THE SOFTWARE DEVELOPERS' MAGAZINE

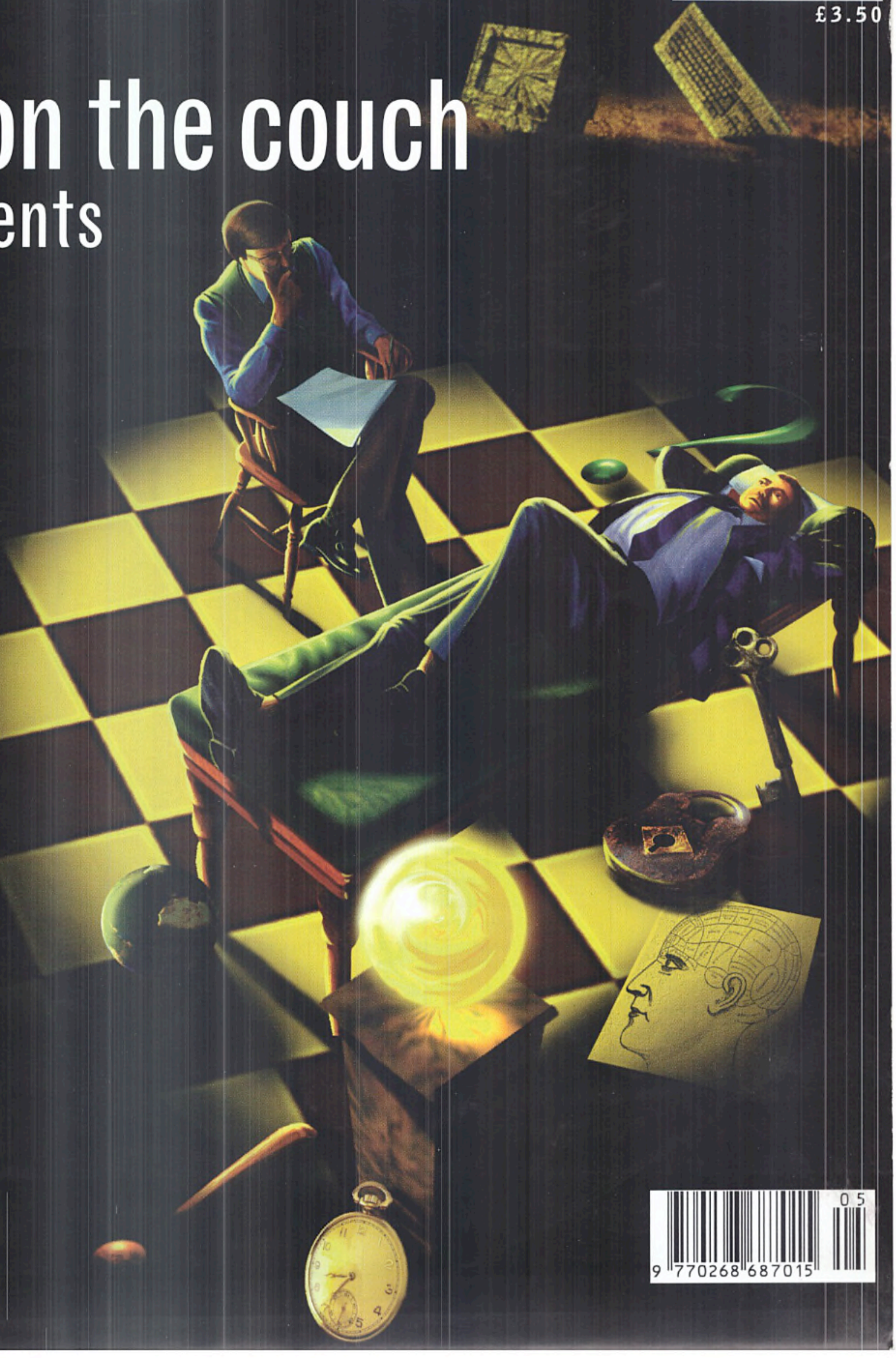
EXE

MAY 1999

£3.50

Users on the couch

Requirements gathering



Unix power!

Palm goes Forth

Delphi in the dock

Harnessing your classes

Open® warfare at Slashdot

Unix in unique perspective 41

A new Dark Age looms large 14

Lie on the couch and tell me about your requirements 18

A Forth in the Palm is worth two on the PC 27

Editor: David Mery (*editorial@exe.co.uk*, 0171 970 6550) **Features Editor:** Neil Hewitt (*neilh@exe.co.uk*, 0171 970 6551) **Technical Sub-Editor:** Alun Williams (*alunw@exe.co.uk*, 0171 970 6549) **Production Manager:** Tarina Starkey (*tarinas@exe.co.uk*, 0171 970 6542) **Group Advertisement Manager:** Mark Parker (*markp@exe.co.uk*, 0171 970 6547) **Classified Sales Executive:** Sarah Horsley (*sarahh@exe.co.uk*, 0171 970 6545) **Administrator:** Amy Williams (*amyw@exe.co.uk*, 0171 970 6541) **Circulation Manager:** Ian Paxton **Circulation Executive:** Tim Chaplin **Group Art Director:** Colin McHenry **Publisher:** James Bennett (*jamesb@exe.co.uk*, 0171 970 6540) **Publishing Director:** Roger Beckett

EXE: The Software Developers' Magazine is independent and not affiliated to any vendor of hardware, software, or services. It is published by: **Centaur Communications Ltd, St Giles House, 50 Poland Street, London W1V 4AX. EXE Telephone:** 0171 970 6541 **Fax:** 0171 970 6741 **Advertising email** *markp@exe.co.uk* **Subscriptions** Tel: 0171 292 3706 Fax: 0171 970 4099 email: *execirc@centaur.co.uk*. EXE is available by subscription at £35 per annum (12 issues) in the UK: see subs card between pages 10 & 11. The magazine is published around the 1st of the month. To subscribe or if you have a subscription query, please call 0171 292 3706 or write to Anna Martin, EXE (address above). We can invoice your company if an official company order is provided. Back issues are available at £3.50 each. **Editorial**. Address all editorial enquiries and comments to The Editor, EXE (address above) or email to *editorial@exe.co.uk* (press releases by fax or mail only). We welcome letters, opinions, suggestions, and articles from readers. These may be edited. Information contained in EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next issue.

Repro & Typesetting: Atelier Dataset Ltd **Printer:** St Ives (Roche) Ltd. **Front Cover Picture:** Mark Duffin **ISSN:0268-6872**

EXE ONLINE
http://www.exe.co.uk

e-business

All the reliability and scalability unique to DB2 is now on Windows NT and UNIX. It's also IBM EuroReady.

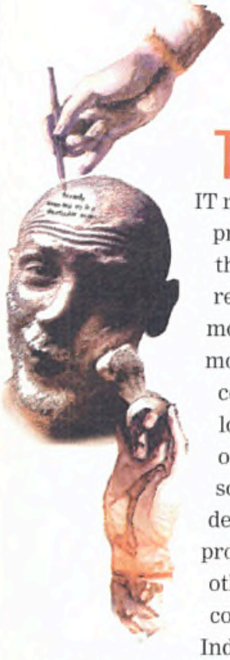
This is the database for what's in your head.

If you can think it, you can make it real today with DB2 Universal Database. Built-in Java** makes it the first database engine optimised for Web applications. New TCP-D benchmarks make it one of the price/performance leaders for any size database on both Windows NT** and UNIX.** Powerful middleware tools make it easy to exploit existing apps and data sources anywhere and extend them to users everywhere, instantly. The demo alone will get you thinking. Visit www.software.ibm.com/udb98 for eye-opening benchmarks and a free evaluation CD.

IBM

Solutions for a small planet

An Indian takeaway



The increasing IT recruitment problem in the UK is resulting in more and more companies looking to outsource software development projects to other countries. India has already established a lead in the offshore development industry, built on a rich resource of skilled software engineers and the promise of lower costs – by as much as 70%. Java is the latest skill shortage in the UK and this is providing additional impetus to the Indian software business where Java expertise is more readily available.

Trends in software development technologies are providing an opportunity for smaller, more specialised Indian companies to enter the market. In particular, popular OO technologies such as COM and Corba lend themselves to the outsourcing of chunks of systems and application development.

The great advantage with COM or Corba is that the software is delivered as a set of objects with well-defined interfaces, which makes the process of remote development much easier to manage. While there is no need for client developers to understand the source code, they can reuse the objects again and again. These OO technologies overcome two major outsourcing issues. How do you outsource some of the development, without providing access to a proprietary engine? And how do you persuade your internal team to accept third-party code?

For example, OO technology has been used in a project to build a credit-checking database that can be accessed via the Internet. While the customer had the resources to build the front-end client system, the company lacked the components for providing TCP/IP links to the server. Indian developers built the components using Microsoft VBX and OCX packaging technologies, and they were written in C++ for optimum performance – despite the fact that the customer's development team had little experience of C++. By using COM, it was possible to deliver a set of components for which there was no need for C++ knowledge to integrate them into the VB application. The interfaces had already been defined and documented and, in fact, the customer's developers never even reviewed the source code.

Another major technology shift, which has aided the cause of this form of development, is the Web. One of the most significant concerns with offshore development is client control. The Internet has already made a huge impact on the way in which offshore projects are managed with the use of email for logged communications and FTP as a means to deliver source code and compiled software. More importantly, the growth of intranets and extranets and the use of web technology to build multi-platform software is forcing many software companies to think of the Internet, rather than Windows, as the front-end operating system. As a result, an increasing number of companies are looking offshore to re-engineer existing Windows front-ends so that the same functionality can be accessed over the Web.

