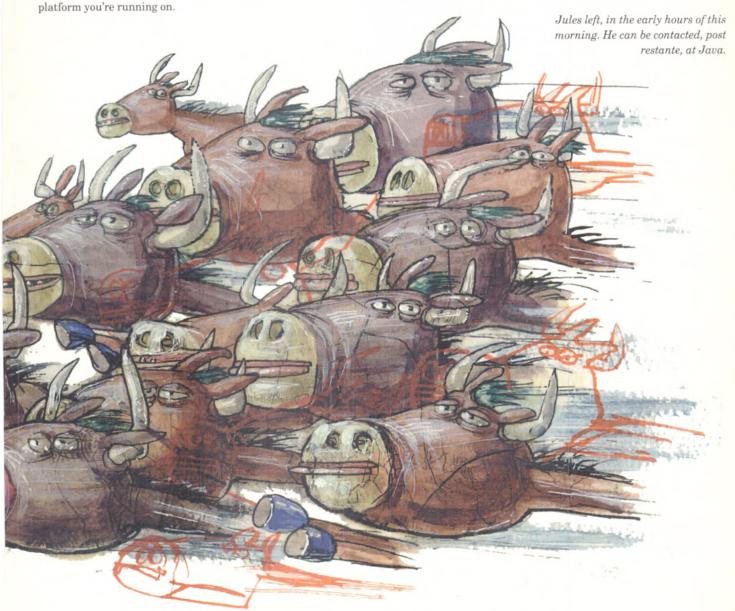
Suddenly, there's a shout. Over there, on the horizon, there's a new tree, perhaps a new oasis. At any rate, it's some other stopping point on the way to the Golden City. The chant starts again; 'gotta go, gotta go', and the massed herds are milling again, pawing the ground, getting ready to be off on their gaderene rush. 'Wait' I shout, 'There's nothing there!', but it falls on deaf ears. They're off again.

There's nothing there. No platform independence, no easy programming, no standardisation or agreement. Every member of that herd has a different idea of what the promised land will be like. When they get there, all they will find is a kit of parts, and what they will erect for themselves will be the same, hopeless, teetering muddle that they built on all the other sites.

So, I leave them to their dreams, and turn back the way they've come. But there's nothing there either; not a tree, not a bush, not a house has been left standing. It's all been trampled into the dirt. That way is only desert.

They might make it. This time it might work. When they get where they're going, perhaps there will be plans there that everybody can work together to construct. Perhaps there will be someone there who can quell the infighting and motivate everybody together. Perhaps somebody will be able to prevent thrusting new ideas from being trampled by sheer weight of numbers the moment they poke a shoot out of the ground. Perhaps, this time, the dream can be realised. There is something there; hope. And that's more than can be said for the land they've left.

But there's nothing really there. There's just nowhere else to go. If I'm going to catch up, I'd better start running now. Goodbye, gentle reader, gotta go. ∎

Jules left, in the early hours of this morning. He can be contacted, post restante, at Java.

Visual J++ Activates Java

Known until recently as Jakarta, Microsoft's Visual J++ (VJ++) is a Java development system based around the Developer Studio environment pioneered by Visual C++ 4.0. The package comes with wizards for building ActiveX components in Java which can then be run in a Microsoft browser. ActiveX controls are supposed to be able to run in any OLE-enabed application such as Word or Excel, and to this end Microsoft has written a version of the Java Virual Machine (JVM) implemented as an ActiveX control which can be wrapped around Java ActiveX controls and ordinary Java applets to allow them to run in non-browser environments.

While Microsoft is at pains to point out that developers can easily use Visual J++ to write completely cross-platform standard Java applets, it is clear that the company wants to make Java/ActiveX a standard part of the Windows developer's arsenal along with C++/MFC. With plans to port ActiveX to both Macintosh and Unix, the need for a cross-platform control standard could push Java to the forefront as the Windows development language of the future.

◗ Announced availability in October.
◗ *Microsoft: 01734 270001*

◗ No price has been set yet.
◗ *Fax: 01734 720002*

Oxford University to host 1996 Developer Academy

After two successful years at Cambridge, the 1996 Visual Tools Developer Academy – which this year is supported by *EXE* – will take place at Keble College, Oxford University between 9-12 September. The conference focuses solely on practical visual development issues and features 48 sessions in four streams over the first three days, while the fourth day will be given over to in-depth technical workshops and tutorials.

An impressive range of speakers from the visual programming world is lined up, including conference chairman Peter J Morris of consultants The Mandlebrot Set (author of the well-respected *Windows – Advanced Programming and Design*), and Tim O'Pry, president of MicroHelp (of *Uninstaller* fame). As an added bonus, all the speakers will be available on an individual basis throughout the conference.

Topics to be covered this year include: Visual C++, MFC, OLE, Client/Server, Visual Basic, WWW, process management, and optimisation and porting.

The conference is residential with accomodation for delegates provided on-site and a range of social activities to be arranged. In addition, delegates will be able to sit the Drake tests on Visual C++, VB, and MFC which will enable them to become a Microsoft Certified Professional.

Each conference place costs £875 which includes all documentation, full-board accomodation for three nights, and all social functions. The cost of the final tutorial day is extra.

◗ *For more information or to reserve a place, call Emma Hilditch*
◗ *Interactive Information Services on 0181 541 5040*   ◗ *Fax 0181 974 5188*

Tools.h++ gets zApped by Rogue Wave

Version 3.0 of zApp Developer's Suite, the cross-platform development system from Rogue Wave, sports an extended set of user interface controls and integration with the Tools.h++ 7.0 foundation class library. Although the traditional zApp function base is still supported, Rogue Wave believes that developers will get the best out of the system if they move over to using the Tools.h++ function set. The additional controls include tabbed dialogs, progress bars, tree control, and tooltips: most of these have been included to guarantee cross-platform compatibility with Windows 95. zApp applications can work with OLE drag and drop, and manipulate the system registry. There is no direct OLE support for any of the other platforms, however.

Price is £1650 per single license, available on Windows 3.x, 95 & NT, OS/2, and Unix (HP-UX, RS6000, IRIX, SunOS, Solaris and DEC OSF). zAppDS 3.0 is distributed in the UK by Hypersoft Europe.

◗ *Hypersoft Europe: 01273 834555*

NEWS

Borland hand in hand with Sun

This year's Borland Developers Conference was at Anaheim. It was a packed week of seminars under the hot sun of California. Last year's conference keynote speech featured Gary Wetsel (then President and CEO of Borland). Since his resignation no one has been named yet as a replacement and the conference was opened by Paul Gross, Senior Vice-President of Research and Development. Paul Gross, Zack Urlocker (director of Delphi product management) and Eric Schmidt (Chief Technology Officer of Sun Microsystems) all denied being runners up for the job. You may wonder why a journalist asked this question of the outsider Eric Schmidt? Apart from the fact that he was the first guest keynote speaker on Monday, Sun and Borland do seem to be fighting the same battle: Java, JavaScript, and BAJA. This is the object model used in Latté: the Borland Java Architecture which has been adopted as part of the Java Beans standard by Sun. Paul Gross said that 'BAJA and Java Beans are all about working reuse through components [giving developers] Plug and Play application components'.

Eric Schmidt described Java Beans as 'the final missing link [between] applets to real applications'. The future of Sun and Borland is certainly becoming more and more dependant on the success of Java, and Schmidt isn't in any doubt about it. He told us during his keynote that 'Java is buzzword compliant and network aware, it will survive for a long time, 20 to 30 years at least'!

▶ Borland: *http://www.borland.com* ▶ Borland UK: 01734 320022

Il Latté, e' pronto?

Most announcements made at the conference were about products that will be released later this year or at the beginning of next year. We didn't get the usual stack of CDs packed with applications: just one CD with a copy of the presentations. All product lines were covered in separate tracks but the stars of the show were definitely Latté and IntraBuilder (see *EXE* August '96 pp. 7 & 8). IntraBuilder was put on the Web right at the end of the conference. Those are the products Borland *was* willing to talk about.

After *EXE* saw a brief mention of the codename *Pronto* in a leaked document, we wanted to know more. Borland is not apparently willing to make much comment on this until December. Hence all that follows is mostly unconfirmed. To start with, two product groups seem to have something to do with Pronto: the C++ and Delphi ones. The goal of Pronto is to make BC++'s compilation as fast as Delphi's: this was as much as Zack Urlocker was ready to admit. The best way to reduce compilation time is to reduce the amount of compilation! To do this, Pronto might use the same technique as the next release of Delphi, ie compile most of the common code found in applications into DLLs which need only be compiled and installed once. Borland calls these extended DLLs *packages*. They are signed with a CRC and all methods are exported (no more Complib).

Another unacknowledged goal of Pronto which would make sense would be to unify the component/object model across languages. With the BC++ update (available late August), C++ programs can call Delphi forms and this is the first step in a more complete interoperability between languages. For communication, Borland seems to have already chosen the COM interface as a standard. As regards to the object model, the situation is less clear: Delphi has VCLs and Latté has BAJA.

From this it seems that next year Borland will have Ivory (a new Delphi), Latté and probably Pronto (or BC++ 6.0). The next big question is, when will Latté and Pronto be available? One rumour heard recently was that Latté will be in the shops by November. Pat Vermont, Group Product Manager Internet Product Group, said that a beta would be on Borland's web site in '8 to 12 weeks and that it would ship at the end of the year'. But one Borland employee (who obviously hadn't been briefed about the secrecy of Pronto) told us that Pronto might appear before Latté and that neither would be here this year whereas the official line is that Latté should ship in 1996 and Pronto doesn't exist...

Client/server and the Internet – together at last!

The second keynote at the Borland Developers Conference was by one-time *EXE* contributor Christine Comaford (*EXE* April '96). For anyone who has not already seen her in public, she was as provocative as usual. She started with this dialog alleged to have taken place in the beginning of 1995 – Bill Gates: 'With software we can do anything...'; Christine Comaford: 'Show us Windows 95'!

Here's one piece of advice grabbed from her fast-paced presentation: when developing for intranets/Internet do *not* start from scratch, reuse any client/server work that you have done. And two tips: on a Web site 'complete a task within 3 jumps – don't make users backtrack to the home page to complete a task'; 'Plan on three to five content developers for each software developer'.

▶ *http://www.christine.com*

NT 4.0 brings single Win32 code-base nearer

Developer schizophrenia about writing for both Windows NT and 95 is almost over with the release of Windows NT 4.0. The system, which has been in development for over a year and went through two wide betas, incorporates the Windows 95 shell and supports a number of additional Win95 APIs previously unavailable in NT. Some support is still missing, in particular true Plug-and-Play, but it does detect new devices at startup and, if it knows about them, installs the drivers relatively transparently.

The shell implementation is complete: those who have used the Shell Update Preview for NT 3.51 or NT 4.0 beta 1 will know that quite a lot of the functionality of the Windows 95 shell was missing from these versions, and the two shells were sufficiently different from a programming point of view to break Windows 95 shell extensions. With the full shell API set implemented and total look-and-feel integration it should be possible to write a large proportion of new applications to a single Win32 code-base and feel confident that they will run on both NT and Windows 95 without change.

Internet Information Server 2.0 is included in the package for both Workstation and Server editions, but one promised element missing is Network OLE support. Microsoft apparently cannot get this to work yet and rather than delay NT 4.0 any further has dropped it from this release.

According to Bill Gates, NT 5.0 is already in development in the labs at Redmond and may incorporate more of the 'Cairo' technologies. While staying tight-lipped about the product at the moment, Gates did confirm that in the next release at least Windows NT and Windows 9*x* will remain separate products and that ensuring API compatibility between the products would be a major concern for the company.

▶ NT Workstation 4.0 is priced at £250
▶ NT Server 4.0 is priced from around £500.
▶ *Microsoft: 01734 270001*
▶ *Fax: 01734 720002*

Software Developer's Challenge to benefit NSPCC

Rhea International has announced that this year's Software Developer's Challenge, to be held at SoftDev/WebDev'96 on 27-28 November, will benefit the NSPCC. Open to all entrants, the Challenge will take place over one day; visitors will be encouraged to observe the competitors at work. While the choice of development tool will be left open to the individual competitor, the specification for the application will be provided by the NSPCC (although it has yet to be set).

This year's competition will be split into several levels to accommodate both professional developers and amateur programmers alike, with a running commentary to be provided to visitors by specialist monitors who will observe the development effort.

Both the Challenge and the SoftDev/WebDev'96 Show are supported by *EXE*, and developers who would have entered *EXE*'s own Challenge – which unfortunately had to be cancelled earlier this year – are encouraged to take part. To enter, call Jon Howell at organisers Interactive Exhibitions.

▶ *Interactive Exhibitions 0181 541 5040*

Powerbuilder joins the Web bandwagon with web.pb

Boldly going where only thirty or forty companies have gone before, Powersoft has announced web.pb, a technology which extends its PowerBuilder product to a full Web client/server development environment. Using web.pb, existing PowerBuilder applications can be retro-fitted for the Web, or new purpose-built applications can be written which take advantage of the browser client. As well as standard CGI, web.pb will support Netscape's NSAPI and Microsoft's ISAPI.

On the opposite side of the fence is the PowerBuilder Window (sic) Plug-In, a Netscape-compatible plug-in which completes the circuit and makes it possible to use PowerBuilder applications within the browser environment.

The tools are in beta at the moment, but will be included in the next release of PowerBuilder. They will form part of Sybase's plan to Web-enable their entire range of client/server products, web.works. Beta versions can be downloaded free of charge from Sybase's Web site, while final versions are expected 'within the third quarter' – in other words by September.

▶ *Sybase Powersoft WWW site URL: http://www.powersoft.com* ▶ *Tel 01628 597100*

"I started running OS/2 Warp Server on all my networks and guess what?

My rainy days are over."

Umbrella It was time to consider consolidating his mixed environment network onto a single operating system, so Steve Conaway, Director of Computer Services at the Financial Times, decided to check out the new release of OS/2 Warp Server. In no time at all, Steve was waxing poetic over OS/2 Warp Server's ability to handle blockbuster-sized databases and make Internet and Intranet access a breeze. He was also impressed with all the advanced printing capabilities and management features that simplified the running of both his network and his life. From now on, Steve definitely regards his OS/2 Warp Server as his umbrella network operating system.

To find out what got Steve so excited, call **0800 96 90 45** for your free 60-day evaluation copy of Warp Server, or you can visit our web site at www.software.ibm.com/info/ws031.

What's more, if you buy both Warp Server and Lotus Notes before 31st October, you can save hundreds of pounds. There's no better way to demonstrate fiscal responsibility.

Send this coupon to IBM United Kingdom Limited, Freepost ANG 5023, Winterhill, Milton Keynes MK6 1YL, or fax it to: +(45) 48 10 19 99.

☐ Yes, I would like to receive my free 60-day evaluation copy of Warp Server.

Name: _____

Surname: _____

Job Title: _____
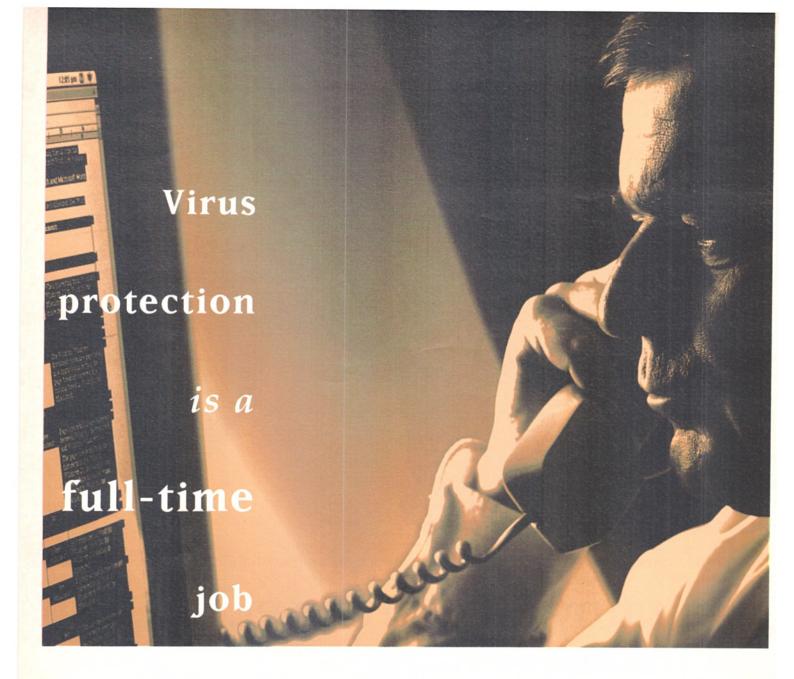
Company: _____

Address: _____

_____

CIWIA800

City: _____   Postcode: _____

Country: _____

Telephone: _____

Fax: _____

e-mail: _____

IBM

Solutions for a small planet

✎ CIRCLE NO. 786

Virus

protection

is a

full-time

job

{ **You**

already

have

one

Thousands of network managers in government, financial institutions and multi-national organisations worldwide rely on **SWEEP** anti-virus products from Sophos.

SWEEP is available for DOS, Windows, Windows 95, Windows NT, NetWare, OpenVMS, OS/2, Banyan Vines and Unix, and features InterCheck technology for client-server protection of networks.

SWEEP is developed by one of the world's most experienced teams of anti-virus experts. The same team also provides 24-hour technical support, 365 days a year.

For more information, contact us today.

S|O|P|H|O|S

DATA SECURITY

Sophos Plc • 21 The Quadrant • Abingdon • OX14 3YS • England
Tel 01235 559933 • Fax 01235 559935
Email info@sophos.com • http://www.sophos.com/

CIRCLE NO. 787

Letters

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

Pgazonga

Exactly ten years ago, I had the luck to find a way to use the Emacs editor on the Multics we had at University Paris 6. Its use was not allowed for students. I can't remember the precise details but I found a reproducible way to crash Emacs. When it crashed it filled the screen with the word 'Pgazonga'. For the past decade I had sleepless nights wondering about the meaning of this word. I recently tried an AltaVista search but didn't have any success. I eventually asked the author of Multics Emacs: do you remember this word? Does it have any meaning? – Ed

Mais, certainement!

This will require some familiarity with Lisp. I doubt it 'filled the screen' with it, it probably mentioned it in an error message.

Lisp supports a facility called `catch/ throw`, which effects a nonlocal transfer of control, from an active environment to a calling environment, aborting all called environments in between. The form which invokes this looks (in Maclisp) like this:

`(throw value foobar)`

and the form

`(catch (do-this here) foobar)`

returns either the value of `(do-this-here)` or the value of `value` when the `throw` above is evaluated someplace within the execution of `(do-this here)`.

The symbol `foobar` (in this case) is known as a 'catch tag', and is used to associate catches with throws.

Now for the specifics:

'Gazonga' is a comic-strip phony-Italian word probably dating from the 30's or 40's. It might have occasionally shown up in my childhood (born '50) in old movies or comics. It has no meaning as far as I know, except it sounds Italian and colourful. Perhaps somebody might say it to accompany a punch in the nose or other crude violent act, the wave of a handkerchief over a magician's hat, or some other situation appropriate to a talismanic utterance. It is what today would be called 'Manga word' in Japan.

In Multics Emacs, the catch tag for 'top level', ie, the command loop, was 'gazonga'. I probably chose this term because it evoked an onomatopoeic sense of a speedy projectile being shot downward, in this case, control, from the deep levels of Emacs execution to the top-level command loop. It should never have been visible to anyone, even extension writers, because the primitive (`command-abort`) was provided to throw it, and it's no one's business who catches it.

Multics Emacs, like its author, is very fond of foreign-language terms and expressions (language being one of my life-long interests), and included a panoply of functions (let alone code comments) borrowing from German, Latin, and French, most of which (but not all!) were invisible to the extension writer (let alone end-user). A documented variable (`der-wahrer-mark`) was even, as can be seen, (inadvertently) in incorrect German syntax (should be 'die wahre').

So the internal function that threw to command level, that `threw gazonga`, as it were, was named `jetteur-des-gazongues`, the latter being what I assumed the reasonable translation of this non-term into French.

Now when the minibuffer was up, and some command caused an abort, ie, called (`command-abort`), this normally would be caught by the minibuffer, so that editing in the minibuffer could continue – the minibuffer's recursive command level would set up a recursive (`catch ... gazonga`) that would trap these aborts.

But in those days it came to pass that this was found not to be adequate – when you type 'control g' to abort out of the minibuffer, you do *not* want it to be caught by the recursive catch. So for this, I invented 'les petites gazongues', the 'p' is for 'petites', and hence 'pgazonga' – the 'petites gazongues' were continued 'smaller' and more agile than the regular ones because they deftly slip through a catcher of 'regular' gazongas, such as the net set up by the minibuffer, and sink all the way down to the bottom, ie, top command level (pardon the upside-down confusion, Multics was highly 'stack oriented', ie, top of stack = more levels into code). So control-G invoked the function `jetteur-des-petites-gazongues` to throw back to top level.

Of course, they were supposed to be caught by top level, and that's how it was all supposed to work. The bug you are somewhat belatedly reporting is almost certainly some esoteric case (during startup, perhaps?) wherein the catch for `petites gazongues` (pgazonga) was not set up, and a throw to that tag failed to find a catch, and hence the error you probably saw.

Have I made myself perfectly clear?

For the definitive paper on Multics Emacs (1979, but in nice HTML), please see *http://www.best.com/~thvv/mepap.html*.

Bernard S. Greenberg
bsg@basistech.com

Jules vs the Rest of the World

Dear Sir,

I have followed the Jules vs the Rest of the World debate with great interest. I can understand why you would rather have correspondence on IT matters but in my experience most programmers I have come across seem to delight in such non-IT related matters which perhaps explains the level of mail you receive on the subject. I wonder if there is a forum somewhere that would welcome such debate?

Anyway, I have two items that may be of interest to your readers. I have recently begun using a shareware installer that was not mentioned in you review in the April '96 issue. The package is called InstallWare and is available from CompuServe or perhaps from the author at 102545,1673@compuserve.com. I have found it to be perfect for the needs of a small installation system and it handles all of the usual features. The download set from CompuServe allows a user to try all of the features before purchase (registration) and I can strongly recommend it.