The beauty of this type of software for an offshore software house is that it is ideal for remote development. Clients can have daily access to progress simply by browsing the application over the Web and this increased day-to-day visibility is starting to have a significant impact on the offshore industry.

We publish the latest version of a data-model across the Web, giving Bombay-based developers instant access to the design.

In a project for the financial software house Strategic Asset Management Solutions (SAMS), Java/Corba technology has been used to convert an existing 2-tier client/server PC product into a 3-tier system. The software takes advantage of all the benefits associated with 3-tier architecture, including a lightweight client and reusable business objects written in Java and C++. The design allowed much of the original code to be reused and ensured that the original stored procedures were unchanged. By employing an object-oriented design, all of the SQL and business processing has been moved into the business object layer.

The Web can also be used to support the management of offshore projects much more extensively than email and FTP. Building a system on top of a changing data-model creates problems when the development is carried out remotely. For instance, we have built a Java applet to publish the latest version of a data-model across the Web, thereby giving Bombay-based developers instant access to the design of the latest version of a database.

Another area that is always more complicated with remote development is defect reporting. Traditionally, email has been used to communicate defects, but too often with insufficient information. Clients can now be provided with access to a web-based defect database system, where they simply fill out the details of the defect on a form within their browser. The defect is then stored in a common repository, which is accessible by teams in India and the UK. This gives the client the ability to both communicate and receive a real-time update of the status of defects.

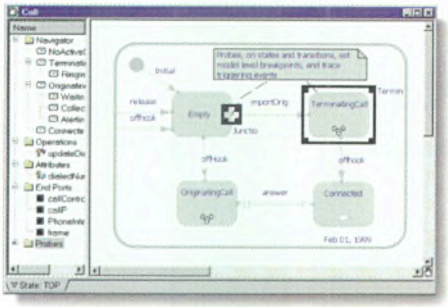
Traditionally, Indian software has been associated with mainframe bug-fixing and, more recently, Y2K work. But the move towards componentisation and the impact of the Web is resulting in a shift towards outsourcing specialist software development work. In particular, these technologies are helping to overcome the major issues that have prevented many UK companies from plunging into the tempting waters of offshore development. ■

Simon Denison-Smith
Rave Software
sds@rave-tech.co.uk

Patricia Deardorff

A real-time Rose by any other name

The reach of Rational now extends to real-time software development: the company is introducing Rational Rose RealTime. This version of its visual modelling environment is aimed at the creation of software for time-critical systems. The environment describes real-time embedded systems using UML, and it integrates with other Rational products such as ClearCase Requisite Pro, and SoDA (for automated reporting).



As mentioned last month (*An Orb embeds itself in lots of places*, News), Rose RealTime has been co-developed with ObjecTime to simplify the complexity of concurrent, distributed, or time-critical systems. It provides a visual depiction of a system's design to allow team members to understand more easily the impact of their design decisions. But as well as generating C++ application code, you can compile and test applications directly from the design models. Developers can simulate the operation of an application on both the host and target system to validate software integrity across the development cycle.

Models can be shared between Rational Rose RealTime and Rational Rose 98I for communication on projects composed of real-time development and general business application development.

Rose RealTime will be available in two editions: Modeler and Developer. The Modeler edition will provide support for C++ executables on the host system, and will be available for single users at an RRP of £5,100, which includes a year of support. The Developer edition will support C++ executables for host systems and target real-time operating systems, such as Wind River Tornado/VxWorks, Microtec VRTX, ISI pSOS, and Enea OSE. It will be available on a single-user basis for an RRP of £7,200. This will again include a year of technical support.

Rational has also announced a 'strategic product development and marketing alliance' with Wind River Systems. In addition, Rational and nine other companies addressing the embedded market have joined forces to help provide integrated solutions. The other nine are: ATA, Altia, Enea OSE Systems, Integrated Chipware, Integrated Systems, Mentor Graphics, Objective Interface Systems, ObjectStream, and ObjecTime.

w www.rational.com

w www.objectime.com

The driver seat for Windows 2000

NuMega DriverStudio is a suite of tools for Windows 2000 driver developers. Addressing a driver development lifecycle, it's intended to accelerate driver development, debugging, tuning, and testing for Windows 2000, as well as Windows 9x and NT. It enables developers to build WDM drivers and VxDs, debug drivers, determine the cause of system crashes, identify memory leaks and performance problems, and debug drivers over the Internet. The suite includes the following components: DriverWorks, VtoolsD, DriverAgent, SoftICE 4.0, BoundsChecker, DriverWorkbench, and FieldAgent.

DriverWorks has wizards for building a framework to develop device drivers. VtoolsD includes all of the documentation, wizards, libraries, and example code needed to develop VxDs for Windows 9x and Windows 3.x in either C or C++. DriverAgent provides Win32 applications with direct hardware access and control. SoftICE 4.0 is designed to reduce debugging downtime by providing system-wide control and visibility for developers to fix system-level problems. BoundsChecker Driver Edition watches all calls into the operating system kernel and detects device driver-specific errors. DriverWorkbench provides device driver development, test, and debug capabilities, including crash analysis and debugging. FieldAgent provides tools to capture information from user sites needed to solve remote problems.

The US list price for an annual subscription is $2,500, and includes products, updates, support for early releases of new operating systems, technical support, and access to the Compuware website for device driver developers.

w www.compuware.com

IBM's JVM

IBM claims its JVM for Windows out-performs competitive products by an average of 30% (SPECjvm98 and VolanoMark were the benchmarks used). The Win32 JVM enables developers to write fully-compliant Java applications for Windows 9x and NT and integrate them with a variety of other operating systems.

Performance features include a Just-In-Time Compiler combined with a Mixed Mode Interpreter. This interpreter compiles only portions of the Java program that are executed repeatedly, to provide faster load times and initialisation together with optimised executable code.

This JVM is the latest of IBM's offerings, adding to those already shipping with the AIX, OS/2, OS/400, and OS390 operating systems. It can be freely downloaded from the Web. The Win32 JVM is part of the Java Runtime Environment included in IBM's JDK.

w www.ibm.com/java/jdk/win32117/

With a commitment to versions of **dBASE** past and future, KSoft has, under licence, taken over dBASE from **InterBase**, a part of Inprise. The dBASE corporation will be a wholly-owned subsidiary of **KSoft**. A membership/subscription model will be followed with seasonal releases. The first should be dBASE 2000/Summer 1999.

www.dbase2000.com

Changes at the top at **Inprise** too, with the official resignation of Del **Yocam** as the company's Chairman and CEO. James Weil and John Floisand, respective presidents of Inprise and **borland.com**, will continue to head the two divisions. Disappointing quarterly results have since been announced.

www.inprise.com/about/press/1999/execs.html

Inner Media's **DynaZIP-AX** v4.0 lets Windows developers add zip capabilities to **ActiveX**/COM-capable programs. Instructions are provided for use with ASP-based websites for automated **zip**/unzip operations. Costing $129, it supports long filenames, disk-spanning and encryption.

www.componentsource.com

Wise for Windows, Wise Solution's installation system, will support Microsoft's new Installer technology in the **Windows 2000** operating system. The installer is part of Microsoft's Zero Administration Initiative for Windows, designed to simplify the process for **installing** and managing systems.

www.wisesolutions.com

Version 2.0.6 of **IBM**'s validating **XML** parser, written in 100% pure **Java**, can be downloaded from the Web. This version sees namespace support for DOMHash, and improved documentation and sample code.

www.software.ibm.com/xml/

VisualCafé Enterprise Suite

Gradient's NetCrusader/Corba transparently provides security services for **Corba**-based applications. The product can be integrated with ORBs such as Inprise VisiBroker and Iona Orbix. Based on the Corba **Security** Level 2 specification, the server software costs $10,000.
www.gradient.com

PHDIo is a generic **device driver** for Windows 98, NT, and 2000. It enables access to ports, the performing of basic interrupt-driven read and writes, and the use of **IOCTL**s to run commands in the driver. It includes an installation program and example application. A licence for a development workstation costs £65.
www.phdcc.com/phdio/

Unwired Planet's **PhoneStudio** is for WAP (**Wireless** Application Protocol) content and application developers. It's designed to be an environment where developers can test the UI of their WML (Wireless **Markup** Language) content on over 20 wireless phone simulations.
www.uplanet.com

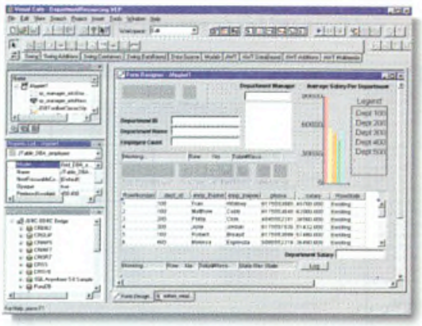
The **ObjectSwitch** 3 server software directs communications services and distributed **transactions** across telecommunications, Internet, and IT networks. It has a model compiler and code generation capability that instruments the application logic, and it delivers pre-built infrastructure services which automatically add fault tolerance, **scalability**, and online versioning.
www.objectswitch.com

Omnis **Web Client**, written using **Omnis Studio**, provides a way to deploy business systems on the Web. It is designed to bring the benefits of a **4GL** to the Web, such as rapid prototyping, easy customisation, and relatively straightforward debugging.
www.omnis-software.com

The Enterprise Suite of VisualCafé is available. Building on VisualCafé version 3, the suite represents an IDE for the creation of distributed Java applications. It inherits productivity features and database functionality from the VisualCafé family, but it introduces server-side development and distributed debugging capabilities for enterprise Java application development.

The suite introduces ORB support, such as OrbixWeb and VisiBroker, for client and server development using Corba and Remote Method Invocation (RMI) architectures. It automates development tasks such as configuring applications for interoperability across different platforms.

'Write once, debug everywhere' is Symantec's phrase, with single-view debugging for multiple heterogeneous servers. A central console can view multiple processes running simultaneously on multiple and different VMs, platforms, and OSs. The goal is that debugging time is substantially reduced with a unified view of source code, variables, threads, and call chains for local and remote processes.



Component integration is simplified with the VisualCafé Enterprise Suite library of commercial components. The adherence to the JFC/Swing data model and the use of JavaBeans Wizards is designed to simplify the integration of Java Beans into applications.

Two-way visual programming allows you to define, display, and edit JavaBeans and component interactions visually. Servlet support takes the form of Wizards and utilities to develop, debug, and deploy servlets.

The VisualCafé Open API enables you to integrate and extend the Suite with third party development tools.

For Windows 95/98 or NT, the suite has deployment options for Windows NT/9x, Compaq Tru64 Unix, IBM AIX, Sun Solaris/SPARC, HP HP/UX, and Linux. The price of a licence is £1,679 from PtS.
w *www.pts.com/static/symvce.html*

That's no mainframe terminal, that's my desktop

Bringing the joy of mainframe programming to a desktop near you is Mainframe Express from Merant. According to Merant, something like 80% of all the business applications in the world still run on mainframes, and the vast majority are written in Cobol. A growing problem, however, is the dearth of experienced mainframe developers, and the unwillingness of conventional desktop developers, weaned on GUIs and IDEs, to move to the character-based, terminal-oriented green screen environment of the mainframe.

Mainframe Express is a PC product that provides a full mainframe development environment on the desktop, including an emulator that can simulate most popular mainframes, eliminating the need to have a constant connection while developing. Merant is also quick to point out that this is the only currently-available product that understands all dialects of Cobol.

The product has a fully-integrated debugger, screen designer, version control – courtesy of PVCS technology – and Test Suite tools based on Mercury's WinRunner, which Merant has licensed. The source code editor is specialised for Cobol development, but the environment will accept and compile assembler as well.

Merant believes that the product will enable companies to attract conventional PC/workstation developers into the mainframe world, but this wisdom remains unproven. While many desktop developers know Cobol, programming the mainframe requires a fundamentally different mindset; one which might be harder to learn when divorced from the green-screen universe.

You can get more information on Mainframe Express, and related products such as NetExpress, Revolve , and PVCS at the company's website.
w *www.merant.com*

Gurus in London

London seems a popular destination for Open Source advocates. Eric Raymond came in January and in March Richard Stallman aka RMS was on his way back from Bali. In the same month Tim O'Reilly showed up to launch O'Reilly's UK branch and the book *Open sources* (ISBN -56592-582-3).

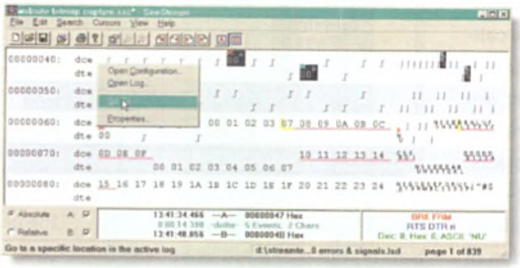
RMS is best known for Emacs, GCC, and GDB. He created the Free Software Foundation in 1984. He was re-iterating his beliefs concerning free software and not advocating a new buzzword (Open Source). For him there's more to free software than just availability of the source code. You should be able to share the code with anyone, be free to modify the code to tailor it for your needs, and be free to reuse part of the code for some of your other projects.

A point RMS stressed was that Linux is just a kernel. What most people use is GNU/Linux, a system mostly composed of GNU software that includes the Linux kernel.
w *www.fsf.org*
w *www.oreilly.com/catalog*
w *www.ukuug.org/index.shtml*

Streaming in with RS232 comms

Are you ACK-ing when you are spoken to? Have you sorted your ETXs from your elbow? Easing the debugging of high-speed serial communication links is the goal behind StreamTeam, a suite of Windows-based tools from Paladin Software. The StreamTeam components, SeeStream and LogStream, enable you to simultaneously log and monitor data patterns, protocols, and line conditions on up to four RS232 serial ports. During development, the effects of faulty software or hardware can be quickly identified. After development, LogStream and SeeStream can be used to provide a communications 'audit trail' for system maintenance.

LogStream is able to start and stop data collection based upon user-defined character sequences or packets. Data can be 'cached' for real-time display, or logged to disk for off-line use. During the data collection LogStream is able to identify and mark packet boundaries for use in later analysis. Characters can also be timestamped. In addition to recording transmitted data, it can capture transmission control signals such as DTR, DSR, RTS, CTS, etc. Each channel of recording can be independently configured, and up to 2 GB of data can be logged.



SeeStream provides users with a data search and formatted display tool. To search data logs, you can specify a variety of selection criteria. For example, hardware control signals can be tested for particular combinations of conditions or transitions. For binary data, up to 32 bits can be used in a search string, and a mask can be applied for 'don't care' bits. A number of logical operators can be applied to the result. For character-based data, up to 32 characters of printable Ascii text can be used in the search string. These can comprise character matches (with wildcards), in-set selection, or range matches. In addition, you can select timestamps, packet ID values, or marked packets. Users can then display the selected data in a variety of formats, including different number bases. Once defined, display setup parameters can be saved to file to simplify subsequent use and any production test procedures.

Windows multi-tasking allows the running of LogStream and See-Stream on the same CPU as the application, avoiding the use of a second computer for logging.

StreamTeam is Windows 95, 98, or NT-based. It costs £280 from Computer Solutions, the UK distributor for Paladin Software. LogStream and SeeStream are available individually, at £110 and £170 respectively.

w *www.computer-solutions.co.uk*

Emotion Engine software

Cygnus Solutions has produced a software simulation environment that allows content developers to begin creating and testing game titles before the availability of the next generation Sony PlayStation. Cygnus' creation of a 'virtual gaming platform' in software apparently marks the culmination of a two-year partnership with Sony and Toshiba. Dubbed the 'Emotion Engine' the new PlayStation features a 128-bit MIPS processor.

The simulation is intended to give developers a running start at delivering production quality titles simultaneously with game console production availability. The virtual hardware platform represents a complete architectural simulation environment, including a 128-bit CPU core, floating point co-processors, and DMA channels. This should allow the creation and debugging of game titles.

In addition to delivering the simulation technology, Cygnus has ported and optimised Cygnus GNUPro, the open source development environment, for the new platform. This includes an optimised C/C++ compiler, debugger, assembler, and utilities. GNUPro is also available for the LSI Logic MIPS R3000-based I/O processing system.

Sony demonstrated titles for the new games platform, at the PlayStation Meeting 1999, by using the Cygnus simulation.

Cygus Solutions is a provider of development tools and operating systems based on an open source model, from GNUPro Toolkit to eCos, the Embedded Cygnus Operating System.

w *www.cygnus.com/gnupro/*

Handheld data

The latest version of Sybase's SQL Anywhere Studio features UltraLite and MobiLink technologies. That's to say the mobile database system now includes the UltraLite deployment option and MobiLink enterprise server synchronisation for the Microsoft Windows CE operating system and 3Com Palm Computing platforms.

The UltraLite deployment option provides an application-optimised, 'ultra' small database that resides locally on the handheld device or embedded system. MobiLink provides two-way server synchronisation, enabling the UltraLite database to send and receive information from an enterprise database.

The goal is to extend business applications, such as sales force automation, to small devices.

The suggested retail price for SQL Anywhere Studio is $399 (1 user) or $999 (5 users).

w *www.sybase.com*

Unify is the latest corporation to line up behind **Linux**. Seven new products scheduled for release in the next four months will be certified for Red Hat Linux 5.2. These include Vision AppServer, the **Internet** application server, and Vision AppBuilder, the OO repository-based component framework.
www.unify.com

There is a suite of development and debugging tools for Motorola's new MPC8260 **PowerQUICC** II embedded communications processor. **Embedded** Support Tools (EST) has produced the visionICE **emulator** and visionCLICK 7.02A C/C++ debugger.
www.estc.com

Developers can integrate enterprise-scale data modelling and **database** design with COOL Data Management Toolkit. The Toolkit combines **COOL:BusinessTeam** 1.2a, a logical data modelling tool, with **COOL:DBA** 2.1, a database design and implementation tool. Databases supported by the toolkit include Oracle, Sybase, and DB2.
www.sterling.com

Cambridge Control's Simulink Report Generator provides report writing capabilities from Simulink and **Stateflow** models. Work from both products is documented in **real-time**, and reports can be created in formats including HTML and XML.
www.camcontrol.co.uk

Centura Software has announced **SQLBase** 7.5 for Novell NetWare with full *YES Tested and Approved compatibility*. The release provides **security** features to block unauthorised data connections, detect changes made to database files and pages, and prevent data from being seen as it's transmitted across the wire.
www.centurasoft.com

Data access by template

Rapid SQL 5.2 Pro is an enhanced version of **Embarcadero**'s database programming environment. Featuring an Oracle PL/SQL debugger, it enables programmers to debug **Oracle** stored procedures, functions, packages, and triggers. For Win32 clients, it supports Oracle 7.33+ and 8.03+ databases running on Unix and NT servers. It is priced at £1,109.
www.embarcadero.co.uk