The second item is a cry for help. Many years ago I came across a 'trick' in assembler that returned the remainder of an integer division without actually performing the division operation. It used (if I remember correctly) just *three* simple operations and it went something like 'XOR the dividend with the divisor-1 then subtract the divisor'. It really was as simple as that, although that is not it. With the passing of the years the actual method has now passed into myth and no-one believes me. I have tried many, many times to reconstruct the method but have failed miserably- does anyone out there know what it is? In fact are there any other interesting tricks, in assembler or otherwise, that can be dredged up from the time when programmers were mighty heroes who battled against the forces of evil using only hand-crafted code cut from living RAM?

Marvey Mills

What a load of junk

I would like to add my voice to Jules' in condemnation of the appalling flood of junk email that is released as soon as one ventures onto the web. Perhaps I might do this by providing the text of my current web page:

Today I received seventeen mail messages – two of these were useful mail, the other fifteen were junk advertising.

Of these fifteen all of them were from the United States of America – in fact all of the junk email I have *ever* received has been from the USA. Of these several were advertising products or services that could only be useful to Americans (which I am not), and some of these indicated that 'research showed that I would be interested in their products'! Three of the fifteen were from AOL addresses (this percentage is typical in my experience), and others had come via AOL addresses like wolist@aol.com, which indicates that AOL itself provides bulk mailing services (I have asked them about this, but received no reply). A number of them were people to whom I had already sent the requisitie remove reply.

This flood started when I created my web page, and has continued unabated. I have flamed the senders of this junk, but as they are mostly autosenders this has little effect; I have (on advice from AOL) sent complaints and copies of all junk from AOL tose-mail1@aol.com, which is supposed to deal with mail abuse; I have also mailed postmaster... for several of the worst offenders (I understand that all net sites are supposed to have a postmaster address, although it seems that many don't). All to no avail.

I wouldn't mind quite so much if I could just throw this junk away as I do with the junk snail-mail, but every item has to be examined to find the correct reply address (usually different from the sender, so I can't just press reply) and the correct way to get removed from the autosender database (remove in the title, in the body, etc.).

I only generated a web page for fun, and I have better things to do with my time than go through this incessant palaver with junk mail – I use email to communicate! For this reason, I have decided to remove my original web page (on which I worked hard) and replace it with this.

I would like to hear from people who agree with me, or who have any ideas as to how to publish a web page without getting drowned in junk. My CompuServe ID is a sixties carpet cleaner followed by the 3rd Fermat prime, and after the comma the only square that is two less than a cube followed by the maximum number of pieces into which a pancake can be cut with six slices. (Thanks to *The Penguin Dictionary of Curious and Interesting Numbers* by David Wells for all but the first). Let's see a spider work that one out!

Dave Midgley
email address supplied
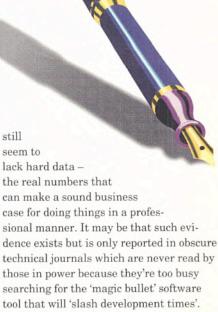
6 of the best

Dear Sir,

I just wanted to say how much I wholeheartedly agree with John Watson's diagnoses of (some of) the problems bedevilling software development. John rightly points out that suppliers and customers need to be aware of these issues. The problem is, how do we educate them ?

Part of the problem is that people are rarely held to account for defective software development or purchasing decisions as they would be in other professions.

The other part of the problem is that we

still seem to lack hard data – the real numbers that can make a sound business case for doing things in a professional manner. It may be that such evidence exists but is only reported in obscure technical journals which are never read by those in power because they're too busy searching for the 'magic bullet' software tool that will 'slash development times'.

Philip Reeves
Knowledge Engineer
email address supplied
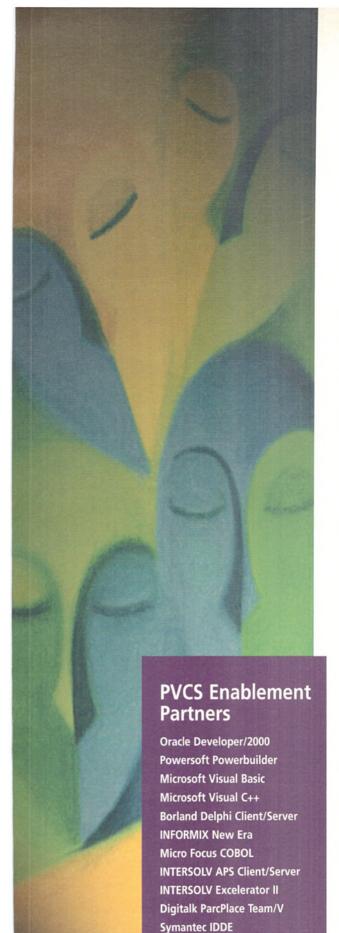
I need somebody's reld

Dear Sir,

I was fascinated by your article on producing Windows help files in the July issue of *EXE*, but baffled by what it says at the top of page 57. In English sign language the symbols read 'Hope Europe Lesbian Rocket', while in American sign language it says 'RELD'. Could you elucidate?

John Winters

Ahem. We were trying to spell out 'HELP', of course. Apparently, one of our characters (the first one) got mixed up. Notwithstanding the fact that 'Hope Europe Lesbian Rocket' sounds like one of those groups in the Eurovision song contest who never score any points, we hope that no-one was too offended by the mistake. It's nice to know that *EXE* readers are so observant, although funnily enough no-one seems to have spotted the dots and dashes at the back of the February 1996 issue which proclaimed 'Next issue of EXE in Morse code'! Unless you know otherwise... – Ed

Getting the (short) message

GSM digital cellular phones can do more than just voice calls. **Richard Griffiths** shows how to *talk* to the Short Message Service.

Why all the noise about digital mobile phones and computer communications? It may be trendy to dial the office LAN from a rowing boat on the Serpentine but there's little in it for software developers. Users buy a cellular phone and a datacard and then simply plug them into the software they're already using over a land line. That's what the card's for, isn't it? To make a GSM phone look like a modem. Apart from some forthcoming extensions to the AT command set, a datacard isn't very interesting. Communications programs don't care whether the connection is beamed through the ether or squeezed through buried copper – digital connections are just clean versions of analogue ones, aren't they?

Well, not quite. First of all, voice, fax and data are different types of call on a digital connection. Additional services are available, such as Calling Line Identity (CLI). And then there's the Short Message Service (SMS). Conceived originally as a paging mechanism for advising mobile phone users that voice mail is waiting, SMS is now increasingly used as a messaging service in its own right.

Figure 1 depicts an example of phone-to-phone SMS exchange. The sender uses the phone's keypad to create and send a short text message (1). The network routes the

Bytes	Information element	Meaning
1	Message Type	Insert SMS Message (the Message Type value is 00000111).
1	Insert Type	Store in phone and send over the air interface (the Insert Type value is 00000011). The two rightmost bits can be set for other types of insert commands, such as 'Store in phone but do not send'.
0-12	RP-Dest-Address	Address of recipient, up to 12 bytes
Up to 164	SMS-TPDU	A 'Submit' PDU, with up to 24 bytes of header information, plus up to 140 bytes of user data.

Table 1 – Sending an SMS in Block Mode.

message to its Short Message Service Centre (2) and stores it on a computer (3). When the recipient switches his phone on (4), the phone sends a signal to the network indicating that it is ready to receive (this signalling occurs every time a phone is switched on). The network detects the 'on' signal and locates the phone. The Service Centre is notified that the phone is on and transfers the message (5) from storage to the network, which delivers it to the phone (6). The phone displays the message on its LCD panel until the user cancels it. The message is delivered to the phone immediately if the phone is already on. Short message delivery uses the phone's signalling channel, rather than the voice channel. Delivery occurs without a call being made to or from the recipient's phone which can be either standing by or making a call.

Messages are stored in the phone or in the Subscriber Identity Module, or SIM card (the small card that you insert in the phone). The Service Centre delivers as many messages as it can, until the storage limit is reached. The rest of the messages are delivered when storage space is freed by the user reading and deleting messages. A phone usually has capacity for about 10 messages. Some networks offer an acknowledgement service: each time a message is successfully delivered to its recipient, the Service Centre indicates this to the sender, using another short text message.

SMS cocktail parties

As the name suggests, the service provides only for short messages – the limit is 160 characters. Typing messages on a mobile phone keypad with multi-function keys is, to say the least, awkward. What rescues SMS from the status of interesting novelty is the fact that the keypad isn't the only way into the world of short messages. There are two other ways in – data interfaces to the phone and to the Service Centre. This enables computer software to *speak* SMS and opens up a huge potential for short message applications.

First let's look at interfaces to SMS via the phone. The GSM standards are owned by the European Telecommunications Standards Institute (ETSI). SMS is defined in

standard GSM 03.40 – *Technical Realization of the Short Message Service Point-to-Point*. ETSI also defines interfaces between

the phone and terminal equipment (*ie* computers) in two other standards: GSM 07.05 and GSM 07.07. A grasp of these three standards is essential if you want to hold your own at SMS cocktail parties. GSM 07.05 defines the interface for SMS functions *only*. GSM 07.07 defines the interface for all other phone functions (*ie* everything *but* SMS).

The key point to remember is that a software application talks to the phone, *not* to the network. Although the title of GSM 03.40 might lead you to expect otherwise, SMS is really a store-and-forward service, not point-to-point. The application hands the message to the phone for onward transmission. The phone may not be able to send it immediately but will hold it until it can: the application does not necessarily have to wait for it to do so.

SMS has a four-layer architecture: Application Layer, Transfer Layer, Relay Layer and Lower Layers (*sic*). The phone implements all four layers. See Figure 2. The Application Layer contains the application interface and all the human interface – message construction, editing and display. The application layer uses services provided by the Transfer Layer. A Transfer Layer entity exchanges SMS-TPDUs (Transfer Protocol Data Units) with a peer layer in the Service Centre.

Three application interface protocols are defined by ETSI in GSM 07.05: Block Mode, Text Mode and PDU Mode. These are different ways of achieving much the same thing, but each protocol hides the Transfer Layer services differently. It is not clear why three protocols have been defined – they overlap in function and are clearly the result of two different lines of thought, as can be seen in the box *Three roads to SMS* on page 24.
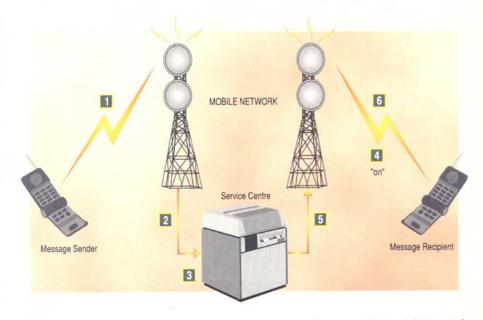

Figure 1 - The mobile network.

AT+CMGS="0123012345"<CR> This is a message to yoooo<CR> Bye^Z	Send two line message to 0123012345
+CMGS: 3 OK	The phone has accepted the message and has given it a reference number of 3

Table 2 – Sending a message in Text Mode.

GSM 07.07 provides an AT command interface for comprehensive control of non-SMS functions. Using GSM 07.07 commands you can, for example: find out the type and manufacturer of the phone; dial numbers held in the phone memory; set the call mode – voice, fax, data; dial and hangup; enable data compression on data calls; set call barring functions in the phone; enable Calling Line Identity to be displayed (if that service is provided by the network); check battery status; check received signal strength; search, read and write phone numbers held in the phone...

SMS applications may use a mix of GSM 07.05 and 07.07. For example, Block Mode might be used to handle SMS, with selected 07.07 commands used to monitor the phone and provide access to its stored phone numbers.

A single Short Message can transfer 140 bytes of user-data. GSM uses a 7-bit character set by default, and the user-data part of the SMS-TPDU is packed in such a way that the space is sufficient for 160 characters. Other character sets are available and all such things are described in yet another ETSI standard, GSM 03.38.

Microsoft's Telephony API (TAPI) lies somewhere between GSM 07.05 and 07.07. TAPI allows limited control of the phone, does not allow data or fax calls, but fortunately does allow access to SMS. The TAPI interface and associated products were covered in the *Wireless & Wired* article in the July '95 issue of *EXE*.

So much for the standards: the fun starts with the implementation of them. The standards are designed to be implementable on different arrangements of hardware; some elements of the standards are optional; and there's nothing to stop manufacturers implementing proprietary interfaces to their own brand phones.

Take the ubiquitous Nokia 2110 phone and its companion datacard. The datacard attaches to an internal bus in the phone via a cable. The card (a PCMCIA card) implements Block Mode, but in case you don't fancy that, it also has an interface of Nokia's own invention called the SMS Interpreter. This is very similar to, but not the same as, Text Mode. The card also implements GSM 07.07, which is why you can use it for data and fax calls. Ericsson and Philips supply phones with datacards implementing similar standards.

There is an Orbitel phone which implements the interfaces internally, and for that only a serial cable is required. It is likely that other phones will follow this model.

For applications where a phone keypad and display are not required, there's the Siemens M1, a black-box phone which can be controlled only by a software application. It is useful in situations where a normal mobile phone might be subject to tampering or misuse if left unattended, connected to the back of a computer.

Networks independence

Services on PCN networks (Orange and One-2-One) are the same as GSM (Vodafone and Cellnet). PCN is just another name for GSM operating at 1800 MHz instead of 900 MHz. The differences in the phones are restricted to the radio parts.

The ETSI standards are independent of service providers, so connecting to different networks is easy. The only network-specific information is the Service Centre number, which is always transmitted with an SMS message. The Service Centre number is usually entered once into the phone. You can do this via the phone's keypad, or via a GSM 07.05 or SMS Interpreter command.

Although no calls are made when using SMS, the sender will often be charged on a per-message basis. Charges vary from network to network but are typically around 10p. The recipient pays nothing.

The other way you can connect an application to SMS is much less glamorous: all you need is a modem. Your application talks directly to the Service Centre via a land line. There are some immediate disadvantages with this approach. First, the dial-up access protocols for the various networks' Service Centres are proprietary, and they are all different. Second, fewer functions are available compared to mobile origination. It is only possible to *send* short messages in this way: the Service Centre cannot call the application back to deliver the messages or receipt notifications.

However, there are many instances where one-way messaging (*ie* paging) is all that's needed, and for these the dial-up service is ideal. The hardware required is inexpensive – because of the small amounts of data any modem will do (2400 bps is fast enough) – and most dial-up services are subscription-free. The Service Centres are often accessible on Premium Rate numbers but even so, sending a single message costs between 5 and 10p at most depending on the network. Several messages can be batched within a single call. Unfortunately though, if you want to send the same message to a hundred phones, you have to send a hundred messages: there is no way of sending the message followed by a distribution list.

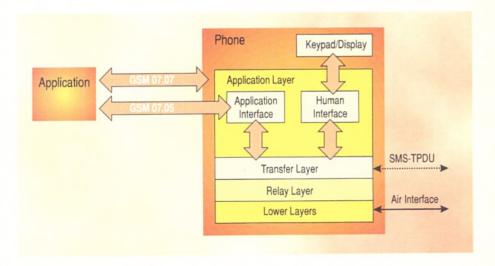Each network has its own Service Centre access protocol and each protocol is command-line based:

Figure 2 - SMS layer architecture

Three roads to SMS

H ere's a summary of Block Mode, Text Mode and PDU Mode, the three application interface protocols defined by the ETSI in GSM 07.05:

Block Mode

Block Mode is a binary protocol with error-correction, making it suitable for use in situations where the link between the application and the phone is subject to errors. The application must construct a binary string comprising header information and an SMS-TPDU. Block Mode is exclusively for SMS – when the application commands the phone to enter Block Mode, the phone is not available for data or voice calls until this mode is terminated. Text Mode and PDU Mode do not impose this restriction.

The application requests the phone to enter Block Mode by sending the AT command **AT+CESP**. The phone either enters Block Mode, or, if Block Mode is not supported, returns an error and remains in its default mode. Once Block Mode is entered, the application can request the following groups of operations:

- List messages held in the phone.
- Transfer of all messages or of a specific one from the phone.
- Set the phone so that all incoming messages are passed immediately to the application or that each time a message is received, the application is notified.
- Submit message for transmission.
- Delete messages from the phone.
- End Block Mode (at which point the **AT+CESP** returns **OK**).

Block Mode commands and responses are built from a predefined set of components or information elements. For example, an *Insert* SMS command, used to transfer a message to the phone, is constructed by concatenating the information elements *Message Type, Insert Type, RP-Dest-Address* and *SMS-TPDU* (see Table 1).

There are six types of SMS-TPDU: deliver, deliver report, status report, command, submit, submit report. The type is encoded into the header, and depending on the type, other elements of header information may or may not be present. One element, always present, is the Data Coding Scheme. It specifies how the user data is coded (which alphabet is used, whether it's compressed or not), a message class and message waiting indications.

There are three possible alphabets for short message text. If the 7-bit GSM default alphabet is used, it's possible to pack 160 characters into the 140 bytes available for user data. If 8-bit coding is selected, then the 'alphabet' is user-defined, and the message consists of 140 bytes into which can be encoded anything the user wishes. If the UCS2 alphabet is used (16-bit), then only 70 characters can be packed into the user data field.

The compression status indicated by the Data Coding Scheme field relates to the compression which occurs before the data

goes into the user-data field, allowing more than 160 characters to be packed into 140 bytes. Unfortunately, the GSM standard compression algorithm is not yet specified.

There are four message classes which can be assigned to a short message. The class determines how the receiving phone handles the message:

Class 0 – the message is displayed on the phone's screen but not stored.
Class 1 – the phone will acknowledge receipt to the Service Centre when the message has been stored in the SIM or other phone memory.
Class 2 – means 'SIM-specific' and requires the phone to store the message in the SIM memory only.

Class 3 – is terminal-equipment specific and indicates that the message is specifically destined for an application sitting behind the phone. The phone is required to respond with an acknowledgement without waiting for the message to be transferred to the application, only waiting for the message to be stored.

Message waiting indication allows the use of compact SMS messages to indicate fax, voicemail, or email waiting on other systems connected to the mobile network. The header is used to tell the receiving unit to store or discard the contents of the user data field.

Other elements of the SMS-TPDU header are used for a message sequence number (assigned by the phone when it submits a message), indication that the Service Centre has more messages to deliver to the phone, request for an acknowledgement to a submitted message, message validity period, Service Centre time stamp and others.

Text Mode

Text Mode is a simple character-based interface comprising a series of AT commands. The application needs no knowledge of SMS-TPDUs because the phone constructs the PDU and passes it to the Transfer Layer. In this mode, sending an SMS message is simple, see Table 2.

Using Text Mode requires a different approach to Block Mode. The application first sets up fixed parameters such as the Service Centre address. The phone uses those parameters to construct an SMS-TPDU when the application requests it to send a message. Functions differ slightly to Block Mode: although you can store messages in the phone using Text Mode, you can't, for example, set the phone to automatically pass all incoming messages to the application, only to notify it, as in the example in Table 3.

PDU Mode

PDU Mode is almost identical to Text Mode, but for certain commands (*eg* send message) the application has to build or decode the SMS-TPDU. PDU Mode mixes the convenience of an AT command approach with the flexibility of being able to construct the TPDU, allowing binary encoded data to be sent, not just characters.

- Orange Direct Access (ODA)
- Vodafone Telenote
- Cellnet Telelocator Access Protocol (TAP)
- One 2 One TAP

Vodafone also offers X.25 access to its Service Centre, via Paknet. The application must use a relatively expensive radio-PAD, but this method does allow bi-directional mail and receipt reports, making it very similar to mobile origination. Other methods of Service Centre access may be provided by the networks, such as voice. Other outputs are already adding value to SMS: Vodafone offers an SMS-to-fax service, for example.

Companion services

SMS has a companion service called the Cell Broadcast Service (CBS). SMS is a one-to-one service; CBS is a one-to-many service. A single CBS message is sent to every mobile in the originating cell. CBS is used for services such as traffic information, which need to be regional or cell-based. The interfaces defined in GSM 07.05 include CBS, but the freedom to access this service is not given away as easily as SMS. Given its commercial potential (and the potential for misuse) CBS may be available only by negotiation with service providers.

Products have been on the market for some time which provide simple SMS clients on PCs. These products use mobile origination and dial-up access to allow a single PC user to send and receive short messages. Some products are single-network, others are multi-network. These products, although useful, do not offer scope for further integration. SMS-mail gateways allow email systems to be connected to SMS, effectively extending office email into the pockets of mobile personnel – without the use of laptops. Multi-purpose SMS-application gateways let system integrators plug existing applications into SMS. The interfaces offered by such products can be surprisingly easy: because SMS is a store-and-forward service, direct control of SMS by applications is not particularly useful and a simple asynchronous file-interface can provide enough
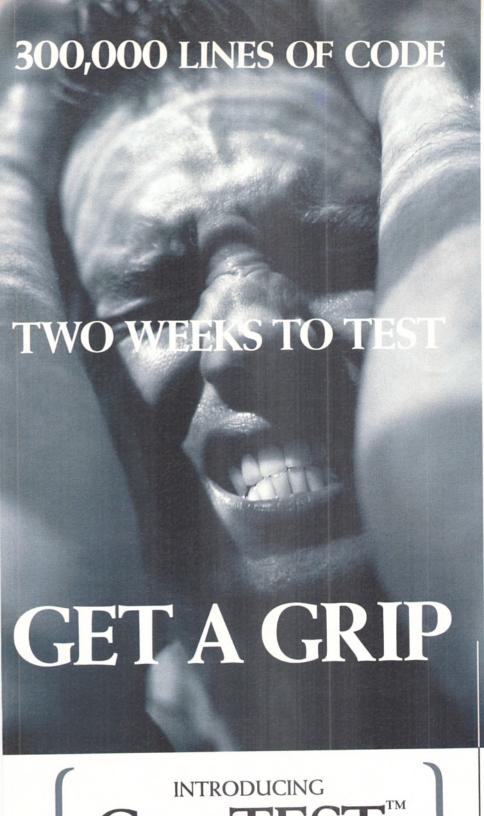
`AT+CNMI=2,1,0,1,0`	Instruct the phone to handle new message indications as follows:
	2 – buffer until the link to the application is available.
	1 – if an SMS message arrives, notify the application with a memory location and index.
	0 – don't tell the application about cell broadcast messages.
	1 – pass status reports to the application.
	0 – flush result codes to application under certain states.
`OK`	
`+CMTI: 5, 10`	Indication that the phone has received an SMS message and stored it in memory location 5, with index number 10.