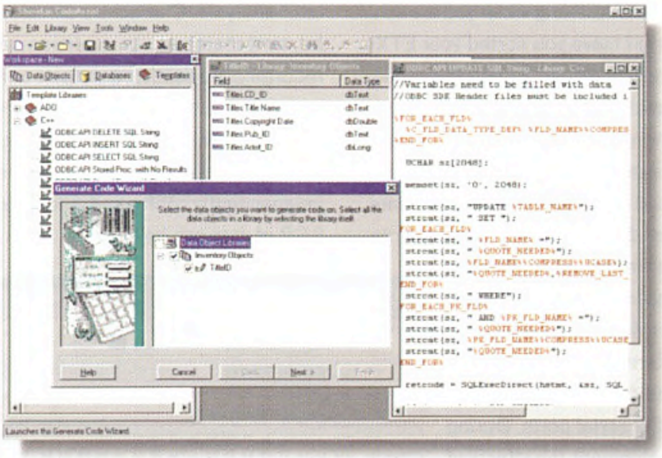
Enea OSE Systems has integrated its OSE real-time operating system with Green Hills Software's Multi **IDE** and optimising compilers. This brings together the OSE Illuminator suite of **debugging** tools and Multi's source-level debugger. Enea has also added a CPU Profiler to its Illuminator suite.
w www.enea.se/ose/

The Component Vendor's **Consortium** (CVC) is a non-profit organisation of software component publishers. Its main objective is to develop an objective testing, measuring, and branding process to identify vendors and **components** that meet a rigorous **standard**. A logo program will be established. As well as code quality, the CVC standard will establish minimum requirements for technical support and product documentation. Founding members include CompuWare, FarPoint, VideoSoft, and Sheridan.
www.components.org

Sheridan's data-access code generation software, CodeAssist, is aimed at Visual Basic developers, though it can also provide templated or customised support for other languages.

Features include more than 100 pre-built templates (covering RDO, DAO, ADO, HTML, and SQL) supporting code requirements for calling data objects from two-tier and multi-tier applications. There are also templates for browsers accessing databases and data objects.

The tool uses a point-and-click database browser to pick database elements from any combination of tables and fields. These elements can be gathered into reusable data objects, which are then 'passed through' the selected template to generate code. This release works with Microsoft Access and SQL Server, but plans are underway to release another version that will include support for Oracle and Sybase databases.



Developer's using C++, or other languages not catered for by the pre-built templates, can create their own templates. These have full functionality.

Available from the beginning of May, CodeAssist is distributed by Contemporary and costs £210.
w www.contemporary.co.uk

Suite and easy print control

VSView 6.0, a suite of ActiveX controls from VideoSoft, expands and simplifies printing functionality to replace the Printer Object in Visual Basic. In this version you can automatically format text into multiple columns for table creation; export to HTML; render pictures, RTF, or HTML; and add headers, footers, text boxes, borders, and word-wrapping.

Within VSView, enhancements to vsPrinter include a printing and previewing control that features thumbnail displays, multi-page print view capabilities, support for both landscape and portrait page orientation in the same document, and smart zooming (where the amount of zooming can be determined by keywords, or by percentages). Document creation features allow the addition of headers, footers, text boxes, and borders to VB documents with unit-aware measurements controlling their precise location at printing.

Table design features include automatically formatting text into multiple columns, and VSView tables support merged cells, cell-specific attributes, and formatting options such as colour, font, and alignment. The suite is ADO compatible, and allows the creation of bound tables that are populated from variant arrays.

The new control, vsDataLabeler, allows end-users to create custom labels while the vsDataReporter is used to generate data-aware, ad-hoc reports with no additional coding.

VSView 6.0 is compatible with Visual Basic 5.0 and 6.0 and is dependency free – no MFC libraries are required for distribution.

It costs from £169 for a single user and from £529 for five users.
w www.videosoft.co.uk

Books received this month

Publisher	Title	Author	ISBN	RRP
Wrox Press	ADO RDS programming with ASP	J.Papa, M.Brown, C.Caison, et al	1-861001-64-9	£45.99
John Wiley & Sons	Building Microsoft SQL Server 7 applications with COM	Sanjiv Purba	0-471-19233-3	£25.95
John Wiley & Sons	Building n-tier applications with COM and Visual Basic 6.0	Ash Rofail & Tony Martin	0-471-29549-3	£32.50
O'Reilly	CJKV information processing	Ken Lunde	1-56592-224-7	£43.50
John Wiley & Sons	Concurrency: state models and Java programs	Jeff Magee & Jeff Kramer	0-471-98710-7	£29.95
John Wiley & Sons	Frame relay for high-speed networks	Walter Goralski	0-471-31274-6	£32.50
John Wiley & Sons	Java 2 and JavaScript for C & C++ programmers	M.C.Daconta, A.Saganich, E.Monk	0-471-32719-0	£38.95
John Wiley & Sons	Lingo sorcery: the magic of lists, objects, and intelligent agents	Peter Small	0-471-98615-1	£29.95
O'Reilly	Perl 5 pocket reference	Johan Vromans	1-56592-495-9	£6.50
O'Reilly	The Unix CD bookshelf (6 books)	Various	1-56592-406-1	£46.50

Dark Ages

Like paper before it, ADA-MEM (very large-scale digital memory with analogue input and output) transformed the work of historians. Jules looks back to the days when all recordings were at the mercy of their formats.

After the Roman Empire had become unrunnable, it receded, gradually draining back into Rome, and taking with it the benefits of Greek civilisation, which empowered the Roman invasions some six hundred years before.

Without culture, technology, or imperialist occupation, the countries of Europe entered the shadowy time known as the Dark Ages. Not that the Europeans knew anything about the Dark Ages; the sun shone as brightly, local trade proceeded much as it had before, and international trade proceeded, if anything, more easily without the threat of coercion to which the Roman occupiers occasionally resorted. No, the Dark Ages were not dark at all to its inhabitants. It is the historians of later ages who perceive a looming, almost impenetrable shadow on the timeline. The fact is that what we know of the Dark Ages suggests a comparatively pleasant time of industry, wealth, and growth. It's just that we don't really know very much at all. We have a few artefacts, and fewer earthworks. What 'facts' we have were passed down, first through an oral tradition that preserved the memorable and the heroic at the expense of the true (remember that truth meant very little in the days before mechanical recording), and then through a written tradition that was controlled by a church that valued political correctness over any other consideration, no matter the violence done to traditions and records to maintain that correctness. The Dark Ages were not dark; they're just hidden.

What put an end to the Dark Ages was the beginning of the widespread use of paper. Laws were enacted on paper, books were written on paper, and paper itself had a value. Paper is, in fact, an amazing substance – its strength is also its weakness. It's very easy to make a permanent mark on paper, so it makes perfect sense to write or print onto it. But, because it's so easy to make the mark, paper faithfully preserves a record of everything that has happened to it. Every fingerprint, crease, water spill, or accidental impression is overlaid onto the intended message, until after a period of use, the paper wears out, even if it does not decay naturally.

Paper has been very successful for a long time, but the beginning of the digital age meant that we no longer had to tolerate the inefficiencies of paper. Digital recordings do not preserve damage and noise, and a digital recording is as clean today as the day it was made. Unfortunately, damage to a digital recording is an all-or-nothing deal – either you read everything or you read nothing at all. It's also a complicated business. Where you merely need to recognise the handwriting on a piece of paper, the extraordinary number of digital formats means that an enormous body of knowledge is inaccessible.

There's another property that digital information has; when it is erased, it leaves no trace of its existence. When digital recordings were first devised, this appeared to be a big advantage, because recycling memory or hard disk space was perceived to be easier than recycling paper, and valuable information was recycled with abandon. Even paper documents were not immune; documents were made which, though they were expected to have a considerable life span, were nevertheless designed to be incomplete, relying on digital information offered through computer networks. The digital information was too extensive to print, or not of a suitable form, but when the space used for these appendices was recycled for other, more timely purposes, the original paper documents became useless as historical documents.

The decline of paper, then, was precisely the cause of the second Dark Age. There was no cataclysm or apocalypse, there was no conspiracy or political movement, but as paper was used less, and data acquired its value through its currency, vast quantities of data were systematically destroyed, leading to a tragedy as heartbreaking as the destruction of the library of Alexandria. Today we know practically nothing of how the inhabitants of the 21st century lived. Nearly all the data from that time was erased because its owners simply couldn't be bothered to preserve it, and most of what little has survived exists only on unreadable formats. For example, musical styles are almost unknown to us since the DVD format (the standard method of storage in the early 21st century) is no longer documented (other than on DVDs, which we can't read), and even the sounds used by musical instruments are preserved only in arcane formats on magnetic disks that are far too delicate to try to decode. Artefacts are few and far between, because anything without intrinsic value was fanatically recycled in a world where resources were diminshing and refining processes were becoming more expensive.

The people who lived in this time, of course, had no idea that they were inhabitants of a Dark Age, any more than they did first time around. These people were drowning in hitherto inconceivable quantities of data, and their primary problem was one of disposal. After the stagnation of the 20th century, it was a golden age of exploration where energy and resources became, at last, too cheap to meter. It was a time of great political upheaval; the fight for civil rights in an interdependent society was reaching its climax, turning privacy and secrecy from one of mankind's most treasured ideas into an expendable archaism. The Nation States were collapsing, governments hastening the hand-over of power to the corporates by bankrupting themselves in pointless wars and unstable treaties among themselves. It was probably the most exciting time that humanity has ever seen. But this wonderful history is lost to us, and all we can see is an Earth divided by nationhood timidly entering a dark shadow, and then, against all the odds, confidently emerging some hundred years later as the Solar Federation. ■

Jules is a technology journalist, and has been so in at least two of his previous incarnations. His old-style Internet address of jules@cix.co.uk still works.