`+CDS:`
`2,3,"0123012345",145,`
`"96/08/06/21:36:45+04",`
`"96/08/06/21:37:00+04",0`
	Status report arrives for a previously sent message:
	2 – first byte of report in integer format.
	3 – reference number assigned to SMS message.
	0123012345 – recipient phone.
	145 – code indicating recipient address type.
	96/08/06/21:36:45+04 – Service Centre time stamp (GMT +4 hours).
	96/08/06/21:37:00+04 – delivery time.
	0 – status code.
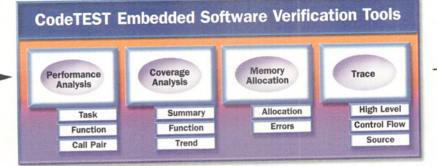
Table 3 – Setting up some configuration parameters.

functions for existing systems to send and receive short messages.

Some of the products on the market allow access to paging networks. Pagers obviously operate on a similar principle to SMS. The input to a paging network is similar to the dial-up access to the SMS Service Centres and again, the protocols used are all proprietary.

The 160 character limit is often quoted as SMS' main limitation. The implication is that only short, character-based information can be transferred. We have already seen that binary data can be transmitted as well and SMS provides for message concatenation. The standard allows for up to 255 short messages to be sent as a set. Each message contains a sequence number, a number identifying the set to which the message belongs and the total number of messages in the set. These enable the receiving application to assemble the sequence of messages in the correct order, identify whether any are missing, and then extract and process the data. It is left up to the applications to sort out how to deal with situations where

one or more messages is lost or significantly delayed. Note that the phone does not assemble the message sequence – the phone receives and stores messages in the order in which they arrive.

Short message concatenation offers rapid transfer of data without having to make data calls. There are, however, two problems. First, when short messages cost 10p, it rapidly becomes cheaper to set up a data call. You can work out the break-even for yourself. On international calls the break even will be different, because the cost to send a short message will remain the same whether you send it to Paris or Sydney. Bulk deals are available from some networks, where short messages are purchased in advance, reducing the per-message cost significantly. But if you figure that SMS might be a great way of doing cheap intercontinental spreadsheet transfer, consider the second problem. SMS uses the signalling channel, which isn't intended for heavy duty data transfer. Service providers might well take umbrage at such flagrant misuse.

Getting the message

SMS seems limited then, to either single short messages, or short concatenations of messages. Nevertheless, there is considerable interest in the service, which is seen by many as being more useful and convenient than voicemail, especially when combined with an office-based email network. SMS can get the message over more conveniently and more cheaply than voicemail.

There seem to be no guaranteed service levels with SMS, and transfer times do vary. Generally though, messages arrive within a few seconds.

One last feature. At today's crowded exhibitions an attempt to make a mobile call frequently results in the 'no service' message because the cell is full of men-in-suits-with-mobiles. Try SMS – it will still be available.

◼

Richard Griffiths is a messaging consultant working with Dialogue Communications, suppliers of SMS messaging products. He can be reached by phone at 07000 400340, by fax at 07000 400341 or alternatively by SMS at 0370 391779.

Dialogue's Web home page is at http://www.dialogue.co.uk. ETSI can be contacted at secretariat@etsi.fr.

BoundsChecker™ 4.0 and SoftICE™ for Windows NT™

NEW! Delphi 2.0 support
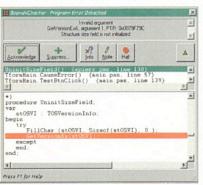
Delivering more power, control, and visibility than ever before...

BoundsChecker

Become even more productive next time you use either Borland's new Delphi® 2.0 compiler or Borland's C++ Compiler v 4.5 or better! NuMega's award winning BoundsChecker is now available for both Windows® 95™ and Windows NT delivering more automatic error detection power than ever before! You'll save loads of time hunting and hacking for coding errors because BoundsChecker's context sensitive help system will not only inform you of the probable causes of coding errors, but also suggest corrections. When BoundsChecker detects an error, it will automatically pop up with a complete description, stack trace and source line for the error. Visit our Web site to see a complete listing of all the errors BoundsChecker detects.

OLE/Internet Debugging

Have you noticed how important OLE/ActiveX™ Technologies and the Internet are becoming to Windows™ application development? Now, not only do you need the best memory error detection tool, you need a product that has been specifically designed to find the nastiest bugs in OLE, ActiveX and Internet applications. Use our comprehensive knowledge of Windows internals, OLE and Internet technologies while you debug your next application!

New In BoundsChecker 4.0

- ❑ Parameter validation and logging of standard OLE interfaces
- ❑ User-extensible API validations
- ❑ Borland Delphi™ 2.0 32-bit applications support

SoftICE
The Advanced Windows Debugger

Absolute Visibility Into Windows NT

If you're an expert Windows® developer, you need NuMega's NEW *SoftICE*, the advanced Windows debugger for Windows NT. It blasts through the Windows NT system barriers and takes you into the depths to help you understand and solve difficult problems. *SoftICE* gives you the control, visibility, and raw power to debug right down to the metal on a single machine. No other software debugger reveals Windows NT so completely.

Total Control

Developing kernel-mode drivers can be tough work, but *SoftICE* gives you the raw power to control the whole system. Developing system services and drivers is a breeze with *SoftICE*. It delivers all of this power on a single machine, without any special debug code, right on the retail version of Windows NT. Remarkable!

See WHAT You've Been Missing!

Don't be afraid to look beneath the surface. *SoftICE* has Windows NT figured out. Do you see something unfamiliar? Just use the new WHAT command and let *SoftICE* explain what you're seeing. Whether it's a process, module, control block, header, symbol, image base, or just about anything else in the system — *SoftICE* can identify nearly everything in the system for you.

Debug Tough Problems With Ease

SoftICE lets you conditionally watch and control system events to help identify resource deadlocks and timing problems. *SoftICE* breakpoints can be qualified by conditional "C" expressions for ultimate breakpoint granularity.

NuMega Products			
Product	**Platform**	**Product**	**Platform**
BoundsChecker *Standard Edition*	NT, 95	SoftICE	NT, 95, 3.x, DOS
BoundsChecker *Standard Edition* Pack	NT & 95	SoftICE Trio	NT, 95, 3.x
BoundsChecker	Win 3.x, DOS	SoftICE Win32 Pack	NT & 95

Don't Wait!
Upgrade to the next version of BoundsChecker or *SoftICE* Today.

BoundsChecker *Standard Edition* 4.0 • *SoftICE* for Windows95 • *SoftICE* for Windows NT

www.numega.com

CIRCLE NO. 801

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

NuMega
Technologies™

SYSTEM SCIENCE

NuMega Technologies, Inc. • 9 Townsend West • Nashua, NH 03063, 603.889.2386 • Fax 603.889.1135 • info@numega.com • www.numega.com

Extended Windows APIs

Ever been frustrated by the limitations of the Windows API? **Dave Jewell** fixes the plumbing and comes up with a brand new FindFirstFile routine.

Y ou may remember that in the July issue of *EXE*, I presented the code for a simple desktop-enhancing utility called QuickStart. This program worked by scanning the contents of the Windows 95 desktop folder (usually stored in C:\WINDOWS\DESKTOP) and used this to build a hierarchical menu which could then be accessed from a special icon in the so-called Tray area of the Taskbar.
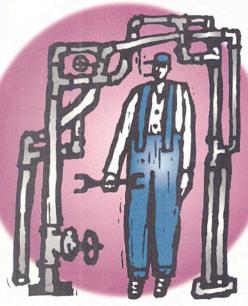
While developing this program, I was forced (for the 'umpteenth time) to write a routine which would *first* scan a directory for all sub-directories it might contain and *secondly* re-scan the directory for any actual files that might be found. The FindFirst-File/FindNextFile API calls won't automatically do this for you in one go – you either have to make two separate passes through each encountered directory, or else specify the file attributes such that you get absolutely everything in one pass. Either way, it's messy.

In addition to the above, the Find-FirstFile/FindNextFile routines happen to return files in the order in which they're encountered in the underlying operating system's directory, rather than sorting them into alphabetical order. I thought that it would be much nicer if the files could be pre-sorted and then passed back to the application program alphabetically. Here's an improved version of FindFirstFile/-FindNextFile. I tentatively christened my new routine FindFirstFileEx...

Enter (& exit) FindFirstFileEx...

Not long after this, I received my first beta copy of Windows NT 4.0 through the MSDN programme. To my surprise, I discovered that KERNEL32 now exports a routine called – yes, you've guessed – FindFirst-FileEx! I couldn't help wondering if the NT 4.0 development team have been using some sort of new ActiveTelepathy OCX gadget. At this point, I wondered if this article was a non-starter: obviously Microsoft had got there first.

The FindFirstFileEx API is documented on the July 1996 MSDN CD: the extensions to FindFirstFileEx make no

provision for alphabetic sorting and there's no provision for passing back all the directory names before file names. Instead, FindFirstFileEx contains some relatively arcane extensions for reporting device names and not much else besides. Clearly, I wasn't going to get much help there so I decided to go ahead and 'roll-my-own'.

The result of these deliberations is shown in Listing 1. This is a Delphi Pascal unit, although it doesn't use any OOP extensions and could easily be wrapped up into a DLL for calling from Delphi, C++ or whatever. The code does not use any global variables and is inherently DLL-friendly. It will support multiple clients at the same time.

In an attempt to distance myself from the FindFirstFileEx code, I've called the main routine FindFirstFileExEx! Similarly, the other two routines exported by this unit care called FindNextFileExEx and FindCloseExEx. These three routines are used much like the Microsoft API routines – you start by calling FindFirstFileExEx which retrieves information on the first matching file and also returns a magic-cookie 'find handle' to the calling application. This handle is essential for both my code and Microsoft's since it contains per-client information and state data which is used to determine how far along we are in enumerating the list of matching files. As with the API routines, successive file entries are returned through calls to the Find-

NextFileExEx routine and finally the find handle is closed by making a call to Find-CloseExEx.

It's important to bear in mind that, at the operating system level, my code performs *all* file matching during the initial call to FindFirstFileExEx. At this time, a complete list of matching files is built and stored in memory. This is analogous to many of the 32-bit TOOLHELP routines which take a 'snapshot' of the current state of play in order to avoid difficulties which would otherwise arise under a fully pre-emptive operating system. This means that – potentially – certain files in the stored file list might be deleted or renamed before being enumerated by the calling application. The code presented here attempts to make a 'just-in-time' (it's a trendy buzz-word, so why shouldn't I use it too?) check of the status of a file before finally returning details of the file to the caller.

Getting sorted out

If you look at the interface part of the file, you'll see that the FindFirstFileExEx has an extra parameter, ExFlags, for which there's no counterpart in the Microsoft code (see listing 1). This flags word may contain one or more of the following values – the first three are mutually exclusive, while the last two are modifiers which are OR'd into the parameter.

When ff_RawOrder is specified, file entries are returned to the caller in the order in which they're encountered. This means that directory entries will typically be jumbled up with ordinary file names. However, the ff_AlphaSort modifier may optionally be used in conjunction with the ff_RawOrder flag. This will perform alphanumeric sorting on the entire list of entries.

Instead of using ff_RawOrder, you can opt for either ff_DirsFirst or ff_Files-First. As the names suggest, these two possibilities will move all the directory entries up to the top of the file list, or down to the bottom respectively. (Many people prefer to see directory entries grouped together, and it is a source of continuing amazement to me that Microsoft's COMMAND.COM shell *still* does not operate this way, even under DOS 7.)

```pascal
unit FindFile;

interface

uses Windows;

const
  ff_RawOrder    = 0;     // Leave things as they are
  ff_DirsFirst   = 1;     // Put directories first
  ff_FilesFirst  = 2;     // Put files first
  ff_DecorateDir = $40;   // Do 'sumat special with dirs
  ff_AlphaSort   = $80;   // Do alpha sort on names

type
  TFind32 = TWIN32FindDataA;   // just less chars to type!

function FindFirstFileExEx (lpFileName: PAnsiChar; ExFlags: Word;
                     var lpFindFileData: TFind32): THandle;

function FindNextFileExEx (hFindFile: THandle;
                     var lpFindFileData: TFind32): Bool;

function FindCloseExEx (hFindFile: THandle): Bool;

implementation

const
  FilesInBlock = 1024;    // number of files in each block

type
  PPChar = ^PChar;
  PFFList = ^FFList;
  FFList = record
             // First six entries only used in first block
             FFExFlags: Word;
             FFIdxBlock: PFFList;
             FFIdxEntry: Integer;
             FFTotEntries: Integer;
             FFDirEntries: Integer;
             FFPath: array [0..MAX_PATH - 1] of Char;

             FFList: array [0..FilesInBlock - 1] of PChar;
             FFNext: PFFList;
           end;

  CompFunc = function (A, B: PPChar): Integer;

// -------------------------------------------------------------
// Name:    AlphaSort
// Purpose: Do a standard alpha sort
// -------------------------------------------------------------
function AlphaSort (A, B: PPChar): Integer;
var
  AA, BB: PChar;
begin
  AA := A^ + 1;  BB := B^ + 1;
  if AA^ = '|' then Inc (AA);
  if BB^ = '|' then Inc (BB);
  AlphaSort := lstrcmpi (AA, BB);
end;

// -------------------------------------------------------------
// Name:    SortDirsFirst
// Purpose: Sort directory entries before ordinary filenames
// -------------------------------------------------------------
function SortDirsFirst (A, B: PPChar): Integer;
var
  AA, BB: Char;
begin
  AA := (A^ + 1)^;  BB := (B^ + 1)^;
  if AA <> '|' then AA := 'X';
  if BB <> '|' then BB := 'X';
  if AA = BB then SortDirsFirst := 0
  else if AA = '|' then SortDirsFirst := -1
  else SortDirsFirst := 1;
end;

// -------------------------------------------------------------
// Name:    SortDirsLast
// Purpose: Sort directory entries after ordinary filenames
// -------------------------------------------------------------
function SortDirsLast (A, B: PPChar): Integer;
begin
  SortDirsLast := SortDirsFirst (B, A);
end;

// -------------------------------------------------------------
// Name:    GetEntry
// Purpose: Return a pointer to the specified entry, NOT the entry itself.
// -------------------------------------------------------------
function GetEntry (list: PFFList; idx: Integer): PPChar;
begin
  // Find wanted block
  while idx >= FilesInBlock do
  begin
    list := list^.FFNext;
    Dec (idx, FilesInBlock);
  end;
  // Return specific entry
  GetEntry := @list^.FFList [idx];
end;
```

```pascal
// -------------------------------------------------------------
// Name:    QuickSort
// Purpose: The ubiquitous QuickSort algorithm
// -------------------------------------------------------------
procedure QuickSort (list: PFFList; iLo, iHi: Integer; Comp: CompFunc);
var
  Lo, Hi: Integer;
  Mid: PPChar;
  T: PChar;
begin
  Lo := iLo;
  Hi := iHi;
  Mid := GetEntry (list, (Lo + Hi) div 2);
  repeat
    while Comp (GetEntry (list, Lo), Mid) < 0 do Inc (Lo);
    while Comp (GetEntry (list, Hi), Mid) > 0 do Dec (Hi);
    if Lo <= Hi then
    begin
      T := GetEntry (list, Lo)^;
      GetEntry (list, Lo)^ := GetEntry (list, Hi)^;
      GetEntry (list, Hi)^ := T;
      Inc(Lo);
      Dec(Hi);
    end;
  until Lo > Hi;
  if Hi > iLo then QuickSort (list, iLo, Hi, Comp);
  if Lo < iHi then QuickSort (list, Lo, iHi, Comp);
end;

// -------------------------------------------------------------
// Name:    SortEntries
// Purpose: Sort the file list as specified in flags word.
// -------------------------------------------------------------
procedure SortEntries (hFindFile: THandle);
var
  aSort: Bool;
  list: PFFList absolute hFindFile;
begin
  with list^ do
  begin
    aSort := (FFExFlags and ff_AlphaSort) <> 0;
    case (FFExFlags and 3) of
      ff_RawOrder:
        if aSort then QuickSort (list, 0, FFTotEntries - 1, AlphaSort);
      ff_DirsFirst:
        begin
          QuickSort (list, 0, FFTotEntries - 1, SortDirsFirst);
          if aSort then
          begin
            if FFDirEntries <> 0 then
                QuickSort (list, 0, FFDirEntries - 1, AlphaSort);
            if FFDirEntries <> FFTotEntries then
                QuickSort (list, FFDirEntries, FFTotEntries - 1, AlphaSort);
          end;
        end;
      ff_FilesFirst:
        begin
          QuickSort (list, 0, FFTotEntries - 1, SortDirsLast);
          if aSort then
          begin
            if FFDirEntries <> FFTotEntries then
                QuickSort (list, 0, FFTotEntries - FFDirEntries - 1,
                           AlphaSort);
            if FFDirEntries <> 0 then
                QuickSort (list, FFTotEntries - FFDirEntries,
                           FFTotEntries - 1, AlphaSort);
          end;
        end;
    end;
  end;
end;

// -------------------------------------------------------------
// Name:    FindFirstFileExEx
// Purpose: Build the initial file list and return first entry.
// -------------------------------------------------------------
function FindFirstFileExEx (lpFileName: PAnsiChar; ExFlags: Word;
                          var lpFindFileData: TFind32): THandle;
var
  hf: THandle;
  p: PChar;
  findData: TFind32;
  FFCurBlock, FFNewBlock: PFFList;
  idx, dir, tot: Integer;

  function AllocName (szName: PChar): PChar;
  var
    sz: Byte;
    p: PChar;
    fIsDir: Bool;
  begin
    fIsDir:=(findData.dwFileAttributes and File_Attribute_Directory) <> 0;
    sz := lstrlen (szName) + 2;
    if fIsDir then Inc (sz);
    GetMem (p, sz);
    AllocName := p;

    p^ := Char (sz);
    Inc (p);
```

Listing 1 – The simple Delphi Pascal unit source code. It doesn't use any OOP extensions or global variables and is inherently DLL-friendly.

TECHNIQUES

Finally, you can use the `ff_Decorate-Dir` bit-flag to put a couple of square brackets around directory names, just to make them stand out from the crowd. If you've got your own preferences for directory name decoration, the source is pretty easy to modify.

How does it work?

The unit's heart is the declaration for the `FFList` record present in the `type` section within the implementation part. When the `FindFirstFileExEx` routine builds a list of matching files, it stores the files as a linked list of `FFList` blocks. Of the various fields in this data structure, the first six fields are only used in the first block of the list. I could have defined a completely different data structure for the 'header block', but doing it this way keeps the code simple. The `FFNext` field points to the next `FFList` block, thus forming a linked list. In practice, there will almost certainly be only one 'link' in your linked list since a single block can store 1024 file names and you rarely find more than that number of files in a single directory. The file names are stored as individually allocated pointers in the `FFList` field.

Each file name is stored as a length byte, followed by an optional '|' character (indicating that this is a directory entry) followed

```
    if fIsDir then
    begin
      p^ := '|';
      Inc (p);
    end;

    lstrcpy (p, szName);
    if fIsDir then Inc (dir);
  end;

begin
  // Assume failure
  FindFirstFileExEx := Invalid_Handle_Value;
  // Get the first entry
  hf := FindFirstFile (lpFileName, findData);
  if hf <> Invalid_Handle_Value then
  begin
    tot := 0; idx := 0; dir := 0;
    GetMem (FFCurBlock, sizeof (FFlist));
    FillChar (FFCurBlock^, sizeof (FFList), 0);

    // Set up the special stuff which pertains only to the first block.
    FFCurBlock^.FFExFlags  := ExFlags;
    FFCurBlock^.FFIdxBlock := FFCurBlock;
    FFCurBlock^.FFIdxEntry := 0;

    // Determine path name to dir entries and store in handle block.
    // ASSUMPTION: That lpFileName specifies an absolute pathname
    lstrcpy (FFCurBlock^.FFPath, lpFileName);
    p := FFCurBlock^.FFPath + lstrlen (FFCurBlock^.FFPath) - 1;
    while (p^ <> '/') and (p^ <> '\') and (p > FFCurBlock^.FFPath) do
      Dec (p);

    if p <= FFCurBlock^.FFPath then
    begin
      // Eeeek!
      FreeMem (FFCurBlock, sizeof (FFlist));
      Exit;
    end
    else (p+1)^ := #0;

    FindFirstFileExEx := THandle (FFCurBlock);

    while True do
    begin
      // We're not interested in '.' and '..' directory names
      if findData.cFileName [0] <> '.' then
      begin
        FFCurBlock^.FFList [idx] := AllocName (findData.cFileName);
        Inc (tot); Inc (idx);
        if idx = FilesInBlock then
        begin
          idx := 0;
          GetMem (FFNewBlock, sizeof (FFList));
          FillChar (FFNewBlock^, sizeof (FFList), 0);
          FFCurBlock^.FFNext := FFNewBlock;
          FFCurBlock := FFNewBlock;
        end;
      end;

      if not FindNextFile (hf, findData)  then break;
    end;

    FindClose (hf);
    PFFList(Result)^.FFTotEntries := tot;
    PFFList(Result)^.FFDirEntries := dir;

    if tot <> 0 then
    begin
      SortEntries (Result);
      FindNextFileExEx (Result, lpFindFileData);
    end
    else
    begin
      FindCloseExEx (Result);
      FindFirstFileExEx := Invalid_Handle_Value;
    end;
  end;
end;

// ─────────────────────────────────────
// Name:    FindCloseExEx
// Purpose: Free up and storage needed by the file list.
// ─────────────────────────────────────

function FindCloseExEx (hFindFile: THandle): Bool;
```

```
var
  p: PChar;
  next: PFFList;
  sz, idx: Integer;
  FFCurBlock: PFFList absolute hFindFile;
begin
  FindCloseExEx := FFCurBlock <> Nil;
  while FFCurBlock <> Nil do
  begin
    for idx := 0 to FilesInBlock - 1 do
    begin
      p := FFCurBlock^.FFList [idx];
      if p <> Nil then FreeMem (p, Integer (p^));
    end;

    next := FFCurBlock^.FFNext;
    FreeMem (FFCurBlock, sizeof (FFList));
    FFCurBlock := next;
  end;
end;

// ─────────────────────────────────────
// Name:    FindNextFileExEx
// Purpose: Return next entry from the file list.
// ─────────────────────────────────────

function FindNextFileExEx (hFindFile: THandle;
                           var lpFindFileData: TFind32): Bool;

var
  p: PChar;
  hf: THandle;
  fIsDir: Bool;
  FullPath: array [0..MAX_PATH - 1] of Char;
  FFFirstBlock: PFFList absolute hFindFile;

begin
  // Assume failure
  FindNextFileExEx := False;
  // Loop until successful or nothing left to try!
  while FFFirstBlock^.FFTotEntries <> 0 do
  begin
    Dec (FFFirstBlock^.FFTotEntries);

    // Build the full pathname to the wanted file
    lstrcpy (FullPath, FFFirstBlock^.FFPath);

    p := FFFirstBlock^.FFIdxBlock^.FFList [FFFirstBlock^.FFIdxEntry];
    Inc (p);                  // skip the length byte
    fIsDir := False;
    if p^ = '|' then
    begin
      Inc (p);                // skip directory specifier
      fIsDir := True;
    end;
    lstrcat (FullPath, p);

    // Update idx fields for next time round
    Inc (FFFirstBlock^.FFIdxEntry);
    if FFFirstBlock^.FFIdxEntry = FilesInBlock then
    begin
      FFFirstBlock^.FFIdxEntry := 0;
      FFFirstBlock^.FFIdxBlock := FFFirstBlock^.FFIdxBlock^.FFNext;
    end;

    hf := FindFirstFile (FullPath, lpFindFileData);
    if hf <> Invalid_Handle_Value then
    begin
      FindNextFileExEx := True;
      FindClose (hf);

      // Did we want to decorate directory name?
      if fIsDir and ((FFFirstBlock^.FFExFlags and ff_DecorateDir) <> 0)
      then begin
        lstrcpy (FullPath, lpFindFileData.cFileName);
        lstrcpy (lpFindFileData.cFileName, '[');
        lstrcat (lpFindFileData.cFileName, FullPath);
        lstrcat (lpFindFileData.cFileName, ']');
      end;

      Exit;
    end;
  end;
end;

end.
```

Listing 1(cont'd) – The simple Delphi Pascal unit source code. It doesn't use any OOP extensions or global variables and is inherently DLL-friendly.