We welcome short letters on any subject relevant to software development. Please write to: The Editor, EXE Magazine, St. Giles House, 50 Poland Street, London W1V 4AX, or email editorial@exe.co.uk

Microsoft is not a monopoly

Dear Sir,

Jules writes that Microsoft is a monopoly, has behaved badly, and deserves death by a thousand cuts (*Mayhem*, EXE, April 1999).

First, Microsoft is not a monopoly. It has a very large market share of operating systems for IBM-compatible PCs. Apart from the fact that there are PCs other than IBM-compatibles, such as Apple Macintoshes, Microsoft's 'monopoly' is not like that of the Post Office's for delivery of First class mail. If I decide to go into competition with the Post Office tomorrow I will be shut down and arrested. If I decide to offer a competing operating system to Microsoft's, I will not be shut down and arrested. This is a profound difference and means that a monopoly established in the free market always has to act as if it has competition even if it hasn't. This is why the price of Windows has stayed low relative to its functionality. In other words, a monopoly, or near-monopoly, in a free market can only maintain this state by continuing to offer ever better and/or cheaper products. This applies to Microsoft.

Second, Microsoft has indeed behaved badly by the standards of the anti-trust laws. But so have many companies who have not been prosecuted under these laws. Prosecutions seem only to arise if the company has a sufficiently large market share of some arbitrarily defined market and enough of its competitors are politically connected and complain loudly enough.

The anti-trust laws betray a fundamental misunderstanding of the free market. A free market does not require that there be some arbitrary number of competitors in a given area. It just requires that should anyone wish to compete in that area they may not be stopped from doing so by the government or by private criminals. If Microsoft chooses to offer its operating system or web browser for sale on terms such that if an OEM or ISP does not accept them it does not get the product, this does not constitute a violation of the freedom of the market.

It is indeed true that, with its near-monopoly on PC operating systems, this gives Microsoft enormous economic power, but that's just tough. However, it doesn't give it unlimited power and over time the market tends to evolve responses to such power. One such response is Java. Another, possibly more promising, response is Linux and the Open Source movement in general. If it were not for the fact that Microsoft is so dominant on the desktop, both in operating systems and mainstream desktop productivity applications, Java and Linux would not have attracted nearly the level of attention they have.

Third, even if we accept Microsoft's guilt, 'execution by a thousand cuts' seems disproportionate to the alleged crimes. Ordering Microsoft to be broken up, or forcing it to auction off the source code for Windows, would be manifestly unjust measures. These are punishments that do not fit the crime. Such punishments would be appropriate if it were considered that being a monopoly per se is a crime. However, the DOJ has insisted that Microsoft is not being prosecuted for being a monopoly as such but for the manner in which it attempts to 'protect and extend' that monopoly.

The major charges against Microsoft relate to exclusionary business practices, product tying, and the like. Any punishment should be appropriate to these 'crimes' and not to the fact that Microsoft has a monopoly. Thus any punishment should consist of Microsoft's being fined, ordered to desist from such practices and, perhaps, to pay damages to the 'victims' of such exclusionary practices.

Kevin McFarlane
kevin@atech.globalnet.co.uk

You seem to be saying that the only kind of monopoly that you would accept is a statutory monopoly – one created and defended by force of law. That's ridiculous – if it were so, then the anti-trust laws wouldn't be necessary in the US, and the Monopolies and Mergers Commission in the UK would concern itself only with Acts of Parliament, not with acts of private companies.

The point about the anti-trust laws is not that they try to prevent market dominance on commercial grounds, but that they try to prevent predatory practices which destroy competition from reaching the marketplace. Nobody objects to a lack of competition because one organisation can make a better mousetrap (that, after all, is what the patent system is there to protect), but the US culture in particular deeply objects to a lack of decent mousetraps because one organisation is threatening any competition with reprisals or wiping them out by sheer financial muscle. That's what I believe MS has done, and it's what the trial will establish legally.

For what it's worth, I'm a great supporter of the anti-trust laws. They are invoked from time to time, and in every case they ruin an organisation that is generally hated – few people lamented the impending destruction of Bell (however many wished for its return afterwards).

Anti-trust is so-called not because it's anti-monopoly, but because there is an implication in law that a powerful organisation has certain duties of care which smaller organisations do not. It's not a hard concept – Premier League football clubs need all-seater stadiums, where Fourth Division clubs do not, and nobody regards that as unreasonable. Microsoft is untrustworthy, and has shown itself to abuse its powers in a systematic manner (even to the extent of lying to the court) and on that basis should be deprived of its power to cause harm, just as a drunk driver gets his licence revoked. Auctioning the source codes, or being broken up and sold off, does not destroy any property or asset value, but it does remove those assets from a culture which can't be trusted to hold onto them. Just because Microsoft originated those assets doesn't mean it should have a right to administer them, any more than you'd get to keep a gun you made yourself.

Jules May

Mark Duffin

Requirements for success

Are you planning for your software to be considered a failure? Without gathering requirements before developing, Andy Brice would say you are.

At the simplest level, the software development process is just two things: find out what the user wants, and implement it. It is a source of some amazement to me that so many projects proceed without anyone actually finding out what the users want, or even identifying who the users are. After all, you can't expect to hit a goal if you don't know where it is. 'We'd better start coding now, because we'll never get it right first time' is a self-fulfilling prophecy.

A survey by the Standish Group found the following among the leading reasons for project failure:

- Incomplete requirements (13.1%)
- Lack of user involvement (12.4%)
- Unrealistic user expectations (9.9%)
- Requirements keep changing (8.7%)

A proper statement of requirements goes a long way to alleviating these problems. It is the height of arrogance to assume you know the requirements for a system without asking the prospective users. The 'good old days' of expecting the user to adapt to the software, rather than vice versa, are long gone. Involving users in the requirements process also helps to involve them in the development process and hopefully 'buy them in' to the new system.

Studies show that some 60% of errors in delivered systems are requirements, specification, or design errors. A change to the requirements of a system can cascade through the whole system development, potentially affecting design, coding, testing, and documentation. It should not be a surprise that mistakes in requirements have been shown to be 100 times more expensive to fix in a delivered system than fixing them at the requirements stage. Writing software is an expensive business. Averaged over the whole development lifecycle (ie including design, documentation, and testing) the cost of commercial code is currently running at around $50-100 *per line*. We really don't want to be wasting time and money writing code that isn't helping the user. It makes sound economic sense to correct mistakes in requirements as soon as they are discovered. Obviously, such mistakes are very unlikely to be spotted if the requirements are not recorded in any structured way. Getting the requirements right can have other benefits; a system that doesn't require substantial re-work after delivery will generally be better structured, which means it is easier to extend and maintain and has a longer lifetime.

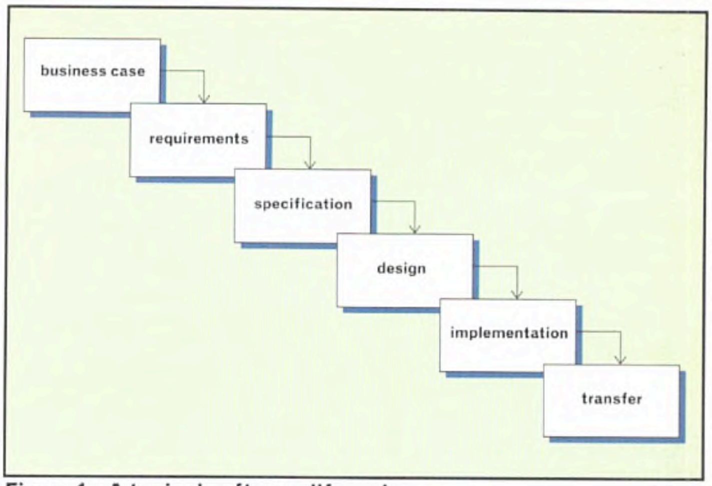


Figure 1 – A typical software lifecycle.

What is 'requirements gathering'?

Requirements gathering is essentially the process of finding out what the user wants (defining needs and opportunities) and recording it in a form that can be used as the basis for development. There are many different software lifecycles and many different terminologies in use. But, in general, requirements gathering is the phase that comes between making a business case for software development and specifying in detail what the software will do. Requirements gathering and specification are sometimes grouped together under 'analysis'. (See Figure 1.)

Requirements gathering may sound rather simple, but it rarely is. Common problems include:

- Incomplete, vague, and contradictory information.
- The lack of precision of natural language.
- The lack of a common vocabulary between users and developers.
- It is not always clear who the user is.
- Unwritten assumptions.
- Conflicts.
- The temptation to rush to solutions before the problem is defined.

As requirements form a bridge between the user and the developer, it is essential that they are written in a form that both the user and developer can understand. Generally, this means natural language (ie English for English speakers), with diagrams as appropriate. The vocabulary used should be that of the user; IT jargon should be avoided as far as possible. Mathematics, entity-relationship diagrams, decision tables, etc, should only be used if you are sure the user will understand them, and explanations and references should be added where appropriate.

Natural language is very rich, but not very precise. A 'customer' could be a person, an organisation, an account, or a relationship, and the difference might be important. Great care must therefore be taken to be precise with language. When I worked on paper mill scheduling software, some mills cut 'reels' of paper into 'rolls', while others cut 'rolls' into 'reels'. Problems can also occur between, for example, British and American English. A glossary of terms is a good way to avoid such misunderstandings.