PROFESSIONAL TOOLS FOR PROFESSIONAL DEVELOPERS

FROM GREENLEAF SOFTWARE

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

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

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

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

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

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

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

GREENLEAF
Software ©

PREMIER LIBRARIES FOR DOS AND WINDOWS

by the name itself. The length byte makes it easy to de-allocate the various filename entries because the Delphi Pascal `FreeMem` routine expects to be given the length of the memory block being de-allocated.

More importantly, I chose to store only the filename in order to reduce memory requirements for the file list. In principle one could store the entire `TWIN32FindDataA` record for each file but this would be tremendously wasteful of space because `TWIN32FindDataA`

Figure 1 – The proof of the pudding is in the eating.

is a fixed length data structure capable of holding very long file names. Instead, I chose to regenerate the `TWIN32FindDataA` record by making a new call to `FindFirstFileEx` at the time that each entry is passed back to the caller. This is what I mean by 'just-in-time' checking; in addition to providing a last-minute check that the file is still there, this approach substantially reduces memory requirements because we only need to save the name of each file. As noted above, directory entries are pre-fixed with a 'l' character so that they can be identified when implementing a `ff_DirsFirst` or `ff_FilesFirst` sort.

Since only file-names are stored, it's necessary to save the path to the filename so that the full path can be reconstructed prior to calling `Find-FirstFileEx`. This is stored in the `FFPath` field of the first block. Similarly, the flags word, the total number of filename entries
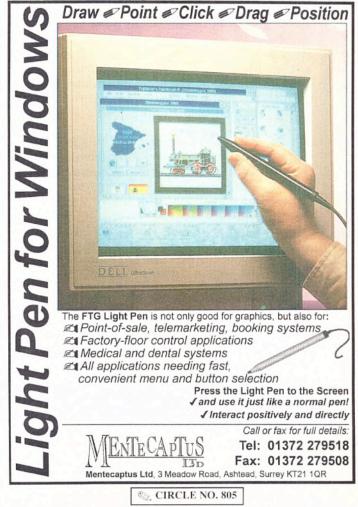
and the number of encountered directory entries are all stored as fields in this block. The `FFIdxBlock` and `FFIdxEntry` fields to point to the next entry to be returned to the caller.

Building the file list

Building the file list is relatively straightforward. When called, the `FindFirstFile-ExEx` routine first invokes `FindFirstFile` to create an API-level find handle and then loops round calling `FindNextFile` until no more matching files are found. For each iteration round the loop, the `AllocName` local routine is used to store the wanted filename in the next free position in the list. Once a complete block has been filled, another block is created and daisy-chained onto the previous one. Assuming that one or more matching files were found, the routine then calls the `SortEntries` routine to do the actual business of sorting the file list. Finally, `FindNextFileExEx` is called to return the first entry to the calling application.

The `SortEntries` code uses our old friend, the QuickSort algorithm. Because we're dealing with pointers, the QuickSort implementation used here can easily swap a couple of file name entries by simply swapping two pointer values. However, since those pointer values can potentially

be in different blocks, I wrote a little helper routine, `GetEntry`, which, given an index value, returns a pointer to a specific entry – *not* the actual entry itself.

I've also adopted the idea of passing a pointer to a function which performs the actual comparison. This enables us to change the behaviour of the sort algorithm depending upon which comparison function is actually used. In this case, there are three comparison routines:

- `AlphaSort` – performs a straight alphanumeric comparison using the `lstrcmpi` function.

- `SortDirsFirst` – place directories at the top of the list.

- `SortDirsLast` – place directories at the bottom of the list.

If the `ff_RawOrder` specifier has been set in the flags word, then the `SortEntries` routine calls QuickSort to perform an alphanumeric sort on the entire list – provided that `ff_AlphaSort` was also specified. If not, then the list is left alone. In the case of `ff_DirsFirst` and `ff_Files-First`, the list is first sorted to get all the directories at the top or bottom of the list, and then two further sorts are optionally performed (again depending on the presence of the `ff_AlphaSort` flag) to sort the directory and filename parts of the list. There are probably better ways of doing this such as adding each filename entry into a tree within the main `FindNextFile` loop, but I didn't think the extra code complexity warranted it – the final result was pretty fast on my machine, even where a very large number of files were being processed.

The remaining piece of the jigsaw is the `FindNextFileExEx` code. This routine picks off the next filename entry each time that it's called. The filename is appended to the end of the path specification and `Find-FirstFile` is called to retrieve a valid `TWIN32FindDataA` record for the file in question. At this point, there's a slight possibility that the call might fail because some other process might have deleted or renamed the file since we first called `Find-FirstFileExEx`. If this happens, then the code simply moves on to retrieve the name of the next entry – it doesn't really make sense to do anything else.
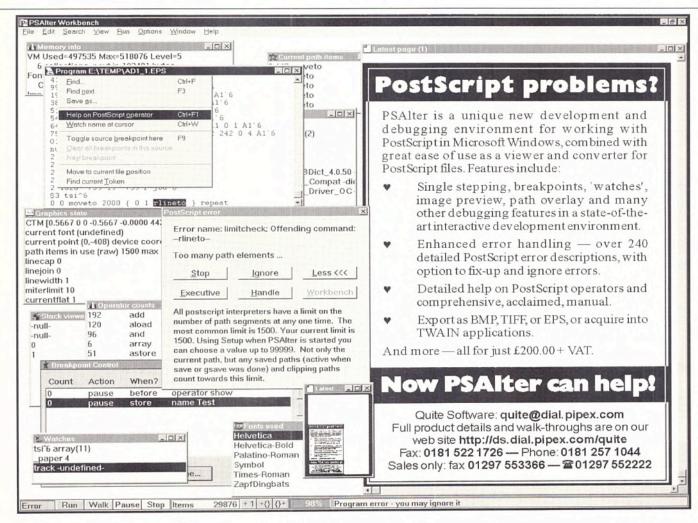
One simple enhancement to this code would be to add another function called (say) `FindFilesLeftExEx` which simply returns the `FFTotEntries` field for the given find handle, it would be useful from the viewpoint of memory allocation. This would give the calling application the oppor-

tunity to know in advance the maximum number of matching filenames that were going to be returned at any point in time (bear in mind that `FFTotEntries` is decremented at least once each time `Find-NextFileExEx` is called). However, the calling program would need to bear in mind that the value returned from `Find-FilesLeftExEx` is a *maximum possible* value, and not an accurate indication of what's going to be returned.

The final job of the `FindNextFileExEx` routine is to 'decorate' a directory name if so desired. This is done simply by sticking a pair of square brackets around the directory name. As pointed-out earlier, you can easily change this to suit your personal tastes or even create an application-supplied hook to allow the program to decorate a directory name as desired. Hmmm… anyone for `Find-FirstFileExExEx`? ∎

Dave Jewell is a freelance consultant, programmer and technical author. You can contact Dave as DSJewell@aol.com, 102354.1572@compuserve.com or DaveJewell@msn.com.
The source code for this Delphi application, along with the complete code for the Find-File unit, can be downloaded from the EXE conference on CIX or from EXE's FTP site.
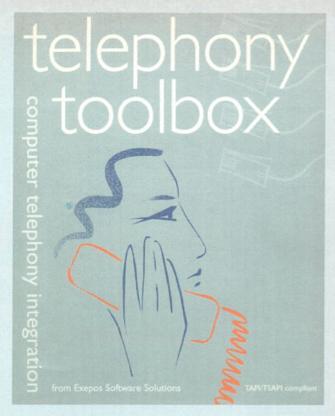
Developing CTI apps?

Need to telephony enable your existing apps?

Wondering whether to use TAPI or TSAPI?

Stop wondering! Start developing!

Telephony Toolbox from Exepos is a full function 16/32 bit control (OCX, VBX, VCL, DLL) which supports both TAPI and TSAPI.

With Telephony Toolbox you can write a dialler in **10 minutes** which will work on any TAPI/TSAPI device (virtually all major switches support one standard or the other).

Supports following telephony commands and events (assuming available with TAPI/TSAPI driver and supported by the telephony device)

- Make call
- Answercall
- Put on hold
- Transfer
- Conference
- Get DDI
- Get CLID
- Hang-up
- Connect to Device
- Disconnect from Device
- Pick-up
- Blind transfer
- Swap hold
- Reconnect held call
- Overdial tones (eg for accessing answering machines)
- Full call progress monitoring (if supported by the line and driver used)

Supported sound functions include:
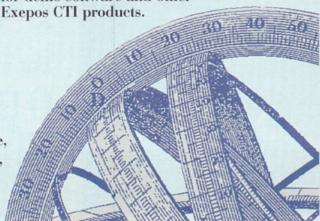
- Open WAV/MIDI files
- Record line to EWAV file
- Playback to line
- Play WAV file
- Play MIDI file
- Rewind
- Set volume
- Stop playing or recording
- Pause
- Resume

Price

TAPI	£350
TSAPI	£595
TAPI/TSAPI	£795

(No run time charges)

Visit our web site www.exepos.com for demo software and other Exepos CTI products.

Exepos Limited, Acorn House, The Straight Bit, Flackwell Heath, High Wycombe, Buckinghamshire HP10 9LS Telephone: 01628 533143 Fax: 01628 533145 email: sales@exepos.com

CIRCLE NO. 808

Designing a network protocol

Peter Collinson had a project involving several computers. They needed to communicate, one way or another. The easiest route to design the protocol was to look at what had been done for the Internet network.

In the last few weeks, I've been constructing what the media calls a client/server application. I hate the term, usually because it's used as a tag with no evident understanding of its meaning. Nevertheless client/server does describe the system I've been writing. The system is to wake up at some point after the end of the working day and move some rather important files from a set of Unix machines onto the filestore of a machine which is a hierarchical storage manager (HSM). The HSM uses its magnetic disks as a cache, backing files up onto a jukebox holding optical media. The HSM presents a huge on-line filestore to the network on which it lives. Unix machines can see the filestore of the HSM over NFS, so the fundamental part of the task can actually be accomplished with the cp or mv commands. Well, that's where the project started.

However, one of the project requirements was that there must only be one master copy of each file. This meant that after a file was copied, it should be automatically deleted from the client machine. At this point, I began to get queasy, uppermost in my mind was the thought of losing files when the unattended program merrily deleted important files in error after the copy had failed for some reason. The hard thing to get right in this type of system is error reporting and also what is to be done when errors happen. If the system is run by a person, then the person can be asked to monitor what is happening and take decisions when things go wrong. This option is not available if the system is to run automatically.

It became apparent that there would need to be a central server program running on the HSM to monitor and control the file copying. The main task of the server is to check for errors and stop any copying if and when errors occur. It's possible to define the sequence of copying actions so that the data is always safe. The strategy for failure is to stop the file copying and send email to a responsible person who will hopefully sort things out the next day.

Of course, if we have a server, then we must have at least a client. The server will perform scheduling functions, controlling the total number of clients that are permitted to copy files at any one time, and it will act as a central logging point for actions from these clients. All transactions for the system will be found in one directory on one system.

The client goes through a well defined sequence of steps to copy any necessary files from its file-system to the HSM. The client keeps the server informed of its progress, sending state change information and logging information that is simply stashed away. The server controls whether the client can enter certain states and this ability is used to reduce contention for resources. In addition, the server supplies the client with all the information that it needs to know about the particular machine it is running on. For example, the client is told where it will find the files that are to be copied. This means that the same client code can be used on all the various client machines. The code is driven by information from the server that is taken from a single configuration file.

If we have a server and a client, then there must be communication between them. A TCP connection is strung between the client and the server to carry data in both directions. The benefit of using TCP is

PLAN DETAIL THRU' SLOTTED UPRIGHT 'D' 1:2

30

25

18 mm MDF PLINTH TOP LAMINATED TO Ref. C17 COQUILLE D'OEUF TEXTURED. BY FOLYREX.

H/W UPSTAND TO TRIM OF PLINTH OAK. 2 PACK SATIN FINISH.

60 x 30 LEG.

19

60

SECTION DETAIL THRU' PLINTH 'E' 1:2

Because there's a nicely consistent structure for commands coming from the client, decoding and deciding what is to be done for each of them is not hard. Most people seem to implement this by translating the four character command into a defined value and then writing one immense routine that consists of a single `switch` statement. I really dislike this approach. It's hard to maintain and actually difficult to make sure that you've got things right. For years, I've been implementing this type of problem by defining a data structure like:

```
typedef struct
{
  char *name;
  int  code;
  void (*fn)(void *,...);
} Cmdtab;
```

You then define a static table that matches the command name to a function:

```
Cmdtab cmdtab[] = {
  { "helo", C_HELO, fn_helo },
  { "quit", C_QUIT, fn_quit },
  ...
  { NULL, }
};
```

You can obviously add extra things to the basic structure as the application dictates. In this project, the server maintains a state variable for each connected client. Some commands can only occur in certain states, so a variable in the table is used to specify a legal state for each command. There is a single piece of code that detects incoming commands that are illegal. The benefit is that you don't need to sprinkle explicit tests for illegal actions throughout the program and the final code is much easier to read.

I tend to implement the sledgehammer approach when using a command table to decode a line from the network. I have a small routine that is passed the line and simply iterates down the table looking for a match using the standard string compare. When a match is found, the service routine is called. This is just a few lines in C.

I've been a little vague about the parameters to the service functions. I usually have two parameters: a pointer to the buffer of network data that has just been decoded and a pointer to the matched entry in the command table. I do sometimes hide the command table details from the service functions and pass some of the data that the table contains for the relevant entry as parameters.

You may be wondering why I've defined a constant for the command and placed it into the table. It's perhaps superfluous. However, it means that you can share service functions

---

that either side can detect catastrophic failure in the other machine or the network, because the connection will simply die. Such direct failure information is very desirable, it allows sensible suspension of activities on the running machines. Hopefully, in turn, this helps to ensure that the entire system is robust.

When you are designing systems, each decision that you take leads you into another problem area. Having come to the conclusion about using a server and a client, it was necessary to consider what data is sent over the TCP connection. Essentially, a raw TCP connection is just a tunnel between two machines. You merrily stuff bytes in at each end, and they emerge somewhat magically at the other, blinking in the sunlight. You need some formalised way of communicating over the link. You are now in the business of creating a protocol. I decided that it wasn't sensible to invent something completely from scratch. I began to investigate extant protocols, looking though the set of documents called 'Request for Comments' or RFCs. You'll find the set in many places on the net, I found mine on *unix.hensa.ac.uk*. I specifically picked SMTP which implements a command/request system which seemed close in essence to what was needed for the job.

Internet protocols

Protocols carry at least two different types of information. They are used to transmit data from the server to the client and *vice-versa*, and they need to send control information to tell when the data is arriving and what type of information is being sent on the link. When sending control information, all

internet protocols use similar formats. Requests for information from the server by a client are sent as commands that elicit a response from the server. Both commands and responses are sent as lines of text, terminated by a 'network end-of-line' sequence: a carriage return followed by a line feed.

The good thing about adopting a character-based protocol is that you can test servers simply by connecting to them with the `telnet` program and typing the data that the server expects to receive. You'll find that `telnet` faithfully sends what you type and prints the responses from the server. It's a healthy bootstrap tool while you write the server code.

Commands and responses have different basic formats. Let's look at commands first. A command from a client is a line that starts with a four character alphabetic string, a space, and some optional text. The control line is terminated by a network end-of-line. The four character command string is case-independent, it can be sent in upper or lower case. In fact, in most protocols, control information is supposed to use only the seven-bit ASCII character set, but this is often relaxed to allow the top bit through.

In this project, I have managed to generate 21 separate commands, six of which are not used by the client but are there to allow an additional control program to interrogate and change the state of the server. The remaining commands are either used from the client to request information from the server; or to send information to the server; or to inform the server that the client is about to enter some new state, saying 'Please sir, is that OK?'.

```
int query(char *cmd, char **ans)
{
  char *res;

  /* send command */
  message(cmd);

  for (;;) {
    /* get line from net */
    res = net_get();

    switch(*res)
    {
    case '2':
      if (res[3] == ' ') {
        *ans = res;
        return RES_OK;
      }
      break;
    case '3':
```

```
      if (res[1] == '5' &&
          res[2] == '4') {
        /* assume no continuation */
        /* for 354 code */
        *ans = res;
        return RES_WANTS_INPUT;
      }
      /* else fall through to */
    case '4':
    case '5':
      if (res[3] == ' ') {
        *ans = res;
        return RES_FAIL;
      }
      break;
    default:
      error("Unknown code: %s\n", res);
    }
  }
}
```

Listing 1 – The minuscule loop needed to decode the three digit response.

between similar commands should this make sense. If you do this, then you'll need some way to determine the command that the service function is dealing with. If the service function is passed a pointer to the command table, then this is a natural place to put the constant used to differentiate between calls.

Another option might be to use the position in the table as the integer code for the command, so for example, the code value for the helo command would be zero, the value for quit would be one, and so on. I find that it's nearly always a bad idea to use the position of data in a table as a constant. First, it's a little magic and might not be picked up by someone who comes after you to maintain the code. Second, it may stop your code being easily changeable later. It's another one of those things I am queasy about.

I hope that you can see that the command table approach is far superior to coding a huge switch statement. It results in a set of very well defined small routines that do the business of servicing the protocol.

It's easily extensible and permits you to place some lengthy complex code in a separate file should you need to do so.

Replies to commands

The golden rule in all the internet protocols is that each command sent to the server from

```
214-This is Sendmail version 8.7.5
214-Commands:
214-  HELO  EHLO  MAIL  RCPT  DATA
214-  RSET  NOOP  QUIT  HELP  VRFY
214-  EXPN  VERB
214 End of HELP info
```

Figure 1 – Here's part of the HELP message from the current version of sendmail that I am using.

the client elicits a reply. The reply comprises a three digit code formed from ASCII characters, a space, an optional string and is terminated by a network end-of-line. The client should be able to understand the response by looking only at the three digit code.

The string should really only convey information intended for humans. It's not supposed to hold anything that the client might need to decode to decide what to do next. I've cheated on occasions in my protocol design. I needed to return some information from the server to the client, and I've passed

it in the reply line. However, I've made sure that the data I wanted to send in the response string is preceded by an unambiguous keyword, so it's easy to detect the importance of the data line.

The three digit codes are not just random numbers picked out of the air by the programmer. There's a well defined method for how they should be constructed. This was first outlined for SMTP in 1982 (see RFC821), and has been adopted by most protocol designers. Some protocols do change the meanings slightly from their original definition.

The three digits are read from left to right. The first digit gives an overall response code, the second says more about the circumstances of the reply, the third is used to sequence responses in the context of the other two digits.

A code of 1xx is a 'Positive Preliminary reply'. The command that the client sent has been accepted but some information needed by the server is not available. The client is supposed to send another command to get a more definite response from the server. I haven't used this code, nor does SMTP. Some protocols, for example NNTP (RFC977), define this code differently. NNTP uses it to mean that the response is an informative message.

A code of 2xx is a 'Positive Completion Reply'. This code is used for the general 'that's OK' type of messages and for messages containing data being sent from the server to the client.

A code of 3xx is a 'Positive Intermediate Reply'. The command has been accepted but the action cannot be completed because the server needs more data. The command is mostly used to tell a client to send data up the line. For example, it's used in SMTP as

```
354 Enter mail, end with "." on a
line by itself
```

to request that the body of the mail should be sent from the client to the server. The client will send lines of text that will not elicit a response from the server until a dot on a line by itself is sent.

The 4xx and 5xx codes carry bad news. A 'Transient Negative Completion Reply' is signified by the 4 and means that the command failed, but the client can try again. The 5 code is a 'Permanent Negative Completion Reply', it means that the transaction failed and the client should give up.

The second digit of the reply code is used to convey information that qualifies the first

TRY PUSHING SOFTWARE WITHOUT AN APPETIZING DEMO AND YOU'RE DOGMEAT.

DemoShield4 gives you the power to effortlessly create convincing point-and-click product demos, presentations and tutorials of your application. Simply use a sample demo or customize your own interactive demo with special effects–all without pro-gramming. DemoShield4 gives your customers a chance to interact with your applica-tion in real time while experi-encing key product features. Let DemoShield4 do the selling for you. After all, it's a dog eat dog world out there.

Order yours today!

DemoShield4
Show and Sell

CIRCLE NO. 811

response. SMTP uses the following meanings. A zero means that there has been a syntax error or the server doesn't understand the command that was sent. A '1' indicates that the response contains some information, such as system status or help. If you telnet to some handy SMTP server and say HELP, you should get a bunch of lines each preceded by a 214 code containing the help file for the server (see below).
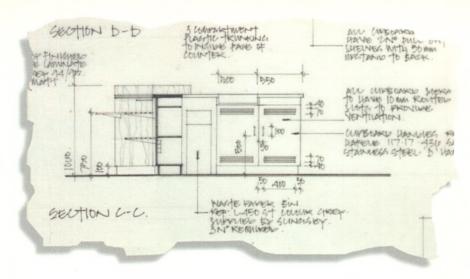
A '2' indicates that the response is a reply referring to the transmission channel. A common code sequence is 221, meaning that the server is closing the transmission channel. The final defined code is '5', which is used for a mail operation, usually the '354 send me mail' response.

Other systems have redefined the use of the second digit. For example one particular specification for PCMAIL (RFC1056) uses: x0x for miscellaneous, x1x for a user operation, x2x for a client operation and x3x for a mailbox operation.

The last digit of the response is used to distinguish between messages of different type that fall into the same category, so it's an index number of the response in that category.

Sometimes the server needs to send several response lines to the client and there has to be some mechanism to indicate that the server is making a multi-line response.

The server indicates that more lines are to come by replacing the space that is normally added after the three digit reply code by a minus sign. See Figure 1.

Decoding loop

Actually this is easy to decode in a client but harder to create, especially if you don't want an extra line at the end of the data such as in Figure 1. I'll guess that the code that sends this is a loop containing:

```
printf("214- %s\r\n");
```

The loop will be followed by:

```
printf("214 End of HELP info\r\n");
```

The coding of the three digit response code probably looks very complicated. It really is not, especially if you set out the responses that you intend to use before you start writing the program. However, I must confess that I was dubious about it all until I wrote the client program and discovered how easy it was to write compact routines to deal with the response codes.

My client needs a routine that says 'send this to the server and give me the result'. The code is actually minuscule: see Listing 1.

I admit to removing some of my code from the example, the deleted code does all the debugging and result testing. The real rou-

tine also picks out the 221 code which tells the client that the server is about to close the connection. In this case, I exit from the routine with a `longjmp`.

In practice, very few calls to the `query` routine need to decode the response that is passed back in the parameters. You end up coding things like:

```
if (query("PATH", &ans) != RES_OK) {
  /* send fail message */
}
```
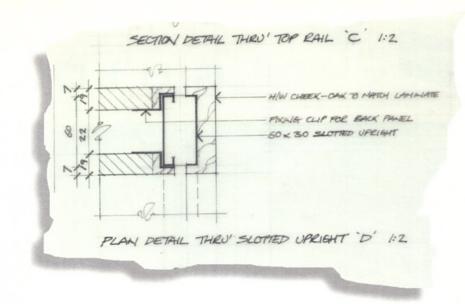
which is very readable and easy to create.

Although the response codes are very comprehensive and ostensibly complex, their decoding is not difficult. You end up with reasonably tight programs to do the job. In fact, for my protocol, I've just taken the values and messages from SMTP and have not defined my own response set. The work of defining the response codes was done for me.

Do not reinvent the wheel

Perhaps the interesting aspect of the Internet protocols is that they are asymmetric. The format of the data that the client sends is different from that sent by the server.

I guess that I find this behaviour a little odd because it's always difficult to determine what is the client and what is the server. For example, in my project, the client sends to the server some text that logs its actions when it receives a 354 code. But the protocol also allows for a program to request the log file to be sent to it from the server, and it's sent as a set of 210 response lines with continuations. The same data is travelling in two different directions using two different mechanisms. This does seem somewhat strange.

On balance, I think that it was a sensible decision to base the protocol design on the standard Internet protocols. It took the grief from the creation process and the code is probably easier to understand by others, especially if those others have looked at how any internet protocol function. ∎

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.

CIRCLE NO. 815

NOW DESIGNING DATABASE FRONT ENDS IS SO EASY IT'S EMBARRASSING.

Introducing Data Widgets 2.0

Data Widgets 2.0

Dramatically increase the productivity of your Microsoft® Visual Basic® 4.0 application development with Data Widgets 2.0 from Sheridan Software. Data Widgets 2.0 is a set of bound OLE controls that lets you design front ends for database applications in the shortest amount of time with all the ease and power you've come to expect from Visual Basic.

Data Widgets 2.0 is designed to optimize your Visual Basic 4.0 database front ends, supplied as 16- and 32-bit OLE Controls. Included are a fully editable DataGrid, a DataCombo (the dropdown and edit portions can be bound to different databases), Enhanced Data Controls, a DataDropDown, a Data Command and a DataOptionSet Control.

Drop-In Replacement for Visual Basic 4.0's grid.

DataGrid™ (SSDBGrid)
- Functionally and visually consistent with data grids in Microsoft Access and Visual Basic 4.0.
- Support for movable groups and columns.
- Optional dropdowns in headings allow users to select from a list of available fields and/or groups at run time.
- Additional cell types include checkbox, button, label, and combo box.
- Multiline row formats.
- Pictures and text in cells and headings.
- Individual fonts and colors for columns, rows, and cells.
- Full design time capabilities.
- Supports multiple data modes including bound, unbound, and AddItem (design and run time).

Data Combo (SSDBCombo)
- Variable edit area height a la Microsoft Access.
- Multiline edit area.
- DropDown display is not limited to the width of the edit area.
- Same formatting capabilities as the SSDBGrid control.
- Full design time capabilities.

DataDropDown (SSDBDropDown)
- Used in conjunction with the DataGrid.
- DropDown display is not limited to the width of the edit area.
- Same formatting capabilities as the SSDBGrid control.
- Full design time capabilities.

Data Option Set (SSDBOptSet)
- One control creates an unlimited number of option buttons that are bound to the same data field.
- Multi line captions and pictures.
- Automatic and manual button positioning.
- Custom color options.

Enhanced Data Control (SSDBData)
- Re-position recordset by selecting from a drop down list of up to 100 previously marked rows.
- Store and sort multiple bookmarks in the bookmark drop down list.
- Buttons that perform database actions such as add, delete, update, plus bi-directional nth record paging.
- Conditional and Soundex searching.
- ToolTips

Data Command (SSDBCommand)
- Multiline captions and pictures.
- Add, Delete, Refresh, Bookmark, and Auto-Positioning functions.
- Click and AfterClick events.
- Auto-Repeat functionality.
- Custom color options.

Tel: 07000 422 224

Visual Basic Programmer's Journal '95 Readers' Choice

Thunderbolt Award Winner

Sheridan™

Reusable Components and Productivity Tools for the Visual Developer

List Price:
£99.00

✎ CIRCLE NO. 816

CONTEMPORARY *software*

Comtemporary Software Ltd
Kingswick House, Sunninghill, Berkshire SL5 7BH
Tel: 07000 422 224 • Fax: 07000 422 225
E-Mail: cssales@contemporary.co.uk

Microsoft Developer Training

At Pygmalion, you will get

all the benefits of training from a foremost **Authorised Technical Education Centre.** Courses are conducted by instructors who are certified as **Microsoft Systems Engineers and Microsoft Solution Developers.** In addition each has achieved a level of technical excellence and practical experience within a blue chip corporate environment.

Classes are restricted to a maximum of **eight delegates.** Courseware follows the **Official Microsoft Curriculum** and therefore is designed to support the **Microsoft Certification Programme.** Fast track, tailored and on-site courses are also available.

For further information or detailed course overviews, please call **0171 436 2254.**

Courses offered include

Upgrading from Visual Basic 3.0 to 4.0	3 days	£795
Fundamentals of Visual Basic 4.0	3 days	£795
Advanced Programming in Visual Basic 4.0	5 days	£1195
Introduction to Visual C++ and Object Oriented Programming	5 days	£1395
Fundamentals of Programming MFC Library Applications	5 days	£1395
Intermediate Windows-based Programming Using MFC Libraries	5 days	£1395

Microsoft
SOLUTION PROVIDER
Authorised Technical Education Centre

We also offer training for **Borland Delphi**

Pygmalion

getting closer to technology

79 New Cavendish Street, London W1M 8N
Telephone 0171 436 2254 Fax 0171 436 224
Email enquiries@pygmalion.co.u
Web http://www.pygmalion.co.uk

Of coloured necklaces

> The most efficient manner to keep information confidential is to encrypt it. **W. Johnson** shows how bytes can be threaded on a necklace that only you can appreciate.

iles held on your PC are easily viewed and to copied by anyone who has physical access to the computer. Enciphering important and confidential files is a way to prevent unauthorised inspection or modification by others. Most enciphering techniques are based on number theory, and rely on the difficulty of factoring huge numbers with unknown factors in reasonable time. The mathematics involved has to generate powers of numbers and find the modulus of the answers, but these techniques are not ideally suited to the PC because of the amount of calculation involved. In this article, we will outline a private key encryption technique based on *de Bruijn sequences* which is as effective and very easy for a PC to handle.

De Bruijn sequences are named after Professor Nicolaas G. de Bruijn of the Technical University at Eindhoven in the Netherlands. De Bruijn studied combinatorial analysis and advanced graph theory. A graph is any set of joined up points and could equally well be the edges of a crystal, a printed circuit board or the garbage collection routes in New York.

Sequences

In studies of graphs representing chemical molecules, geometric figures, mathematical tree structures etc. sequences of numbers are frequently associated with the labels on the graph vertices or edges. These are important when similar graphs need to be distinguished or enumerated. For example, the eight adjacent triplets in the sequence 00011101, when arranged in a circle, ie 000, 001, 011, 111, 110, 101, 010, and 100 represent a path between the corners of a cube touching each only once. The sequences are very compact and every digit is part of several larger groups. The 16-digit sequence 1001101000010111 when arranged in a circle has 16 four-digit binary numbers hidden in it representing the decimal numbers 0 to 15. Hence the expression:

```
Hexa$ = MID$("936DA480125B7FEC",
    INSTR("1001101000010111100",
    Bin$), 1)
```

is a very compact binary to hexadecimal conversion function.

The simplest way to envisage a de Bruijn sequence is to think of a necklace of coloured beads in which every arrangement of a given number of beads and a fixed number of colours can be found once and once only. This is explained in more detailed in Figure 1.

Two properties of de Bruijn sequences make them a good choice for small computers with limited processing power.

- With a reasonable alphabet, we are limited to relatively small word lengths otherwise the sequences become too big to handle. If $(m + 1)$ is the number of letters in the alphabet (or the number of colours in the necklace) and n is the word length then the length of an (m, n) de Bruijn sequence is given by: $(m + 1)^n$.
- The number of sequences for a given m and n also rises extremely rapidly. It is

---

De Bruijn sequences can be represented by a necklace of coloured beads. For example, if we take the three colours red, blue and green which we can designate as R, B and G, then the de Bruijn sequence:

RRRRBRRRGRRBBRRBGRRGBRRGGRBRBRGRBBBRBBGRBGBRBGGRGRGRGBBRGBGRGGBRGGGBBB–BGBBGGBGBGGGG(RRR......) where the bracket is a repeat of the sequence as it is circular, contains the 81 possible arrangements of four beads in three colours ie

RRRR	RRRB	RRRG	RRBR	RRBB	RRBG	RRGR	RRGB	RRGG	RBRR	RBRB	RBRG
RBBR	RBBB	RBBG	RBGR	RBGB	RBGG	RGRR	RGRB	RGRG	RGBR	RGBB	RGBG
RGGR	RGGB	RGGG	BRRR	BRRB	BRRG	BRBR	BRBB	BRBG	BRGR	BRGB	BRGG
BBRR	BBRB	BBRG	BBBR	BBBB	BBBG	BBGR	BBGB	BBGG	BGRR	BGRB	BGRG
BGBR	BGBB	BGBG	BGGR	BGGB	BGGG	GRRR	GRRB	GRRG	GRBR	GRBB	GRBG
GRGR	GRGB	GRGG	GBRR	GBRB	GBRG	GBBR	GBBB	GBBG	GBGR	GBGB	GBGG
GGRR	GGRB	GGRG	GGBR	GGBB	GGBG	GGGR	GGGB	GGGG			

If you change 'coloured beads' to 'letters', 'fixed number of colours' to 'alphabet' and 'every arrangement of a given number of beads' to 'words of a given length' then you can say that this de Bruijn sequence contains all the four letter words that you can make from an alphabet containing only B, G and R

Figure 1 – A de Bruijn sequence.

"Sure, NT's the wave of the future...

but how do we get there from here?"

Moving to Windows NT is a crucial step for your company. But how do you give up UNIX without sacrificing your budget, timelines and quality? Making the transition to a new development environment means changing and checking thousands of lines of code, usually without the necessary tools or training.

Essential development tools for programmers

The move from UNIX to NT doesn't have to be expensive or time consuming. MKS customers do it everyday with MKS Toolkit. Rely on the KornShell on NT to move your UNIX code over with minimal changes. Only MKS Toolkit's tape utilities allow you to read all your archived information onto your new NT machine, giving you the power to access and use stored information.

The power of UNIX — on your PC

Fortune Magazine's top listed companies all use MKS Toolkit. Discover how MKS Toolkit can help you leverage your investment in your valuable code base. MKS Toolkit 5.1 is a comprehensive suite of 190+ software development utilities for the PC. You'll find powerful new tools such as graphical scripting for Win32, customizable toolbars for Windows 95, and enhanced NT security support.

Now Features:

* Graphical scripting for Win32
* Customizable toolbars for Windows 95
* Enhanced NT security
* Native 32-bit utilities
* CD ROM

"MKS Toolkit turns Windows NT into a world-class set of commands and utilities. I wouldn't run without it."
- Tom Yager
Open Computing Magazine

Call today!

0171 624 0100
http://www.mks.com
Email: uk@mks.com

MKS
MORTICE KERN SYSTEMS INC.

Mortice Kern Systems (UK) Ltd
239 Kilburn Park Road
London NW6 5LG
Fax: 0171 624 9404
Alternatively, contact:
System Science
Tel: 0171 833 1022
Grey Matter
Tel: 01364 654 100
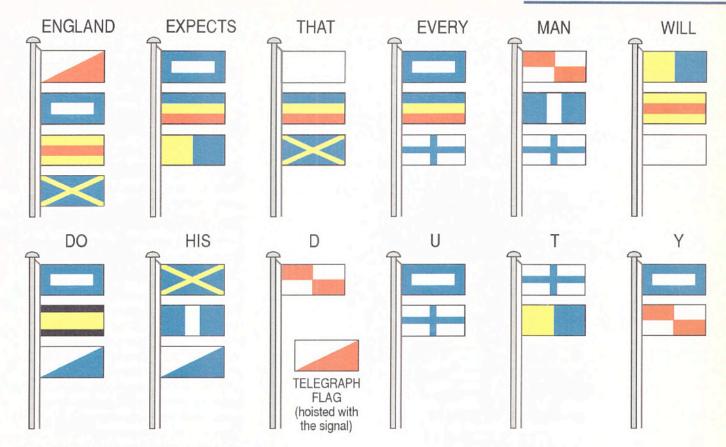Admiral
Tel: 0276 692 269

MKS Toolkit. A developer's best kept secret.

Now supports Windows 95! (also available for DOS and OS/2)

Your challenges are our challenges. All people depicted in this ad are MKS personnel.

CIRCLE NO. 818

Figure 2 – Nelson's signal at the Battle of Trafalgar.

calculated from: $(m + 1)^{((m + 1)^{(n-1)} - n)}$. See Table 1. Herein lies the security of the system as the brute force method of code cracking is going to have some difficulties finding the specific sequence out of all the possibilities.

Keep a secret

To make a private key enciphering system using a de Bruijn sequence involves the following steps:

1. Chop the file into short strings say two or three bytes long.
2. Make a de Bruijn sequence from a suitable alphabet and a word length which matches the word length in step 1.
3. Find the position of each short string in the de Bruijn sequence and make the enciphered string by concatenating the short strings a known distance further around the sequence.
4. Deciphering is the reverse of enciphering.

The known distance can be fixed or variable and it is easy to add redundant characters during the enciphering which can be ignored on deciphering. This would further complicate any attempt at brute force deciphering. To illustrate the above steps, we can choose

an alphabet of a space, a comma, a full stop, a colon, a semicolon and 26 capital letters. This gives an alphabet of 31 letters and, with a word length of 3, a de Bruijn sequence of length 29,791 which can be handled as a single string. There are 31^{958} sequences to choose from (which is approximately 5.3 x 10^{1428}) so there should be no difficulty find-

$N = (m + 1)^{((m + 1)^{(n-1)} - n)}$ where $(m + 1)$ is the size of the alphabet, n is the word length.				
m = 1	n	Length	N	N
	2	4	2^0	1
	3	8	2^1	2
	4	16	2^4	16
	5	32	2^{11}	2048
	6	64	2^{26}	67108864
	7	128	2^{57}	144115188075855872
	8	256	2^{120}	1329227995784915872903807060280344576
m = 2	2	9	3^1	3
	3	27	3^6	729
	4	81	3^{23}	94143178827
	5	343	3^{76}	1824800363140073127359051977856583921
	6	729	3^{237}	~1.196 x 10^{113}
m = 3	2	16	4^2	16
	3	64	4^{13}	67108864
	4	256	4^{60}	1329227995784915872903807060280344576
	5	1024	4^{251}	~1.3094 x 10^{151}
	6	4096	4^{1018}	~7.8899 x 10^{612}
m = 25	2	676	26^{24}	9106685769537214567998140360 94976
	3	17576	26^{673}	~1.8926 x 10^{952}
m = 127	2	16384	128^{126}	~3.2245 x 10^{265}
m = 255	2	65536	256^{254}	~4.9312 x 10^{611}

Table 1 – Numbers of de Bruijn sequences.

TECHNIQUES

```
'Program to generate a de Bruijn sequence for m=30, n=3 (SeqLen = 29,791)
'--------------------------------------------------------------------------
DEFINT A-Z

DECLARE SUB MakeSequence (FileName$)

CONST L1 = 961                          '(M + 1)^(N - 1)
CONST L1M1 = 960                        'L1 - 1.
CONST M = 30                            'Alphabet size minus 1.
CONST N = 3                             'Word length.

Main:
'~~~~
INPUT "File name for sequence"; FileName$
CALL MakeSequence(FileName$)

DEFSNG A-Z
'----------------------------
SUB MakeSequence (FileName$)
'----------------------------
DEFINT A-Z

REDIM Ptr(0 TO L1M1)                    'Holds pointers to alphabet.
REDIM Z$(0 TO 30)                       'Holds total alphabet.
A = 0
DO WHILE A < L1
    Ptr(A) = M                          'Fill Ptr() with 30's.
    A = A + 1
LOOP
Z$(0) = CHR$(32)                        'Space.
Z$(1) = CHR$(44)                        'Comma.
Z$(2) = CHR$(46)                        'Full stop.
Z$(3) = CHR$(58)                        'Colon.
Z$(4) = CHR$(59)                        'Semi-colon.
FOR P = 1 TO 26
    Z$(P + 4) = CHR$(64 + P)            'The 26 letters of the alphabet.
NEXT
OPEN FileName$ FOR BINARY ACCESS WRITE AS #1
    Count = 0: Cycle = 0: D$ = "": J = 0
    DO
        DO
            X = Ptr(J)
            D$ = D$ + Z$(X)             'Or PRINT D$;
            Ptr(J) = Ptr(J) - 1         'New pointer value.
            J = ((M + 1) * J + X) MOD L1    'Next J.
            Cycle = Cycle + 1
        LOOP UNTIL Cycle = 512 OR Ptr(0) = -1
        PUT #1, 512& * Count + 1, D$    'Save a block of 512 bytes.
        Cycle = 0: D$ = ""
        Count = Count + 1
    LOOP UNTIL Count > 58
CLOSE #1

END SUB
```

Program 1 – Program to generate a de Bruijn sequence for m=30, n=3 .

```
Pythagoras's Theorem a≈ = b≈ + c≈ is a special case of the Cosine Rule
a≈ = b≈ + c≈ - 2bcCos(A) where the angle A is 90°.

I wander'd lonely as a cloud that floats on high o're vales and hills,
when all at once I saw a crowd, a host, of golden daffodils.

+----------------------------------------------------------------------+
ı Anything goes! Including the lines there are just 219 characters here.ı
+----------------------------------------------------------------------+

pi'ck-a-back, *pi'ggyback (-g-). 1. adv. (Carried) on back or top of a
larger person or object. 2. n. Ride thus given. [orig. unkn.]

You can use File Manager to start your applications. You start an
application by opening a program file or document from a directory
window. Program files always have a .COM, .EXE, .PIF or .BAT filename
extension.

ISRAEL ı 00 972 (This code also gives access to West Bank, Gaza Strip,
Golan Heights and East Jerusalem) ı Followed by Afula 6 Ashdod 8
Ashkelon 7 Bat Yam 3 Beer Sheva 7 Ben Gurion Airport 3

Dear Bill, Wish you were here. Weather passable, accomodation fair,
food good and swimming excellent. Company unbelievable but wine superb.
```

Table 2 – information which generates keys for enciphering.

ing one! Program 1 generates the lexically largest sequence ie the one that starts with *ZZZ* and finishes with *space space space*.

If we encipher the well known message depicted in Figure 2, which first appeared in an entirely different code, with this de Bruijn sequence we obtain:

NEGDNJ:E;O;XTQQTGJNSRVD,MXXDNK WHPPIKLGNM:SGSSTVZS

To decipher this rather cryptic message we just need to run the Program 2 with a simple shift of 100:

ENGLAND EXPECTS THAT EVERY MAN WILL DO HIS DUTY

Getting practical

To make a viable program to encipher and decipher however, requires the ability to make a de Bruijn sequence known only to you. To do this we must look at the way de Bruijn sequences are generated. The excellent algorithm by Xie (published in 1987 in Notes on De Bruijn Sequences, *J. Disc. App. Math.* 18) employs a table hopping technique. Program 1 is a sample implementation.