It is important to identify the target users for the system. Will they use the system occasionally (in which case ease of use is important) or a lot (in which case shortcuts and the ability to customise may be more important)? There may be 'hidden' users, eg system administrators and other programs, whose needs have to be taken into account. Every system is part of a larger system. The boundaries of a system must be identified so there is no misunderstanding about what is inside the system and what is outside the system.

Unstated assumptions can be real system killers. You may have assumed that the system is a single-user system, the user may assume that it is a multi-user system. Trying to re-engineer a single-user system into a multi-user system after delivery could be *very* expensive. The general rule of thumb is: if in doubt, write it down. Assumptions can only be challenged if they are recorded.

The line between gathering requirements and specifying a solution can be a fuzzy one. Developers are used to thinking in the solution domain, but the temptation to rush ahead before the problem is properly defined must be avoided. The solutions that occur to you during requirements gathering should be noted separately for use later, not recorded in the requirements document.

Communicating with the user

Potential sources of information for requirements include: interviews, surveys, or informal chats with prospective users; consulting domain experts; and analysing existing systems, procedures, and documentation. Discussion with users should focus particularly on what users want to do, and problems they are currently experiencing in achieving this. It may be worth watching prospective users do their job, or even try their job yourself for a day. Hopefully, this will lead to a clear definition of needs and opportunities that can be documented.

Once this document has been reviewed and agreed by the user, it can be used as a basis for specifying an appropriate solution.

Identifying users to interview is not always straightforward. A completely new software product doesn't have any existing users. A standard software product, such as a wordprocessor, may have potentially millions of users. The best one can do in these circumstances is to find appropriate people to act as 'surrogate' users.

It has been said that we have one mouth and two ears, and we should use them in equal proportions when talking to users. Above all, requirements gathering is about *communication*. You have to find out what the user needs and record this in a way that both users and developers can understand. Never challenge the user's knowledge of their own domain.

In other words, the skills required for requirements gathering include: listening, note taking, understanding the user's domain, dealing with incomplete, conflicting and vague information, and being able to see the bigger picture.

Communication means being pro-active and asking the right sort of questions – it is not good enough to say later, 'You never told me that!' Continually clarify your understanding by asking users, and domain-experts, questions such as 'so what you mean is...'. But don't ask a user 'Would you like...?' They will usually say 'yes', on the grounds that even the most obscure functionality might be useful at some point. Users often don't understand the cost implications of additional functionality (or don't care – they may not be paying for it).

It may require some digging around to find out the real problem. A user may tell you that it currently takes them an hour to generate a report, and they want this reduced to a minute. The real requirement here may be that report generation should not prevent access to the system for more than a minute; the time taken to generate the report may not be very important. If so, solutions other than a 60x increase in speed may be possible, for example generating the report from a separate thread or allowing them to schedule it overnight.

Conflict is almost inevitable. Different users will have different requirements for a system. When a new system is delivered some people will see themselves as gaining (eg their job becoming easier) and some as losing (eg their skills no longer being necessary). And the customer, who pays for the development, may not be the same person as the end-user. There can also be a conflict between what the user wants and what the user needs. The best that can be done is to try to engineer a consensus, without getting involved in the personal politics.

Other than ears and a brain, no tools are required beyond a word-processor, and perhaps a diagramming tool. Specialist requirements gathering tools, such as DOORS (*http://www.qssinc.com/products/doors/index.html*), are available. They may be appropriate for very large and complex projects.

Writing the document

The length of a User Requirements Document can vary from a few pages to hundreds, depending on the complexity of the problem described. But the principles are the same. A good requirements document should be concise, complete, consistent, unambiguous, testable, and feasible.

Aim for clarity and simplicity in your writing. Examples make things clearer and diagrams are often helpful. Repetition is dangerous and should be avoided. It increases the size of the document unnecessarily and, even worse, leads to inconsistencies if one instance is changed and others are not.

All requirements should be testable so that it is possible to say whether they have been met by the delivered system. 'The system shall be easy to use' cannot be tested in any meaningful way and is not a good requirement. A better requirement would be: 'Users of the existing system shall be able to use the new system after two hours training'. It is always tempting to be vague to give developers maximum leeway, but this isn't professional. From my experience, such vagueness is more likely to be used by the user to their own advantage. Quantifying requirements means less chance of surprises for developers or users on delivery. It is always possible to re-negotiate the requirements later if they turn out to be problematic. (See *Bad requirements*.)

When you gather requirements you should be focusing on the needs of the user, not on implementation issues. However, there is little point in drawing up a set of requirements that you know can't be satisfied, whether due to lack of development resources or the limitations of available technology. Users often have very little idea of what is feasible. For example, they might ask you for functionality to calculate, in a few seconds, the optimal route for delivering up to 40 parcels – not realising that this is the computationally intractable 'travelling salesman' problem, which would severely tax the largest supercomputer. It is up to you to try to steer them away from requirements that you know are infeasible. Where feasibility isn't clear, some prototyping may be appropriate.

If you want the delivery of a system to be as smooth as possible you should include requirements that are not strictly part of the software, for example: administration, support, training, documentation, etc. Furthermore, each individual requirement should stand on its own, be necessary, be identifiable, and be accompanied by source references where appropriate

Identification and classification

Some form of identification scheme is necessary so that requirements can be cross-referenced against each other and traced to the specification. I tend to use a mixed words and numbers system. For example, essential performance requirement 1 might be PE/E/1 and desirable internationalisation requirement 3 might be IN/D/3. Try not to change the scheme when requirements are added or deleted as this may make existing cross-references invalid.

Bad requirements

What is wrong with these requirements?

a) Errors shall be notified to the user by a dialogue.
b) The system shall be 99% reliable.
c) The system shall give a rapid response.
d) A Laplace transform solver shall be used.
e) A two-phase commit shall be used for all transactions..
f) The system shall be written in Java.

a) Is a solution. b) Isn't clear. What does this really mean, 99% uptime, 99% of calculations correct? c) Isn't testable. d) Is a solution. e) Is jargon the customer is unlikely to understand. f) Is a solution.

Dangerous myths

'I can't afford the time to find out what the user wants.'

'I know what is best for the user, I don't need to ask them.'

'Any system is better than what they have now.'

'The requirements will work themselves out during development.'

It is helpful to classify requirements by importance, stability, and category. For importance, it is generally sufficient to classify each requirement as essential or desirable. Essential requirements *must* be fulfilled. Desirable requirements would be helpful, but are not essential for the system's success. Desirable requirements can be added or subtracted to the development plan, depending on resources. This gives useful flexibility to the developer later on. For stability, it is generally sufficient just to identify requirements that are volatile, so that developers can design for change. Finally, categorisation is a useful way to identify related requirements. Many categorisations are possible. Example categories might be: exception handling, performance (time and space), scalability, reliability, recovery, accuracy, security, auditing and monitoring, extensibility, data format, data retention periods, installation, user interface, interoperability and external interfaces, and documentation. Grouping requirements by category makes them easier to find. Categories can be further grouped into functional and non-functional, but I have never found this distinction useful.

It may be helpful to distinguish between requirements (things the system should do) and constraints (restrictions on the possible range of solutions). For example, 'the system will run on Windows 95' is a constraint. Excessive numbers of constraints restrict the freedom of the designer and should be avoided.

In an ideal world, we would like all software to be easy to use, efficient, never fail, and always give the right answer. But in the real world, we have constraints on development resources, so it makes sense to decide the relative importance of quality attributes. For example, with a cash-point machine it is probably more important that the software is correct (debits the right account) and easy to use (the users are not experts) than reliable (the occasional crash can be tolerated) and efficient (the difference between a 0.01 and 1 second response probably won't be noticed much by users). But with a nuclear reactor control system it is more important that the system is reliable (a loss of system functionality is not acceptable) and efficient (the system must respond fast enough to counter hazards) than correct (the occasional over-cautious shutdown may be acceptable) and easy to use (the operators will be trained experts).

It can be important to define how exceptions are handled. An Ariane space rocket and its payload was lost due to something as humble as an unhandled floating-point exception. If the requirements for an air traffic control system state that it will be able to handle up to 50 planes, we need to decide what will happen for 51 – shutting down the system with an error message probably isn't acceptable!

I find it helpful to record the rationale (the 'why') behind requirements. If you understand why a requirement has been added, it is easier to decide later on whether it can be safely changed or removed, particularly if more than one person is working on the document. If you can't think of a rationale, then you should ask yourself if this requirement is really necessary. As the rationale isn't strictly part of the requirements, I generally put it in italics to make it distinct. For example:

```
Requirement: Access to the system shall be password
protected.
Rationale: Terminals are physically accessible to
staff for whom access would be undesirable. The users
are familiar with the idea of passwords.
```

The requirements phase is a good time to think about acceptance tests. You should agree with the user an outline of the tests that they will carry out to show that the delivered system meets its requirements.

Requirements gathering is a process of discovery and as such it is generally messy and iterative. Answering one question will often raise several more. When you have gaps in your requirements it is much better to put 'TO BE DEFINED' in the document as a reminder, than to put nothing. (See *An example library system*.)

Generally, requirements gathering will take 5-10% of the total cost of a software development project, depending on how innovative the project is. If you take less time, you may be storing up trouble for yourself later. If you take a lot more, you are likely to run into diminishing returns.

Once you have completed your requirements document, it is essential to get the prospective users, or an appropriate representative, to review it. Users can be nervous about their competence to do this, so it is generally a good idea to give them an overview presentation before you deliver the document.