A table of $m + 1$ pointers (0 to m) is set up with $m + 1$ columns and $(m + 1)^{(n-1)}$ rows giving $(m + 1)^n$ pointers to the alphabet which is held in the `Z$()` array. Starting with `J = 0`, the first character of the de Bruijn sequence is `Z$(X)` where `X` is the pointer value in the last entry in the `J`th row. This entry is removed from the table and `J` is transformed using the line:
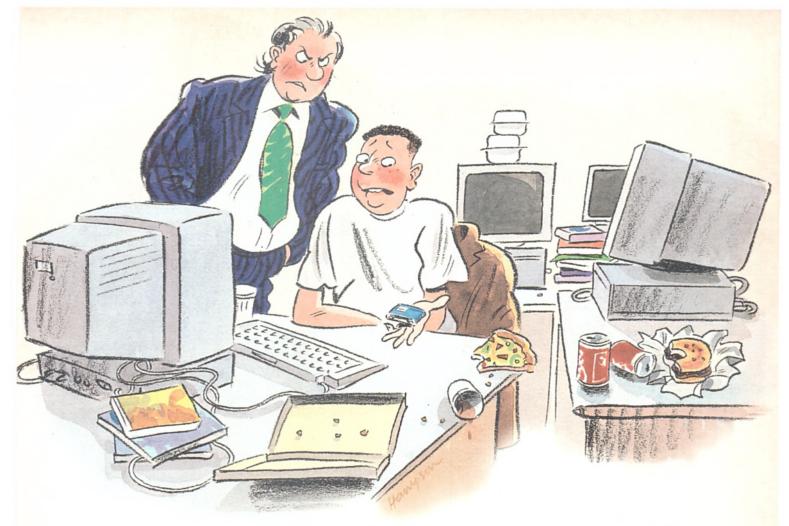
```
J = ((m + 1)*J + X) MOD L1
```

where `L1` is the number of rows. This process has to be repeated in order that every entry in the table is visited once and once only and that `J` finishes up as zero. This condition is governed solely by the entries in column zero. Each row in the table is some permutation of the pointer set (0 to m) and each column some arrangement of `L1` pointers from the same set, but column zero always begins with a zero.

If the table can be traversed as above then its left-hand column is called a *Label(m, n)*. A column of zeros is always a *Label* and gives the lexically largest sequence as demonstrated above. A column of n zeros separated

A Label is a sequence of numbers from the set 0 to m: $0t_1t_2t_3t_4...t_{L1-1}$ such that for each j between 1 and $L1 - 1$ there must exist another sequence $j_1j_2j_3...j_s$ where $s <= L1$, $j_1 = j$, and $j_{k+1} = ((m + 1)*j_k + t_{jk})$ $MOD L1$ which has j_s equal to zero (ie $j_s = 0$ and $j_k <> 0$ where $k < s$)

Figure 3 – Is it a Label?

I can't crack this one, DESkey security is in the hardware

If your product is worth protecting, then it is worth stealing.

Protected applications are constantly under organised attack and it is the determined "Hacker" who is most likely to do the damage.

To add insult to injury, time and money spent on second rate software protection is as good as lost along with sales revenue.

DESkey security is built into the hardware. ASIC's 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.

Don't just Dongle it – DESkey it

PC   MAC   PC Card   UNIX etc

Ready for **OS/2.WARP**

ACS ASIC

Member of **FAST** Promoting the legal use of software

Mac™ OS

PC CARD

D DESkey

CIRCLE NO. 819

```
'Enciphering Program Using de Bruijn Sequences.
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
DEFINT A-Z

DECLARE FUNCTION Exist% (F$)

LINE INPUT "Enter the file name and path of the de Bruijn sequence "; FileName$
IF Exist(FileName$) = 0 THEN
    OPEN FileName$ FOR BINARY ACCESS READ AS #1
        DB$ = "": P = 1
        LenFile = LOF(1)
        NumBlock = LenFile \ 512       'Break file into 512-byte blocks.
        FileTail = LenFile MOD 512     'Bytes left over.
        Temp$ = STRING$(512, 0)
        DO
            GET #1, P, Temp$
            P = P + 512
            DB$ = DB$ + Temp$
            NumBlock = NumBlock - 1
        LOOP UNTIL NumBlock = 0
        Temp$ = STRING$(FileTail, 0)
        GET #1, P, Temp$
        DB$ = DB$ + Temp$
        Temp$ = STRING$(2, 0)          'Add wrap-around.
        GET #1, 1, Temp$
        DB$ = DB$ + Temp$
    CLOSE #1
ELSE
    PRINT "Can't find file"
    END
END IF

PRINT "Enter a message (letters, spaces, commas, colons and full stops only)"
LINE INPUT ; Message$
Message$ = UCASE$(Message$)
L = LEN(Message$)
Code$ = ""
FOR P = 1 TO L STEP 3
    X$ = MID$(Message$, P, 3)
    Z = (INSTR(DB$, X$) + 100) MOD 29791
    Code$ = Code$ + MID$(DB$, Z, 3)
NEXT
```
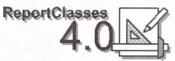
```
PRINT : PRINT "Coded message is "; Code$
Message$ = ""
FOR P = 1 TO L STEP 3
    X$ = MID$(Code$, P, 3)
    Z = INSTR(DB$, X$)
    IF Z < 101 THEN
        Z = Z + 29791
    END IF
    Message$ = Message$ + MID$(DB$, Z - 100, 3)
NEXT
PRINT : PRINT "Deciphered code is "; LEFT$(Message$, L)


'-----------------------------
'Test for existence of file.
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

DEFINT A-Z

DECLARE FUNCTION Exist% (F$)
COMMON SHARED /ErrorCode/ Errcd

END

REPORT.ERROR:
Errcd = ERR
RESUME NEXT

'--------------------
FUNCTION Exist% (F1$)
'~~~~~~~~~~~~~~~~~~~~~

DEFINT A-Z

Errcd = 0
Ch = FREEFILE
ON ERROR GOTO REPORT.ERROR
OPEN F1$ FOR INPUT AS #Ch
CLOSE #Ch

Exist = Errcd

END FUNCTION
```

Program 2 – Enciphering program using de Bruijn sequences.

Open up Hoskyns Training Directory.

Open up unlimited opportunities for Client/Server.

- Architecture and Design
- Managing Client/Server Projects
- Oracle, Sybase, Ingres, MS Exchange
- SQL Windows, Visual Basic, Power Objects, C++
- UNIX, OS/400, Windows
- End User and Implementation Training
- Internet

The Hoskyns Training Directory has over 120 courses each designed and delivered to help you to exploit the opportunities that Client/Server will bring to your organisation.

To receive YOUR FREE COPY telephone 0171-830 6830, fax 0171-630 5581 or complete the coupon opposite:

Please send me my FREE copy of the current 1996/7 Hoskyns Training Directory.

Name

Job Title

Company

Address

Postcode

Phone

Fax   AEX 90

Hoskyns Group, Glen House, Stag Place London SW1E 5AG.
Visit us at http://www.hoskyns.co.uk

HOSKYNS
CAP GEMINI SOGETI

CIRCLE NO. 821

by *n - 1* groups of *m*'s (where each group contains $m(m + 1)^r - 1$ times *m*'s with *r* varying from *0* to *m - 2*) is also a *Label*.

(So, for example, for m = 2 and n = 4, 0222222222222222222020222222020 is a Label).

But in general, it is necessary to test if any specific arrangement which is generated by the routine qualifies as a Label and this is done in the first part of Xie's algorithm (see Figure 3). Only $1/(m + 1)^{(n - 1)}$ of all possible arrangements qualify

So, if we choose an alphabet of 256 ASCII characters, and a word length of 2 we need a pointer table of 256 x 256 to give the de Bruijn sequence of length 65,536. The zero column is then some 255 digit number to any base up to 256 and this becomes the private key for the system.

What the number is needs to be kept secret but in the words of Sir Humphrey in 'Yes Minister' the best way to keep a secret is not to tell anyone you have one so the number should be derived from widely disseminated information available to all such as telephone directories, dictionaries, instruction manuals, common books or newspapers. A short paragraph of 20 to 30 words will yield a set of numbers which will give the starting point for the algorithm to search for a Label.

Randomise

One more property of the de Bruijn sequence can be used to advantage and that is its ability to generate pseudo-random numbers. Take a binary sequence of word length 4, for example:

0101000011011110

then it is possible to generate the following sequence of numbers:

0, 5, 1, 10, 7, 6, 3, 8, 13, 12, 14, 9, 11, 15, 2, 4

by using $Seed_{new} = Value$ at position $Seed_{old} + 1$, starting with $Seed_{old}$ as zero. Not all sequences do this but it is usually possible to rotate most sequences to give the full cycle of numbers. In our application, a binary de Bruijn sequence of word length 8 needs to be made and this will give a set of 256 random pointers for the alphabet. Column zero for that table can be derived from the first few words of the key.

The attached list (Table 2) shows a number of keys that have been tested and they all work well illustrating the point that practically anything will do if you can type it in.

The whole program needs protecting with a pass phrase (50 to 125 characters) which can conveniently be turned into a 16,384 long sequence to encipher your Labels and the file and path information for saving them. Other users then cannot read your Label file names or decipher your keys and vice versa.

The current version of the program can handle any type of file which is less than 16 MB long. On a 386-20 PC, it can encipher at 7,500 characters per second and decipher at 3,000. Facilities for viewing files, changing the keys and deleting confidential files (first changing the contents to CHR$(255) before deleting) are also available. If no redundant characters are added, the enciphered file is a binary file two or three bytes longer than the original file. The deciphered file is exactly the same as the original. ∎

William Johnson is a physicist. Bill graduated from Cambridge and spent his working life in the steel industry working in diverse areas such as crystallography, physical methods of chemical analysis, electron optics, instrumentation and product inspection. He received an MBE on retirement and since then has concentrated on developing efficient algorithms for a number of practical problems. Bill can be contacted by phone or fax on 01709 531 349.

The program can be downloaded from Cix, and from EXE's FTP site at ftp://ftp.exe.co.uk/pub/exestuff/bruijn.
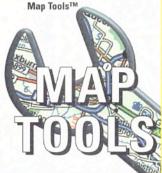
Portability everyone can agree on

"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. 824

Stockholm by night

The Stockholm Standard Committees meeting was much more fruitful that the previous one. Francis Glassborow reports on the progress made.

The joint ISO/ANSI committees for standardising C++ meet every four months for a week of intensive work consolidating all that has been done via electronic communications. Any substantial change to the Working Paper (the document that will eventually become an International Standard) must be supported by a written paper explaining both the problem and the proposed correction. These papers together with other minor issues are distributed to a number of specialist work groups – seven at the most recent meeting.

Each work group contains specialists in a specific area of the language whose role is to consider the merits of the proposed solutions. They may modify the proposal and issue a revised paper for immediate consideration. If they think the solution has sufficient merit it is presented to the joint committees who decide by straw vote whether there is sufficient consensus to take the issue to the formal vote at the end of the week.

It is almost impossible for a paper to get through a work group without an active sponsor present. There is too much work to do. This is unfortunate because it makes it hard for technical experts to contribute any work unless they can attend the meeting. National delegations tend to be small and more than once good ideas have fallen by the wayside for the lack of a sponsor – someone who understands the subject, is able to cope with expert discussion and is respected by other committee members.

Even though the official part of the week finishes around lunch on the Friday, work continues for a group of volunteers till late Saturday evening to give a chance to the official editor, Andy Koenig of *Traps and Pitfalls* fame, to complete the job as well as earn his living. The various hotels where we have met remain bewildered by the way that some will still be working at 3am. They are even more surprised when they discover that we are not being paid for the job. Ten or more pay their costs directly from their own pockets.

Stockholm

The March (Santa Cruz) meeting of the C++ Standards Committees was dominated by the issue of separate template compilation (see *EXE* May '96 p.49). We left that meeting with the issue unresolved and many participants felt a sense of despondency. We had only avoided complete breakdown by becoming politicians and brokering a holding exercise.

A great deal of work has been done in the four months leading up to the Stockholm meeting. The whole C++ community owes a large debt of gratitude to John Wilkinson and his colleagues at SGI for developing a more refined approach to template compilation that promises to make separate template compilation acceptable to implementers. We seem to be down to a single technical problem, the so called intermediate context problem. This comes about when a template instantiation in one file causes the definition (in a different file) to instantiate another template in a third file. The problem is knowing how much information needs to be kept in order to allow correct name lookup at the time each template is compiled. Name lookup problems have been a continuing feature of C++ and it is hard for the ordinary programmer to understand how big a problem results from allowing the same name to refer to more than one thing.

Features like overloading and scoped names are important elements of C++, ones that are deep in the foundations of the language.

Names in templates are of two main types:

- independent names – ones that should be bound in the context of the definition and do not rely on the template parameters, and
- dependant names – ones that can only be bound in the context of an instantiation where the details of the template parameters are known. The instantiation context must take into account the contexts of each of the template parameters.

John Wilkinson and colleagues developed a new set of rules to determine which names in a template definition should be considered as independent names. They did much more. The result is that many implementers think that they may be able to deliver the long promised separate template compilation with a performance acceptable to users. One cost will be yet another keyword: `export`.

Fortunately, the ordinary programmers will not need to understand all the technicalities that underpin the language. They only have to be able to use the tools that are eventually delivered. However they do reap considerable benefit from all the work that goes into standardisation. Programmers need portable skills. English is independent of the word-processor you use. In the same way the basics of a computer language should be independent of the compiler you use.

The guillotine

This term is often used to refer to limiting the time for debate. The WG21/X3J16 Committees effectively applied the guillotine to language fixes as the end of the Stockholm meeting. This caused a degree of focused activity that even threatened the sponsor's reception (a very enjoyable boat trip and meal). Not only have technical issues to be thrashed out but the results have to be encapsulated in formal motions which eventually lead to appropriate changes to the working paper.

TECHNIQUES

By the time the last computer was switched off late on Saturday the Working Paper was almost ready for review by the editing group. This is the process by which we attempt to catch unintended changes.

We believe that we are near enough to the final product to dispatch it to National Bodies as a Committee Draft. We are taking one last look for major flaws. We expect to vote out a CD during the next meeting (Hawaii). Apart from minor work (we hope) in response to public comment that CD will eventually become the International Standard some time in 1998. Then we will have to go into handling default reports (bug fixing).

Significant changes

Several relatively minor improvements to C++ squeezed past the Stockholm guillotine.

Most were refining what was already there, or rewording text to say what we had meant rather than what had actually been written. In some places we strengthened guarantees to programmers. For example we have limited uses of **new** to the programmer and the Standard Library. The intended effect is that implementers may not use **new** to place temporaries on the heap (they may use some other method to get memory for temporaries). Strictly speaking implementers could still surprise programmers with Standard Library uses of **new** but locking that door would take too long.

There were, in my opinion, two items that directly impinge on ordinary code. As a result of a paper from Kevlin Henney (a UK based expert) both string literals and wide string literals will have **const** types. It is a shock to some programmers to discover that they are not already **const**.

The other item was that, after over two years of work in the Core 1 work group (the one I regularly attend) we have substantially reworked the section on **extern "<language specification>"** to provide better support for all aspects of linkage across languages.

I will deal with both these issues in future columns.

Last month's problem

This was last month's problem.

```
#include <iostream.h>
class T {
    int i;
    T(const T& t): i(t.i) {}
public:
    T(int j): i(j) {}
    void print () { cout << i << endl; return; }
    // rest of interface
};

int main() {
    int k = 5;
    T t = k;
    t.print ()
    // other code
}
```

95% of the books and most course presenters tell you that initialisation with an '=' is just the same as the alternative (T t(k) in this case). They are wrong. The former uses a copy constructor which all but the worst compilers will optimise out. However before optimising out a copy constructor, a compiler must check access. In this case the copy constructor is **private** so T t = k; fails. I think this is stupid and that T t = k; should have been defined in terms of T t(k); , however it is too late to change the grammar of the language so you will have to learn to live with it. The same problem will arise where a copy constructor has been qualified with **explicit** (that is when your compiler catches up with the language) so you will have to learn to write:

```
auto_ptr<T>(new T);
```

instead of the more natural:

```
auto_ptr<T> = new T;
```

This was the second problem:

```
class U {
    int i;
public:
    U(int j): i(j) {};
    operator int () { return i; }  //conversion to int
    // rest of interface
};
class V {
    int i;
public:
    V(int j): i(j) {};
    operator U() { return U(i); }  //conversion to U
    //rest of interface
};
int main() {
    V v = 5;
    T t = U(v);
    t.print();
    //other code
}
```

This code fails for a slightly more subtle reason as it requires two user defined conversions to go from a U to a T via an int so it fails even if the copy constructor is public. However look and see what happens when we thoughtlessly write it in the other form:

```
T t (U(v));
```

That, believe it or not, is a declaration of a function t taking a U parameter by value and returning a T by value. We have either got to write it as T t((U) v); or as T t ((U (v))); The latter version is particularly weird because we add an apparently redundant set of brackets to cancel the redundancy of an apparently non-redundant set.

This month's problem

The C++ standard authorises a compiler to optimise out copy construction whenever it can 'prove' that either the original or the copy is not subsequently used. Unfortunately there are some odd uses of objects where the intended subsequent 'use' is the call of the destructor. Look at the following code and decide how you can change it so that the compiler cannot optimise away the copy. Ideally this fix should not cost you anything in code size or speed.

```
void foobar () {
    Mytype mt;
    //code using mt
    Mytype nt (mt);  //copy can be optimised
                     //away reusing mt
    //more code. but no
    //subsequent use of mt
};
```

Dates for your diary

Put a note in your diary to keep July 18th & 19th 1997 free. I'll tell you later what is going on then, but if you are a serious user of C or C++ you will not want to miss out. ∎

How much **time** are you spending learning your configuration **management system......**

instead of actually using it?

You've made the decision to use software configuration management (SCM). And you know that a streamlined development process can help you meet your deadlines and make your developers more productive. So why are you still figuring out how to use the software? You need an SCM tool that produces results instantly.

MKS Source Integrity

At last, there's a software configuration management system that eliminates the learning curve. MKS Source Integrity orchestrates your entire development process, providing your team with a choice between a stand alone GUI and command line interface, across 20 different development environments. Color visual merging and differencing show changes to files at a glance. Sandbox environments provide each developer a safe place to do their daily work.

Thinks and works like development teams do

MKS Source Integrity is a complete SCM system designed to maximize your team-based development. More than 45,000 developers worldwide use MKS Source Integrity's complete suite of tools to help them increase productivity, protect software assets, and guarantee overall source code integrity.

"Would you like to know how MKS Source Integrity integrates with Microsoft C++? One word: seamlessly."
— LAN Times, August 95

Source Integrity provides full integration with leading development environments including:
- Borland C++
- Borland Delphi
- Microsoft Visual Basic
- Microsoft Visual C++
- PowerBuilder
- Watcom C++

Call today!
0171 624 0100
http://www.mks.com
Email: uk@mks.com

To find out how MKS Source Integrity can accelerate your team's development, call today for your FREE demonstration copy, or download it from our website.

Your challenges are our challenges. All people depicted in this ad are MKS personnel.
30-day unconditional money back guarantee.
MKS, MKS Source Integrity and Sandbox environments are registered trademarks of Mortice Kern Systems Inc. All other trademarks acknowledged.

MKS
MORTICE KERN SYSTEMS INC.

Mortice Kern Systems (UK) Ltd
239 Kilburn Park Road
London NW6 5LG
Fax: 0171 624 9404
Alternatively, contact:
System Science
Tel: 0171 833 1022
Grey Matter
Tel: 01364 654 100
Admiral
Tel: 0276 692 269

CIRCLE NO. 825

Java Workshop

Sun had planned to wait before releasing a fully graphic development tool for Java. **Jon Vogler** explains why Sun changed its mind and the result of this change.

There is much debate about the impact of Java and the network computer on users of standalone PCs. For companies the picture is clearer. Compared to PCs, the network computer offers cost reductions and improvements in managerial control. The obstacle to its widespread adoption (in addition to the minor point that the hardware barely exists) is the need to write a new generation of application software that will run on corporate intranets. This is a massive opportunity for the application development industry. The first necessity, already happening, is to train programmers to use Java. The second is effective, productive, software development tools. Sun invented Java and has been a major force behind both the Internet and Intranets, so a good starting point for examining Java development tools is its Java Workshop (JWS).

The original Java Development Kit (JDK) provided only a command-line interface. JWS Version 1.0 was originally planned as a base toolset, with the Visual component to be added later, in Version 1.1. Sun published a beta version of JWS 1.0 on the Web and the package has had between 4,500 and 5,000 downloads a week. The feedback voiced a strong demand for immediate publication of the GUI builder. So, despite bugs and incomplete documentation, it was included in Release 5 of Version 1.0, the subject of this review.

Portability vs performance

JWS is a substantial product; the full version occupies some 10 MB (see figure 1). It is written entirely in Java, which substantiates the language's claim to be a serious tool for programming on any scale, not just a toy for scripting pretty, multimedia applets. Currently Java Virtual Machines (JVMs) are available for Solaris (2.4 and higher) on Sparc and Intel and for Windows 95 and NT. Mac is promised within three months. IBM is work-

ing on JVMs for Windows 3.1, OS/2, OS400, AIX and MVS. The major operating system providers have all promised to include JVM support. Sun is currently assessing which JVM's implementers will be licensed to use JWS. One possible outcome is that they may all get JWS. This would consolidate Sun's claim that its portability makes JWS superior to competitive products.

While Java's portability is undeniable, its down-side is performance. The platform for this review was a Model 70 SparcStation 10 running Solaris 2.5. This is no meagre hardware: probably equivalent to a high-clocked Pentium, running Windows NT. Nonetheless an awful lot of time was spent waiting for things to happen. For example, the debugger took over twenty seconds to open with a small applet and a further twenty-five to display the debugging browser window (in which the applet runs). Sources within Sun, who have run the not-yet-published Release 6 of JWS, claim that it is between two and four times faster. Sun has promised to include Just in Time compilation (JIT) within JWS, although it is unlikely to appear until Version 1.1.

How will JIT help? When writing software there is a trade-off between performance and portability. At one end of the scale are those PC programs that short-circuit DOS and Windows, and run like the clappers, but only on specific platforms. Java's claim, to 'write once, run anywhere' places it way towards the other end of the scale: highly portable, but performance suffers. JIT drags it back towards the centre. It dynamically compiles Java class bytecode files, to something close to native machine code, before running them. (In the conventional Java arrangement, the bytecodes are interpreted at run-time). This gives finished applications something approaching the performance of C binaries. JIT technology will be selectively included within JWS itself.

It is instructive to see how other vendors are tackling the portability-vs-performance conundrum. Symantec's Café was written in C/C++ and is therefore targeted at only a single category of platform: Intel 32-bit PCs. However it is undoubtedly faster than Release 5 of JWS 1.0 and it *is* available! Symantec has announced a Visual Café for later this year which should be more suited to RAD developers' needs.

Borland's Latté has the GUI-builder written in Java. The development environment is written in Delphi and the debugger is a hybrid of Java and C++. The first release, due to ship at the end of this year, will thus only be for 32-bit Intel platforms. In the meantime Borland offers an add-on to its C++ 5.0 product but no full development environment.

Developing in a browser

JWS's browser is based on HotJava. Sun has moved away from trying to sell HotJava as an end product to release it as an embedded technology. The browser which provides JWS's underlying windowing and hyperlink system is a good example. The icon toolbar is displaced by nine distinctive icons, at the bottom of the screen, representing the principal Workshop components. Fly-by text, appropriate to each icon, appears at the browser base and only a single mouse-click is needed to trigger an icon. The default window size is considerably larger than Netscape's. Hence a minimum screen of 800x600 pixels is needed whatever the platform. Unlike Netscape, the JWS browser does not accept standard X11 options and pre-release documentation gives no hint on how to set X defaults, you are stuck with the out-of-the-box configuration.

Many components use the tabbed cards format for dialogs. Despite adopting this PC idiom, Sun has retained its traditional restraint: black text on a grey background with data entry fields in white. Source-code browser windows are smaller: on a standard workstation screen (1160x980), two can be stacked vertically, with all code visible in both. Horizontally there is only a small overlap so it is entirely practicable to have four separate source files simultaneously visible. However this should rarely be necessary because components operate in concert and automate cross-referencing. For example the compiler generates error messages which have a hyperlink to the defective line in the source editor (see figure 2). Similarly you can click on a method in the class browser and the source editor displays it. The debugger permits particularly fluent movement between source-code editor, run-time browser, thread stack, breakpoint definition pane and so on. Cut-and-paste operates between all components using standard Microsoft keyboard shortcuts, even on a Sun workstation. However, the Edit menu fails to display keystroke alternatives to menu items.

There are two benefits from developing software within a browser, rather than within a static window. First, applets intended for use on the Web can be viewed, from the outset, in the environment in which they will be run, which should lead to (urgently needed) improved standards of Web page design. Second, developers needing support can move, through a continuum, from JWS's own on-line help texts (resident on the development workstation) to help from support sources and Newsgroups out on the Net. One of the innate features of Java is that, while the pure language is

small and simple, the standard class libraries are vast and external help is essential. The main index has some 4500 entries and to these must be added internally developed classes plus any from third parties.

Workshop tools

A project may be a local or remote applet, a standalone application, an image or a Java package. Each gets its personal environment for compilation (options, CLASSPATH, package, etc), debugging, run-time (complete with distinct icon, specified space in the browser, alternate text for non-Java-enabled browsers and the rest) and publication. Projects can be segregated into distinct portfolios. A Portfolio Manager is used to create, delete and select them and to import projects (read-only) from out on the Web. Each user has a personal portfolio, plus access (where permitted) to the portfolios of other team members.

Once a Project has been selected, tools (except help) operate on it by default. However you can, at any time, open a second main browser window to handle other projects – or even other portfolios. The mouse-driven source-code editor displays the cursor line and column numbers, and icons to indicate file permissions and version control status. Any of the common version control systems can be used: PVCS for Windows 95/NT; RCS or SCCS for Solaris. Version control operates on the file currently loaded into the editor. You check it out to make it writeable and prevent another programmer altering it, then check it back in when you have finished to make it read-only and available for other users to check out.
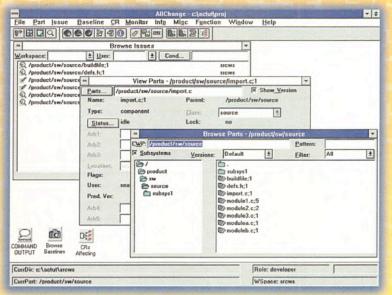
A Build Manager will rebuild everything, or just the current selection. There is a stop-button for those who remember the semi-colons too late and, for those who never remember them, controls to browse through errors. Successful compilation is greeted with a laconic 'Done'.

Class browser

The class browser will search, either a given project or a whole directory by class or by string. Matching class definition files are listed. Click on one and the browser reveals its entire ancestry, starting right from `java.lang.Object`, with the generations laid out in a family tree. Beneath appear its access, constructors and methods: both template methods, such as `init()` and `start()`, and those you have written yourself. All wear the full regalia: arching manuscript headings and coloured, ball-bearing bullet points. A click on the hypertext and there is your method, arrayed in magenta highlight, in the source browser.

Search for a string and every occurrence is listed, with module name, line number

Bring it all together

with **AllChange** for
Configuration Management

What is **AllChange** . . .

AllChange is a complete change control and configuration management system that may be tailored to site and project requirements. It is a database based system enabling it to know about the relationships between items and to control off-line items such as hardware or paper documents, as well as on-line files. Its unique action triggers enable **AllChange** to actively participate in enforcing procedures. It is a truly unique configurable system enabling it to match your specific requirements.

- Configuration item identification
- Version control
- Workspace management
- Baselines
- Bug tracking
- Life-cycle management
- User roles for access control
- Change requests
- Configuration build
- Release management
- Unix/PC client/server support
- Open interface to other tools
  (MSVC interface now available)

Our products will help you with standards (e.g. ISO9000, BS5750) and are backed by our outstanding support.

Platforms:
32-bit Windows 95 & NT
Windows 3.x
Unix

CIRCLE NO. 828

INTASOFT
Quality tools for professional software developers

INTASOFT LIMITED, Tresco House, 153 Sweetbrier Lane, Exeter, EX1 3DG, England · Tel: 01392 217670 · Fax: 01392 437877

and the code line in full. Click on the hyperlink and the source code browser immediately loads the module, scrolls to the line in question and highlights it in blue (and this is one task that runs extremely fast).

Debugger

The debugger shows the most dramatic advance over its command-line predecessor in the JDK. Mouse-click by mouse-click, you can suspend or restart the execution of a thread, step through your code line by line and, if a line represents a method, step into or over it. There are three components:

- The editor window to view source code. An improvement on many debugger code tracers, because you can edit on the spot (provided you have checked out from version control). Coloured bars highlight the different items: red for break points, green for the method executing when the

program stopped, yellow for the line containing the current error and so on. There are icons (disabled when the window is operating just as an editor) for stepping through code, toggling breakpoints and moving up and down the stack.

- The debugging browser runs your applet. You can pause at any time to surf the net, the Java API or the departmental coke machine. When the debugger is launched, the main browser reveals tabbed dialogs which enable you to examine individual threads, evaluate expressions, set or alter breakpoints, catch or ignore exceptions, stop in or show the source for specified classes or examine the project's and the debugger's input or output messages.

- The thread (and stack) browser. It displays system as well as applet threads in a format reminiscent of File Manager. Coloured glyphs distinguish between thread groups, threads (running or suspended) and frames in the thread's call stack (see figure 3). You can expand a group to display constituent threads or contract it to keep the display simple. Likewise clicking on a thread glyph displays all methods which have been called but not returned, with the frame in which execution stopped at the top. A further click, on the frame glyph, displays variables local to the stack frame, dimmed if they have not yet been assigned a value.

Debugging an object-oriented module is different from tracing flow through procedural code. Multiple threads compound the confusion. This attractive graphical display, easy to manipulate and relate to source code and the running applet, will delight veterans and programmers new to OO.

Figure 1 – The HotJava technology is embedded in JWS.

GUI builder

The most unusual component of JWS is the GUI builder. It provides the conventional widget palette, but uses a grid system for the canvas. This enables it to reflect the stringent rules, that enable Java bytecode to run under any windowing system for which a virtual machine has been written. Fields in the grid can be selected and resized by sliders in the margin. The palette bears sixteen pre-built widgets: single-line and scrolling text fields, text label, buttons with text or an image on the face, check-box, etc. A few are more of a surprise: a bordered layout panel that can contain other panels which can contain other panels ad infinitum, a database column list and a labelled bar. You can add those of your own manufacture, by importing the appropriate URL. Widgets are easily placed (click to select one from the palette, then click the field where it is to go), resized and configured: a double click opens a dialog in which attributes can be edited. Radio buttons toggle between layout and preview modes. In the latter, the grid miraculously disappears, leaving the widgets stark and functional: the buttons press, the scroll-bars slide and so on.

Once the GUI has been built, a single menu item generates java code, in the shape of three .java files. The first contains the main() method required to run an application from the command line, a 'helper' class to run an applet, and the init(), start(), stop() and destroy() methods. The second contains the GUI definition. Neither of these should be edited. Into the third is a template in which you can add callbacks to appropriate methods for the objects present in the GUI definition file. Keeping the editable file separate is one of the benefits of objectness. What joy not to lose manual edits when a forgotten button is added belatedly!

Despite the documentation, most operations of the GUI builder can be puzzled out. Writing callbacks, however, proved problematic. The GUI builder introduces Group and
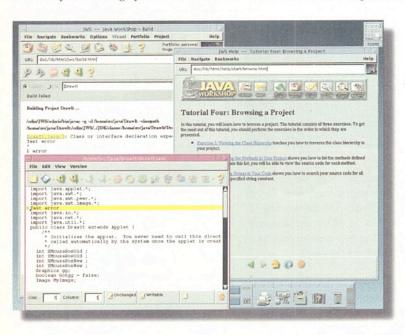
Figure 2 – The JWS environment in action.

Shadow classes, which use of is quite different from the use of widgets straight from the Abstract Window Toolkit (AWT) and demands explanation. A simple solution: don't use them until the documentation comes. The JDK enables perfectly respectable GUIs to be created from text files; it just takes longer. Sun's original impulse, to defer publication of the Visual
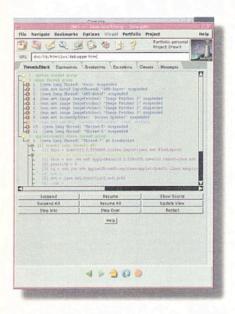
Figure 3 – The debugger showing the project Drawit.

Workshop, was sound. However, no harm has been done by appeasing our curiosity and, who knows, some may even have managed to write callbacks without instructions!

JWS's help system covers the tools but not the Java language. A set of tutorials, which take a couple of hours to complete, will leave the programmer confident of the basic operations of the JWS. Neither has the degree of detail necessary for problem-solving and it is to be hoped that the full release will be more extensively documented. The JDK includes an excellent, hypertext version of the full Java API and many coders will keep this permanently on-screen. However if you browse this using Netscape, then try doing so using this (pre-release) version of the JWS browser, you'll cringe at the loss in performance. Hopefully it is caused by debug code and is not a consequence of HotJava's intrinsic lack of poke!

The Java proof

The enthusiasm with which Java has been embraced, as a means of providing web animation and exploiting multi-media, has obscured its wider potential. However some examples of its potency for serious programming are appearing.

The availability of a steady stream of third-party tools, class libraries and virtual machines for other platforms could result in

Java becoming the commercial programming tool of choice for many software developers. Clearly the availability of a top-quality, cross-platform development toolset is an enormous additional asset. Whether it can dislodge the huge critical mass of C/C++ remains to be seen. Perhaps the deciding factor will be the serious problems, mostly unreported, that corporate users have experienced in implementing client/server solutions.

No need to crystal gaze on this. Because of the strange nature of the Web, you can download both a stable JDK and the pre-release JWS and (taking care to keep them well separate) use both to work up your skills. Even if Java goes off the boil the fun you will enjoy and the insight you will gain into programming for network computing will be your own reward. JWS has the feel that it was written for programmers by programmers who cared desperately about personal productivity, but ultimately, it *must* go faster than it does under pre-release 5.　■

Jon Vogler is a Chartered Engineer. He currently freelances for the computer press, specialising in UNIX and Open Systems, and works as a computer expert witness. If you have a dispute about intellectual property or copyright in your code, email jon@vogler.demon.co.uk.

BOOKS

Inside Linux reviewed by **Paul Dunne**

The Linux kernel is not a subject that has received much attention from writers on Linux. Though there are quite a few books devoted to this operating system, they deal with the user side of things: installation, configuration, and what to do with the damned thing once it's up and running. Bentson is to my knowledge the first in print to examine the Linux kernel.

The book is quite short, and less than half is devoted to the kernel proper, the remainder covers Linux's history, command syntax for the various shells, etc. From one aspect, this is a disappointment; on the other hand, the OS is commonly understood as being more than just the kernel, so it is a justifiable approach. It does have the advantage of showing how the kernel integrates with the higher levels of Linux.

He starts by telling us what Linux is; then places it in historical context; descends through an examination of the user interface to the details of the kernel; then up to networking support and development tools; and ends with well thought-out bibliography, a useful list of the Linux system calls, and a credit listing of those 'who have made direct contributions to the kernel and run-time library'.

Bentson is objective, and does not shrink from remarking on some of Linux's failings: for example, as regards to security, where he makes the apposite comment (on the risks associated with `sendmail` running as root): 'If the mail spool files could be tagged with an entry which says "`sendmail` can append", a large class of problems would disappear.'

One double-edged feature is the heavy use of URLs as references. While it may be that, such is the reliance of Linux development on the Net, there is no other source for the information, such references are commonly short-lived and unstable. By the time the book becomes redistributable under the GNU Public Licence, in 2010, I wonder how many of them will still be live? In fairness, it must be said that until the problem of inconstant URLs is solved, Bentson had no way, other than including the bulk of all those web pages, ftp sites, etc in his book, of providing the reader with access to their contents.

In conclusion, I can say that this is a worthwhile book for several audiences. For anyone with good Unix experience, and for whom therefore the standard 'introducing Linux' text is a little basic, this is a great means to understanding the similarities and differences between Linux and its kin. It also works as an introduction to OS design in general. Finally, anyone who runs Linux would benefit in their understanding of their system from reading it.

✔ **Verdict: Highly recommended**

Title:	*Inside Linux*
Author:	*Randolph Bentson*
Publisher:	SSC
ISBN:	0-916151-89-1
Price:	$22.00
Pages:	290

Managing Software Maniacs reviewed by **Philip Harris**

Management of technical teams, particularly software development teams, is renowned for being difficult. Many of today's managers were promoted from a technical position and lack basic management skills. This failing goes some way to explaining the problems even industry giants like Microsoft face when trying to produce quality software on time. Managing Software Maniacs is one of a growing number of books which set out to rectify this problem.

The book begins with setting priorities, not only of individual projects but also for the whole company. This gives Whitaker an opportunity to explain one of his key theories, namely that the priorities for a company should be customer, company, employee.

The book then moves on to more familiar ground, hiring developers both through advertising and via recruitment agencies. There are useful pointers on how and how not to reward developers, again including some eye opening anecdotes. The third chapter covers how to become an inspirational leader covering everything from what character traits a good manager needs to time management to leadership styles.

Next comes the development team itself. Three groups are covered, software engineering, quality assurance and technical publications and Whitaker describes the properties of each group and then explains how to handle the interactions between the three. Chapter five covers schedules and how to use them to motivate staff, in particular how to use milestones correctly. A brief chapter on creating an effective work environment is followed by coverage of the marketing department, specifically how to improve relationships between marketing and development. The importance of the customer is also reinforced with hints on customer visits, usability testing and customer contact.

The final chapter briefly covers general technological subjects including demonstration versions, copy protection, beta testing and even how to get customers to return registration cards. This chapter is a little brief, covering a wide range of important areas with barely enough detail to be useful.

The author, Ken Whitaker, has a strong management background with twenty years of experience, including stints at Data General and Software Publishing Corporation. His experience shows in his light writing style which mixes punchy *Rules for the Unruly* and *Maniac Tips* with relevant stories from his career as a manager. Its these anecdotes that make the book so valuable, making it fun to read and putting Whitaker's theories into a context that most people will be able to appreciate. While this book is quite short, it covers a lot of ground and as a result tends to touch only briefly on topics which deserve more coverage. It is an excellent basis for a manager to work on, no matter what size of company they work for.

✔ **Verdict: Recommended for managers everywhere**

Title:	*Managing Software Maniacs*
Author:	*Ken Whitaker*
Publisher:	*John Wiley & Sons (A Coriolis Group Book)*
ISBN:	0-471-00997-0
Price:	£24.95
Pages:	219

EXE DIRECTORY

TRAINING

Blueneck Computer Training
9 Shandles Road
Redland
BS6 6PG
Tel.0117 974 9134

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

Hoskyns Group Plc
Glen House
Stag Place
London
SW1E 5AG
Tel.0171 830 6830

IBM PC User Group
PO Box 360
Harrow
Middlesex
HA1 4LQ
Tel.0181 863 1191

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

Inscape Communications
35 Victoria Avenue
Rugby
Tel.01788 562052

Kibworth Computer Training
68 Springfield Crescent
Kibworth
Beauchamp
Leicester
LE8 0LH
Tel. 01533 792653

Learning Tree International
3 Swan Court
Leatherhead
Surrey
KT22 8AD
Tel.01372 364600

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

Object Designers
Western House
Cambridge Road
Stanstead
Essex
Tel.01279 816846

Oxford Computer Training
Wolsey Hall
66 Banbury Road
Oxford
OX2 6PR
Amanda Hill
Tel.01865 512675
Fax.01865 310407

Peritas Ltd
Beaumont
Old Windsor
SL4 2JP
Tel. 01753 851483

Platinum Technology
Platinum House
North Second Street
Milton Keynes
MK9 1BZ
Tel.01908 248400
Fax.01908 274888

QA Training Ltd
Cecily Hill Castle
Cirencester
Gloucestershire
GL7 2EF
Tel. 01285 655888

Rhino Software Ltd
Shaw Wood Business Park
Leger Way
Doncaster
DNT 5TB.
Tel. 01302 364861

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

Skilladvance Ltd
707 High Road
Finchley
London
N12 0BT
Tel.0181 446 6481
Fax.0181 446 9143

Tech
Surrey House
34 Eden Street
Kingston
Surrey
KT1 1ER
Tel.0181 549 3444
Fax.0181 546 6642

SECURITY PRODUCTS

Aladdin UK Ltd.
16a St. Leonards Road
Windsor
Berks
SL4 3BU
Tel.0753 622266
www.aks.com

BL Computer Security
101 Hendon Lane
Finchley
London
N3 3SH
Tel. 0181 343 0734
Fax. 0181 346 2672

Data Encryption Systems Ltd
Silver Street House
Silver Street
Taunton
Somerset
TA1 3DL
Tel.01823 352357
deskey@silver.cityscape.co.uk

GW Associates Ltd
3 High Street Studley
Warwickshire
BA0 7HN
Tel. 0527 510105
Fax. 0527 514229

Rainbow Microphar Ltd
Number 4 The Forums
Hanworth Lane
Chertsey
Surrey
KT19 9JX
Tel.01932 570066
Fax.01932 570743
sales@uk.rnbo.com

DEVELOPMENT TOOLS

AL Downloading
Voysey House
Barley Mow Passage
Chiswick
London
W4 4PT
Tel. 0181 994 5471

Bits Per Second Ltd
14 Regent Hill
Brighton
BN1 3ED
Tel. 01273 727119
Fax. 01272 727929

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

Centura Software Ltd
Lunar House
Globe Park
Marlow
SL7 1LW
Tel. 0800 834399

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

Contemporary Software Ltd
Kingswick House
Sunninghill
Berks
SL5 7BH
Tel.01727 811999
Fax.01727 848991
cssales@contemporary. co.uk

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

Geosoft Ltd
3M Springfield House
Hyde Terrace
Leeds
LS2 9LN
Tel. 0113-234 4000

Great Western Instruments
Redwood House
Bristol Road
Keynsham
Bristol
BS18 2BB
Tel. 0117 987 0444
Fax. 0117 987 2977

Highlander Software
106 - 108 Powis Street
London
SE18 6LU
Tel.0181 317 4321
sales@highlander.co.uk

Hypersoft Europe Ltd
PO Box 901
Hassocks
West Sussex
BN6 9ZS
Tel. 01273 834555
Fax. 01273 834596

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

Intasoft
Tresco House
153 Sweetbrier Lane
Exeter
EX1 3DG
Tel.01392 217670

Intersolv
Abbey View
Everard Close
St Albans

Herts
AL1 2PS
Tel. 01727 812812

LBMS
1 Evelyn House
62 Oxford Street
London
Tel. 0645 526785

Logic Programming Associates
Studio 4
Royal Victoria Patriotic Bldg
Trinity Road
London
SW18 3SX
Tel. 0181 871 2016
Fax. 0181 874 0449

Microcosm Ltd
1 Eastfield Road
Westbury-on-Trym
Bristol
BS9 4AD
Tel. 0117 983 0084
Fax. 0117 942 7295

Microprocessor Engineering Ltd
133 Hill Lane
Shirley
Southampton
Hants
SO1 5AF
Tel. 01703 631441
Fax. 01703 339691

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

Oxford Computer Consultants
Barclays Venture Centre
Uni of Warwick Sci Park
Sir William Lyons Road
Coventry
CV4 7EZ
Tel. 01203 690934

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

Power Software
8 Albert St
Aberdeen
AB1 1XQ
Tel. 01224 622201

QBS
10 Barley Mow Passage
Chiswick
London
W4 4PH
Tel.0171 956 8000
www.qbss.com

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

Quite Software
105 Ridley Road
Forestgate
London
E7 0LX
Tel. 0181 522 1726
Fax. 0181 2571044

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

Sophos PLC
21 The Quadrant
Abingdon
OXON
OX14 3YS
Tel. 01235-544012
Fax. 01235 544012

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

PROGRAMMING TOOLS

Atria
Wyvlos Court
Swallowfield
Reading
BERKS
RG7 1PY
Tel.0990-561516
Fax 0990 143096

Grey Matter
Prigg Meadow
Ashburton
Devon
TQ13 7DF
Tel.01364 53499
Fax.01364 53071

Oxford Computer Consultants
Barclays Venture Centre
Uni of Warwick Sci Park
Sir William Lyons Road
Coventry
CV4 7EZ
Tel.01203 690934

PUBLISHING

Addison Wesley Longman
Edinburgh Gate
Harlow
CM20 2JE
Tel. 01279 623623

AP Professional
24/28 Oval Road
London
NW1 7DX
Tel. 0171 267 4466
Fax. 0171 485 4752

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

John Wiley
Baffins Lane
Chichester
Tel. 01243 770 270

McGraw Hill
Shoppenhangers Road
Maidenhead
SL6 2QL
Tel. 01628 23432
Fax. 01628 770224

For details of how to put your entry in the EXE Directory, call Kieran Watkins on 0171 287 5000

The EXE Directory is a new service designed to enable you to reach over 10,000 programmers and developers in the most cost-effective manner possible. It has been created to help your chances of getting business from the buyers of your products and services in an environment where you are not outshone by competitive advertising.