Once the user has agreed the requirements, which may take a few iterations, it is useful to get them to 'sign off' to show their agreement. Users are often reluctant to commit themselves and it can be quite difficult to achieve this. In theory, the users could write the requirements document themselves, but few will have the necessary skills or experience to make a good job of it. It is generally better to get someone from the development team to write the document on behalf of the users.

It is very difficult to cost a development accurately from the requirements. Generally, this is better left to the specification stage, where a

Imagine you are drawing up the requirements for a library loan system, the sort of things you would need to establish are:

- Is the system for a single library or multiple libraries?
- Is the system only for books or does it cover CDs, cassettes, and videos as well? Will it be necessary to add other item types in the future?
- Are there different types of user, eg adults and children?
- Is the maximum number of items, loan period, and fine policy the same for each type of item? Will it change?
- What is the maximum number of borrowers and loan items the system will have to be able to handle?
- What is the maximum number of people expected to access the system simultaneously? What happens if this number is exceeded?
- What information needs to be stored about each borrower and loan item?
- How long should information on completed loans be stored?
- What is the maximum rate of checking loan items in/out?
- What is the acceptable response time for checking in/out a loan item?
- What other systems will it have to interact with? Does it need to handle loans with other libraries?
- Is access security needed?
- What sort of reports need to be output (eg the most-borrowed books, or a list of all overdue books)?
- Will the public be able to use the system or just librarians?
- Will the system be responsible for printing overdue letters?
- What sort of user documentation is required?

Not considering the answer to any one of these questions at the start of the project could cause considerable expense, delay, and inconvenience later on in the development.

A requirements document template

Cover page
(Title, author, date, version number.)

Glossary
Define terms the user may not know, including acronyms and abbreviations.

Introduction

Scope
The scope and responsibilities of the system.
What is inside and what is outside the system.

Context
How this system relates to other systems (a diagram may be useful).

General capabilities
Describe the main capabilities and why they are needed.
What the system does and doesn't do.

General constraints
Describe the main constraints and why they exist.

User characteristics
Describe who will use the system, when, and why.

Assumptions
Describe any assumptions made.

Detailed requirements
Describe each detailed requirement, clearly distinguishing between the essential and the desirable.
Each requirement must have a unique identifier.
Group by category.

Critical success factors
Identify attributes of the system that are critical for its success.

Acceptance test plans
An outline of the tests that will be undertaken to show that the delivered system meets its requirements.

References
Any relevant documents referenced.

Appendices

Requirement	Specification
SU/E/1	2.2 'Administration', 3.1 'Users' window
SU/E/2	2.2 'Administration', 5.7 'Off-line back-up'

Figure 2 – A simple traceability matrix.

References

European Space Agency Software Engineering Standards (PSS-05)
This is a good framework for software engineering, with some detail on the requirements gathering phase and how it fits into the overall process. *http://dxsting.cern.ch/sting/ESA.txt*

±30% estimate should be possible. Alternatively, if your budget is already fixed you will have to decide which requirements to implement at the specification stage – hopefully, you will have enough time and effort for at least the essential ones.

Change control

Requirements will always change during the life of a project. These changes can be driven by a number of factors. First, the more the users see, the more they want. Second, the user or developer may realise that there is something they have overlooked. Third, new systems create new needs and opportunities. Fourth, there can be a changing business environment. And finally, people do change their minds.

To try and 'freeze' the requirements simply isn't realistic. But allowing requirements to change faster than developers can keep up with will send a project into free fall. The users and developers must agree a 'change control' process for managing changes. The usual procedure is that the user or developer requests a requirements change. Next, the developer decides the impact this will have on the project budget and schedule. Then the user decides whether to approve the change. And if the change is approved, project plans, documents, and code are amended accordingly

In practice, it is often possible to do some 'horse trading', for example adding one requirement and dropping another to avoid schedule and budget changes. If changes to the requirements are agreed, they should be made to the requirements document first, and then to other relevant documents (specification, test, etc) before changing code. This ensures that the documents are consistent and up-to-date. Often it will be sufficient to issue an addendum to a document, rather than re-issue the whole document. The whole change control process should be documented, and a standard change control form makes this easier.

Specification

Once a requirements document has been agreed, the next step is to define the best possible solution to meet the requirements. Generally, this will take the form of a system specification document that defines the behaviour of the new system, including the user interface and file formats, etc.

Note that every item in the specification should be traceable to one or more requirements and every requirement should be fulfilled by one or more item in the specification. If this isn't the case, you have superfluous items in the specification or unfulfilled requirements. Traceability can be shown by a simple traceability matrix in an appendix of the specification. See Figure 2.

A difficult skill

As Lewis Carroll once said, 'If you don't know where you are going any road will get you there'. Developers must resist foul urges to 'get on with the real work' (ie coding) before they understand the problem. Without a good statement of requirements, you might end up producing a brilliant solution to the wrong problem, and then no amount of good design or programming is going to save you.

Requirements gathering is a difficult skill. Some people now talk about 'requirements engineering' as a separate discipline. Certainly the communication skills required for requirements gathering are rather different to the analytic skills required for design and programming. As developments become ever more ambitious, and time-scales shrink, the skill of requirements gathering can only become more important.

Andy Brice is a Principal Software Engineer at scientific and engineering software house QuantiSci (http://www.quantisci.co.uk) and product manager for design rationale management software DRAMA (http://www.quantisci.co.uk/drama).

Rome to Arabia (with Forth)

Neal Bridges believes Forth is the perfect fit for the Palm platform. To demonstrate this, he walks us through a handheld application – a calculator for Roman numerals – from original concept to distribution.

A little history. It was in July 1997 that I heard about the PalmPilot. (If you've been living under a rock, the PalmPilot – now the Palm IIIx and V – is a palm-sized handheld Motorola DragonBall-based computer with an LCD touch-screen interface. It can operate off two AAA batteries for several weeks, and has scooped a huge amount of the market share for Personal Digital Assistants.) As a thinking person, my first question was 'can you write your own software for it?' The answer was 'yes', so I bought one immediately.

After exploring the various options for developing applications (see *Software to go* and *PalmPilot software development*, EXE, February 1998, for another exploration), the thought occurred to me that Forth was a perfect fit for the device (see *A thumbnail sketch of Forth*). With an on-board Forth compiler I'd be able to write real applications anywhere I could take the PalmPilot.

Not much noise is made about the language, but it's lean, fast, expressive, and has an ISO/ANSI Standard (X3.215-1994). Around since the late 1960s, it is the tool of choice for a growing number of developers, and is ideal for embedded systems like the PalmPilot. I set about writing such a compiler – Quartus Forth, which went into release in late December 1998.

A full-featured Forth compiler can be implemented in an amazingly small space. The Quartus Forth kernel is less than 20 KB in size; this includes an optimising compiler, interactive console, and all connections to the underlying operating system.

The next step was to show that Forth is the right choice for Palm development by writing applications to demonstrate how it's done. This article walks through just such an application, from the original concept to distribution.

The application concept

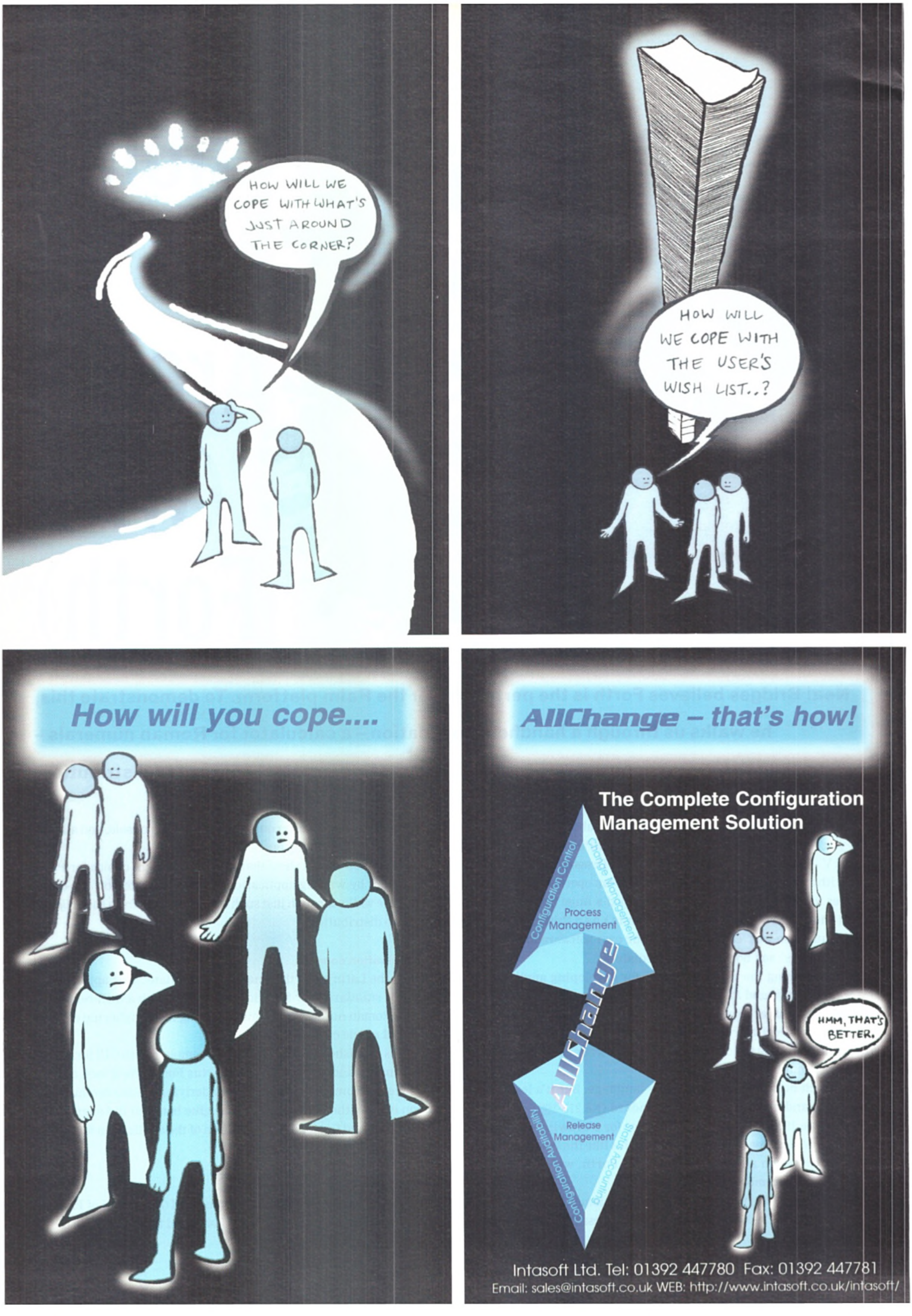
'Duco' is the Latin word that means 'to calculate'. Our application, Duco, is a standard four-function calculator with a difference – it works in Roman numerals. It's just the thing for deciphering cornerstones and movie credits.