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

B&T On-line

Cromwell Media have kindly offered all EXE readers one year's free membership to B&T On-line, and guarantee a minimum of one hour free usage per month (a saving of £695).

Supported by Informix, Silicon Graphics and Uunet Pipex, B&T On-line includes Daily IT news, a comprehensive daily news service; a full press release service and guides to every sector of the IT market. The most advanced feature, however, is Jeeves, a Digital Agent which personalises the service to each user's individual requirements.

Readers should simply access the URL: http://www.cromwellmedia.co.uk/newuser and use the invatation code EXE. Within 24 hours you will receive a user ID and password by post and email.

For further information, please contact Toby Ducker on 0181 563 1000 or email tducker@cromwellmedia.co.uk

Essential Java

Essential Java is a guide to the Java-related technologies developed by Sun Microsystems and Netscape Communications Corporation. Java and JavaScript are among the newest and most exciting technologies to hit the Web, bringing executable content to an otherwise 'static' universe of Web pages. At long last, your HTML documents can become fully fledged applications, and with Jason Manager's Essential Java you can learn how to incorporate interactively, sound animation - and more - into your Web pages.

Essential Java is written for HTML authors, Web-developers and novice programmers who want to get to grips with Java and JavaScript. The book begins with a comprehensive introduction to Netscape's JAvaScript, an ideal first language and starting point for HTML authors wishing to make the transition to real Web-programming. Moving on to the larger and more powerful Java language, you will discover step by step how to develop Java 'applets' and applications. A companion CD includes the Development Kit (JDK) for Windows 95, Windows NT and Solaris. Reccomended retail price: £26.95. Price to EXE readers: £21.00.

Being a generous bunch here at EXE, we have five copies of Essential Java to give away. All you have to do is send your name and address on a postcard to the FREEPOST address. Please mark your postcard JAVA.

Information supplied by the publishers

Competition Winners
June Crossword: Derek Charmer, Essex   July Crossword: Chris Towers, Essex

---

ADVERTISERS INDEX

ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE	ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
Aladdin I	Smart Card Systems	829	71	Intersolv	Development Tools	788	17
Aladdin II	Security Systems	835	IBC	Mentacaptus	Computer Accessories	805	38
Atria	Programming Tools	783	4	Microcosm	Copy Control	796	29
Bits per Second	Graphics Tools	822	60	MKS I	Toolkit	818	54
Bloomsbury Software	Development Tools	813	48	MKS II	Source Integrity	825	65
Citadel	Comms Library	803	37	Oxford Computer Consultants	Programming Tools	782	4
Contempory Software I	Development Tools	802	34	Popkin	System Architect	809	42
Contempory Software II	Development Tools	816	51	Power Software	Development Tools	795	28
Contempory Software III	Development Tools	836	OBC	QA	Training	806	39
Contura Software	Development Tools	798	29	QBS I	Development Tools	785	11
DES	Security Systems	819	57	QBS II	Development Tools	792	25
EXEPOS	GTI	808	41	Quite	Developing & Debugging	807	40
Grey Matter	Programming Tools	781	2	Rainbow	Security Products	790	20
Great Western Instruments	CASE Tools	815	50	Readmar	Development Tools	810	44
GWA	Development Tools	804	38	Rhino	Training	827	68
Highlander I	Development Tools	799	30	Ridgeway Press	Computer Products	812	48
Highlander II	Development Tools	800	31	Soffront	Bug Tracker	793	26
IBM	Software Server	780	1	Sophos	Development Tools	787	14
IBM II	Software Server	786	13	System Science I	Development Tools	784	8
Installshield Corporation I	Installshield	791	22	System Science II	Development Tools	801	32
Installshield Corporation II	Demoshield	811	47	Wextech	Doc to Help	789	18
Intasoft	Development Tools	828	70	Windows NT	Event	833	77
Interactive	Event	834	81	Zinc	GUI Library	824	62

Is this the button you were looking for?

⚛ Finish

A Codewright Programmer's Editor gets you there sooner.

Codewright™ V4.0
Professional
Editor for Windows

What it takes.℠

What it takes to get it done

It seems, at times, like the job will never be done. That's why it's so satisfying when you do get to press that "Finish" button. When you make Codewright your programmer's editor, you'll get more of that satisfaction. And quicker, too. Supplement your current tools and methods with an editor that increases your productivity like no other can.

What it takes to be #1

For a programmer's editor, it takes: powerful features, world-class support, and people like you -- ready to take a chance on something better. Call now, and in just a few days, you'll be moving faster toward the finish button.

What it takes to meet a deadline

It takes: *Selective Display* that works like an outline to hide unwanted details until needed; *Help Manager* that lets you get context sensitive help from any combination of helpfiles; *API Assistant* to ensure that your function parameters are correct before they get to a debugger.

> Now shipping version 4.0!
> Includes VCSync program
> for Visual C++
> and support for .MVB files.
> 32 bit and 16 bit in one box!

What it takes to be productive

It takes: *Construct Templates* that adapt to the way you work and take the drudgery out of repetitive tasks; *File Differencing and Change Merging* to protect the fruits of your labors; *Multi-file Search and Replace* to cut those global changes down to size.

premia®
Premium Quality Software

1075 NW Murray Blvd., Suite 268
Portland, Oregon 97229 USA
Phone: +1.503.641.6000
Fax: +1.503.641.6001
Web Page: http://www.premia.com
Email: sales@premia.com

What it takes to get home

It takes: *Button Links* that let you tie non-text elements into your source code, and help you organize a To Do list; *DLL Extensibility* that lets you write extensions in the language of your choosing.

Premia is a registered trademark of Premia Corporation. Windows is a trademark of Microsoft Corporation.

In the U.K. contact:

Grey Matter, Devon
Telephone +(364) 65 41 00

System Science, London
Telephone +(718) 33 10 22

✎ CIRCLE NO. 834

Discounted books – exclusively to you

Addison-Wesley is the supplier of this month's selection of bargain books

▲ **ADDISON-WESLEY**

The Java series is the official source of complete, expert, and definitive information on Java, from the creators of the Java technology at Sun Microsystems. The books provide the inside information you need to understand Java thoroughly and are an indispensible resource for anyone programming in Java.

The Java Programming Language

By Ken Arnold and James Gosling
Price: £28.95
Price To You: £23.16

SAVE 20%

Co-authored by the creator of the technology, this book will serve as the definitive technical guide to Java. Written from a real-world programmer perspective with 'insider' details from two Java developers, the book explains the design motivation of the language as well as the trade-offs involved using specific features.

Information supplied by the publisher

The Java Application Programming Interface

Vol 1: Core Packages
Vol 2: Window Toolkit and Applets
By James Gosling and the Java Team
Price: £36.95
Price To You: £29.56

SAVE 20%

The Java Application Programming Interface describes the libraries that every programmer can use to write Java applications and applets. These two volumes were written by the development team that created the Java API. They are authoritative, accurate, comprehensive, and completely up-to-date. Volume 1 describes the languages that are the foundation of the Java language. Volume 2 describes the libraries that you can use to create sophisticated and visually appealing Java applications and applets.

Selection	RRP	Your Price
The Java Programming Language	£28.95	£23.16
The Java Application Programming Interface Vol 1	£36.95	£29.56
The Java Application Programming Interface Vol 2	£36.95	£29.56

Descriptions of all books below can be found in *EXE Magazine*, Sep 95 to August 96

Selection	RRP	Your Price	Month
Java Essentials for C and C++ Programmers	£24.95	£19.95	Aug 96
Civilising Cyberspace	£22.95	£18.35	Aug 96
Software Requirements and Specifications	£19.95	£15.95	Aug 96
Java Unleashed	£46.95	£37.57	July 96
Visual Basic 4 Unleashed	£41.50	£33.20	July 96
Database Developer's Guide with Delphi 2	£49.95	£39.96	July 96
TeachYourself Java in 21 Days	£37.94	£29.99	June 96
Delphi 2 Unleashed	£54.95	£43.96	June 96
Programming Windows 95 Unleashed	£46.95	£37.56	June 96
Schildt's Advanced Windows 95 Programming	£22.95	£18.00	May 96

Selection	RRP	Your Price	Month
Essential Java	£26.95	£21.00	May 96
Cross-Platform Programming for Windows	£26.95	£21.00	May 96
Foundations of Visual C++ Progr for Windows 95	£38.99	£29.95	April 96
Visual FoxPro Developer's Guide	£41.67	£31.25	April 96
Delphi Programming for Dummies	£18.99	£14.25	April 96
Unix Internetworking, Second Edition	£52.00	£39.00	Mar 96
Survival in the Software Jungle	£39.00	£29.00	Mar 96
Distributed and Multi-Database Systems	£49.00	£37.00	Mar 96
The Fax Modem Sourcebook	£24.95	£18.75	Feb 96
Window Multi-DBMS Programming	£45.00	£33.75	Feb 96
Object Orientation - Second Edition	£24.95	£18.75	Feb 96
Internet World 60 Minute Guide to VRML	£18.99	£14.25	Jan 96
World Wide Web SECRETS	£38.99	£29.25	Jan 96
Internet World 60 Minute Guide to Java	£18.99	£14.25	Jan 96
Windows 95 Secrets	£38.99	£29.25	Oct 95
Power PC Programming for Intel Programmer	£48.99	£36.75	Oct 95
201 Principles of Software Development	£20.95	£16.80	Sep 95

Title	QTY	PRICE
_____	___	_____
_____	___	_____
_____	___	_____
_____	___	_____
_____	___	_____
_____	___	_____
Shipping at £3.50 per order		£3.50
TOTAL		_____

Name:_____
Address:_____
_____
_____ Post Code:_____

BOOK ORDERS

My Subscribers Club Number: _____

Payment Options

Cheques or purchase orders only.

O I enclose a cheque for _____
(payable to *EXE Magazine* and drawn on a UK bank)

O 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.

Windows NT, Windows 95, Unix C++, Java, Delphi Developers £28k-£58k

We offer absolutely outstanding benefits to absolutely outstanding software developers.

DATA CONNECTION

Data Connection has, for the past 15 years, supplied key software technologies to the likes of Microsoft, Sun, Lotus and Hewlett Packard. For example, Microsoft's Exchange product and their recently announced Net Meeting Conferencing product both have Data Connection software at their core.

We employ 100 developers, primarily at our offices in North London. To augment our very active graduate recruitment programme, we are now looking to recruit extremely able designers/programmers with 1-4 years experience in systems software development. Our current development projects range from Web audio-access, through cross-platform application sharing and distributed directory services, to high performance ATM switching support.

The technology is genuinely interesting, the environment is truly stimulating and the benefits package is exceptional. As a senior developer in Data Connection you could earn well in excess of £50k.

It only takes two minutes to phone for application details......

Data Connection Ltd
100 Church Street
Enfield, Middx EN2 6BQ

Justine McLennan
Recruitment
tel: 0181 366 1177
fax: 0181 363 1039
email:recruit@datcon.co.uk

For more information on Data Connection see http://www.datcon.co.uk

Ctrl Break

Send your rants and raves and interesting tales to:
EXE Magazine, 50 Poland Street, London W1V 4AX
email: editorial@dotexe.demon.co.uk

The ultimate tribute?

You've got to take your hat off to those guys at Microsoft. Faced with the task of knocking up the Developer Roadmap – essentially a point-and-click guide to all the Microsoft developer tools out there – they put pride in their work first and foremost and selected the best tool for the job.

And the Microsoft product they used? Er… Borland Delphi, actually.

Bizzarreville, Redmond

The Microsoft campus is an odd place. Visitors are advised to check out the Microsoft Company Store, where the company's employees queue up to *buy* all manner of Microsoft-badged goods. Don't forget to see Lake Bill, a thirty-foot duck-pond which is guarded by the fearsome Microsoft rooster (it doesn't have a name, apparently, but boy, is it vicious. Ctrl-brk still has the scars to prove it). And if you get chance, take a stroll through some of the development buildings, where t-shirted programmers will accost you with questions such as 'how many pixels to an inch?' and 'what's the airspeed of an unladen swallow?'. One word of warning: watch all the Monty Python films before you go.
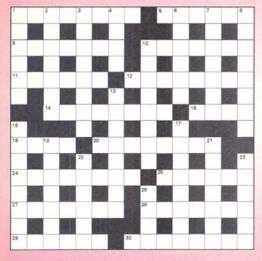
Windows 95 released in Japan!

The following factlet was sent anonymously to Ctrl-Brk – here it is verbatim. The translation of the Japanese characters intended to state 'Where do you want to go today?' are apparently 'If you do not know where you are going, we will make sure you get taken.' Perhaps a Freudian slip.

Brian and Betty

by Neil Kerber

An act of war

EXE staffers couldn't help but feel paranoid when our Production Manager's PC came down with a virus: a check with Dr Solomon's revealed it to be the 'AntiEXE' virus. This blighter is apparently designed to seek out and destroy a particular .EXE file (no-one knows which one). However, our paranoia was hardly eased by the fact that this virus is also known as D3. (This virus disguises its size and attributes.) Not that we believe for one minute that it has anything whatsoever to do with the magazine of that name, of course.

CROSSWORD

ACROSS
1. Use part of OS to maintain a tight one (8)
5. Early mainframe with one tube? (6)
9. Supposedly meaningful abbreviation (7)
10. Minor goddess with network and word of assignment (7)
11. I pout crookedly with a brave new world (6)
12. Processing in stand-alone chunks ... (8)
14. ... and giving up the profit (8)
15. A bit of cloth brought back to make jelly (4)
18. Helps the modern disease (4)
20. NOT a keyboard worker (8)
24. Task of systems specialist (8)
25. 50 vines may whine (6)
27. Go round and round IT with a tree that's crashed (7)
28. Taste of a victory sign wrapped in carbohydrate powder (7)
29. "Certainly, Sir!" (sycophant) (3-3)
30. Most users' main store (surprisingly)? (4,4)

DOWN
1. Standing of a flag (6)
2. Lack of discord (7)
3. Snide ten changing main parts of teeth (8)
4. Using 50 A, Florence carried it (4)
6. International group returns with church to carry an interesting shade (6)
7. Putting numbers on ... (7)
8. ... a class of pet from the orient, so bloody (8)
10. They look over things quickly and give useful images (8)
13. Being able to change, I've come after advertisement that's appropriate (8)
16. We count it for 30 in MB usually (8)
17. Goal of ISO or BSI, say (8)
19. Crazy leaders who can make a mint in the market (7)
21. Pasta packets (7)
22. One-legged person's nightwear? (6)
23. 150 little by little to make a priest (6)
26. EU's early rival (4)

SOLUTION TO AUGUST'S CROSSWORD

ACROSS:
1. PHOTON 4. BARYONIC 9. RIPPLE 10. SPILLAGE 11. MISUSE 12. BACONIAN 14. TIDEMARKS 16. INGOT 18. RADIO 20. NULLGATES 22. ABSOLUTE 23. LAPTOP 25. DELEGATE 26. ENTRAP 27. CRYSTALS 28. SLIDER
DOWN:
1. PARAMETER 2. OPPOSED 3. OGLES 5. APPRAISALS 6. YELLOWING 7. NEARING 8. CLEAN 13. GRANDTOTAL 15. MOONLIGHT 17. TESTPAPER 19. DISPLAY 21. TUTORED 22. ASDIC 24. ANNUL

Junior makes three

What's it like in the Gate's household, now that the stork has visited? No need to speculate: Verity Stob has once more managed to hack into the diary of Mrs William Gates.

… Dear Bill couldn't be there, owing to how he was explaining the advantages of NT 4.0's new interface to the New Jersey PowerPoint User Group. However, he had arranged for Steve Ballmer's deputy's PA's gardener to give me a lift to the hospital, but as it turned out it was five guys from the MFC development team who took me, as they wanted to video the birth. They explained that it would make a really cool test AVI file to go with Visual C++ 4.3.

The Maternity Unit of the hospital is special because it is just one of many local public works named in honor of Dear Bill's charitable activities. When we arrived at the West Redmond Built With No Money From Bill The Tight Get Birthing Facility, my contractions had stopped coming, which was a worry. But Randy Dreedle, who Dear Bill says has done some really great work on the exception handling stuff, figured I was so dilated that I had better be induced under anaesthetic, so that's what the doctors did. A beautiful girl was born at around three o'clock. I had hoped to preserve the umbilical chord in dolphin spit, as recommended in the *Californian Woman's Practical Guide to Modern Birthing*, but while I was under gas the guys took it, as they wanted to use it in their water pistol fights with the VB team.

Wednesday. Dear Bill hasn't managed to get here yet, owing to how he is explaining the Internet Strategy to the New Birmingham Women Guild's Campaign for Colorful Winterwear. But he has sent a lovely email to celebrate the birth of our darling daughter:

```
Dear
{Female Microsoft Employee's first name}
Congratulations on {the successful launch
of a new Microsoft product | the birth of
your {son | daughter}}. At this {happy |
critical} time, we hope that you will
{take some time off to celebrate your
achievement | not forget your duties to
the company}. After all it's not every-
body who can {boast having created a
Microsoft product | take nine months paid
leave and still come back to a job}. So
this message is to remind you {of how
proud we all are of your efforts | that
there are plenty of young men out there
who would give their eye teeth for your
position}.
   Sincerely

{Microsoft Line Manager's Name}
```

I guess they still haven't quite fixed the macro expansion in that new beta of Exchange.

We have been transferred to the Redmond General Not A Cent From You Know Who Post Natal Clinic, and the guys came round this afternoon for a visit, which was sweet, especially as they are having a tricky time with the new collection templates. Randy was most interested and helpful at feeding time, and warned me that I should be careful because persons with large areolas are liable to chaff; but the other guys were stupid, and said I was being cruel making the baby drink something which wasn't even chilled, never mind carbonated.
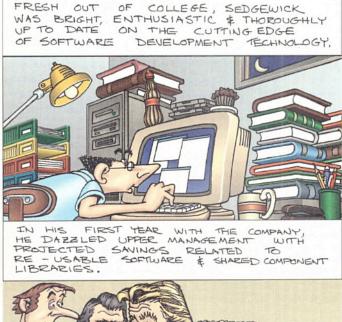
Friday. We can go home today, although I will have to go to the Washington State No Thanks To Billionaire Bill Gates Outpatient Facility next week to have the stitches out, but this is ok as Randy and the sweet guys from Visual C++ have offered go along, even though they are having a tough time with their WOW thunking right now.

Dear Bill managed to put in a video call from New Newcastle, where he is lecturing the AA on in-process OLE objects. I asked him what he thought we should call the baby. He said: We must call it William, of course. I said: But Bill, it's a girl. He seemed puzzled. So why can't we call it William? he said.

Sunday. Bill has delegated the business of choosing a name for the baby to the Visual C++ team, which has decided to call her grlSarah, in accordance with Hungarian notation. Suddenly feel overwhelmed by PND…

FRESH OUT OF COLLEGE, SEDGEWICK WAS BRIGHT, ENTHUSIASTIC & THOROUGHLY UP TO DATE ON THE CUTTING EDGE OF SOFTWARE DEVELOPMENT TECHNOLOGY.

IN HIS FIRST YEAR WITH THE COMPANY, HE DAZZLED UPPER MANAGEMENT WITH PROJECTED SAVINGS RELATED TO RE-USABLE SOFTWARE & SHARED COMPONENT LIBRARIES.

OBJECT LESSONS © 1996 BILL PROCTOR

BUT SEDGEWICK WAS OBLIVIOUS TO AN UNDERCURRENT OF RESENTMENT AMONG HIS PEERS WHO WERE RELUCTANT TO EMBRACE CONCEPTS SUCH AS SHARED LIBRARIES, EGOLESS PROGRAMMING AND RECYCLED SOFTWARE.

EVENTUALLY SEDGEWICK BECAME AN INDEPENDENT RECYCLING ENGINEER.

THE LAST TIME A REPORTER MADE THIS KIND OF TRANSFORMATION, HE WORE TIGHTS AND A BIG "S" ON HIS CHEST.

Get ready for a giant leap in reporting capabilities. Because now there's new Crystal Reports Professional 5.0 — giving you the power to develop and report like never before.

This major release offers an incredible new level of reporting flexibility. With Crystal Reports 5.0, you can create anything from subreports to form-style reports, conditional reports, multiple detail section reports, Web-ready reports, multiple summary cross-tab reports, BackOffice® reports and more.

You also have the added flexibility to publish presentation quality reports directly to the Internet and Intranets using new HTML output options. Plus, you get additional power and control with over 25 new functions, over 15 new ActiveX® (OLE/OCX) properties and many other super new features.

As a Visual Basic™, Visual C++™ or Delphi developer, or even as a user of a competitive product you qualify for a special upgrade price of **£149**, a saving of more than 50% off the normal list price.

To order your **"Super Reporter"** Crystal Reports Professional upgrade call the Contemporary Software upgrade hot-line on **07000 422 224.**

Upgrade NOW call: 07000 422 224

✎ **CIRCLE NO. 836**

CONTEMPORARY
software

⑊ **Seagate** Software

Information Management Group
Formerly CRYSTAL *A Seagate Software Company*

Tel: 07000 422 224 Fax: 07000 422 225
E-mail: cssales@contemporary.co.uk
Sales office, Contemporary Software Ltd, Kingswick House, Sunninghill, Berkshire SL5 7BH