Duco is a polished Palm application – complete with GUI form controls, menu, and icon – written in Quartus Forth. We'll look at it in four stages: the inner workings of the underlying Roman-decimal conversion engine, the user interface, the glue between the engine and the interface, and finally the distribution of the application for use on any Palm handheld.

Getting ready

For this exercise, you'll need a Palm handheld (any model of Palm or PalmPilot). You'll also need Quartus Forth and its libraries installed

Chris Rutterford


HOW WILL WE COPE WITH WHAT'S JUST AROUND THE CORNER?
HOW WILL WE COPE WITH THE USER'S WISH LIST..?
How will you cope....
AllChange – that's how!
The Complete Configuration Management Solution
Configuration Control
Change Management
Process Management
AllChange
Release Management
Configuration Auditability
Status Accounting
HMM, THAT'S BETTER.
Intasoft Ltd. Tel: 01392 447780 Fax: 01392 447781
Email: sales@intasoft.co.uk WEB: http://www.intasoft.co.uk/intasoft/

in the handheld, as per the instructions in the manual provided in the Quartus Forth distribution. It's available at *http://www.interlog.com/~nbridges/quartus.html*. The free evaluation version can be used to create and run the application. To generate a standalone PRC (a Palm executable) the registered version will be required.

To create graphical resources, you'll need a resource editor/creator (we will use RsrcEdit, a free on-board form-driven resource editor written by Roger Lawrence, available at *http://www.individeo.net*).

Lastly, each publicly released Palm application requires its own unique four-character creator ID. These can be registered quickly and free of charge at *http://palm.3com.com/devzone/*. Duco has already been registered; its creator ID is 'Duco'.

Useful, but not required, is the Palm OS Emulator. This is an excellent package maintained and supported by 3Com. It's available from *http://palm.3com.com/devzone/*. It runs on both the Mac and the PC, and looks just like a Palm PDA glued to your monitor. It has features for debugging, automated testing, and profiling, and it supports Palm OS ROM images all the way back to version 1.0. You can also use it to take screenshots.

Let's begin with the underlying Roman-decimal conversion engine.

In Roman times, slaves who could handle the multiplication and division of Roman numerals were often given their freedom after just a few years of such a difficult task. Duco skips the harder bits by converting values to decimal before performing any calculations. This is managed via *roman*, a module designed to perform conversions both to and from Roman numerals. See Table 1 for a brief overview of these values.

No more than three symbols of any type may appear together; symbols with a line drawn over them represent their value times 1000. Any symbol preceded by a symbol 1/10th or 1/5th of its value has the preceding symbol's value subtracted from it (IX and IV are examples of this).

The algorithms used in *roman* are of interest in and of themselves. There are four key functions.

The first is `roman>`, which has a prototype of:

`roman> ( addr u1 - u2 )`

It converts a string (represented by the address `addr` and the length `u1`) into an integer `u2`. Malformed Roman strings are converted without error (eg 'XIIIIIIID' will convert to 517). Its algorithm can be expressed as:

Start with 0 as the output integer.

Arabic numeral	Roman numeral	Times 10	Times 100	Times 1000
1	I	X	C	M
2	II	XX	CC	MM
3	III	XXX	CCC	MMM
4	IV	XL	CD	$\overline{\text{IV}}$
5	V	L	D	$\overline{\text{V}}$
6	VI	LX	DC	$\overline{\text{VI}}$
7	VII	LXX	DCC	$\overline{\text{VII}}$
8	VIII	LXXX	DCCC	$\overline{\text{VIII}}$
9	IX	XC	CM	$\overline{\text{IX}}$
10	X	C	M	$\overline{\text{X}}$

Table 1 – An overview of Roman numerals.

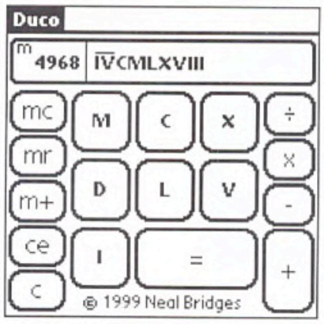


Figure 1 – The calculator's user interface.

For each Roman symbol in the input string (left to right):
Convert the symbol to its decimal value.
If the previous symbol was 1/10th or 1/5th the value of the current symbol,
subtract twice the value of the previous symbol.
Add the resulting value to the output integer.

The second key function of the `roman` module is `>roman`:

`>roman ( u1 - addr u2 )`

This converts a positive integer `u1` into a Roman numeral string at address `addr` with a length `u2`. It handles input values from 0 to 3999 ('MMMCMXCIX') and its algorithm is:

Start with a null Roman output string.
Convert the input value into a string of decimal (Arabic) digits.
For each decimal digit (left to right):
Multiply each Roman symbol in the output string by 10 (I becomes X, X becomes C, etc).
Append the Roman representation of the decimal digit to the output string (note that the digit '0' is represented by a null string).

The third important function is `split-roman`, which has a prototype of:

`split-roman ( u - nnn 1000's )`

This splits the input integer into two parts. If it is less than 4000, the two parts are 0 and the original integer (`nnn`). If the input integer is greater than or equal to 4000, the two parts are the quotient and remainder resulting from dividing the original integer by 1000. This routine is used by `romantype` (see below). The algorithm is:

Divide the input integer by 1000.
If the quotient is greater than 3,
continue;
otherwise
set the quotient to 0, and set the remainder to the input integer.

Finally, for the Roman-decimal conversion engine, there is `romantype`:

`romantype ( u - )`

It converts the unsigned integer *u* into Roman numerals, and displays it. Any part of the result over 3999 is displayed with a line over it, as per the accepted system of displaying Roman numerals. It handles input values from 0 to 65535:

Perform `split-roman`.
Display the quotient as a Roman number, with a line drawn over it.
Display the remainder as a Roman number, immediately following the quotient.

Note that the file `roman` must be present as a memo in your Palm MemoPad. Be sure that the memo starts with a backslash and a space, followed by the filename *roman*, so that it will be recognised by Quartus Forth as a source file.

The interface

Because of the small screen and 160 x 160 pixel resolution of the Palm handhelds, it's important to keep the graphical user interface simple and easy-to-read.

As you can see from Figure 1, the interface will have the expected calculator buttons, but with Roman numerals in place of the typical 0-9. Along with output areas for both Arabic and Roman representations of the currently-entered number, a tiny 'm' shows the status of the 'memory' feature of the calculator. A standard titlebar provides access to the menu items (Help and About).

There is more than one way to create Palm OS GUI resources (ie forms, buttons, menus, etc). PilRC is a script-driven resource compiler that runs on the desktop – available from *http://www.scumby.com/scumbysoft/pilot/pilrc/index.htm*. There are various layout assistants (eg PilotMAG) that run on the desktop and allow for drag-and-drop placement of graphical objects; most such programs produce output in PilRC format. RsrcEdit, as mentioned earlier, is a resource editor that runs on-board the Palm itself, and it presents a simple form-driven interface for creating resources. I used this to create the interface for Duco.

If you want to cut to the chase, you can download a pre-made resource file (along with all sources from this article) from EXE OnLine or *http://www.interlog.com/~nbridges/dist/quartus/Duco.zip*. Rather than walk through each step of creating each button, etc, I would refer you to the zipfile, where the required Palm OS interface features are shown as a PilRC script.

The items not shown in the PilRC script (the 'm' memory indicator, and the frame around the numeric display area) are not provided by the Palm OS, and will instead be drawn later by the Duco setup code.

The next step, using RsrcEdit (or another tool), is to create a resource file containing the above interface objects. Every resource database needs a creator ID and a type; give this resource database the already-registered creator ID of 'Duco' (without the quotes) and a type of 'rsrc'. The name of the database isn't critical; call it 'Duco Resources'. Safeguard yourself and use RsrcEdit to set the backup bit on the database, so that it'll be backed-up to your desktop the next time you HotSync.

The large and small icons for the application are defined using RsrcEdit. Such icons will appear in the Application Launcher on the Palm PDA.

Filename	Needed for
`toolkit`	for `enum`
`core-ext`	for `value`, etc
`roman`	as described in the article
`textalign`	for `type.right`
`bitmap`	for `bitmap`
`condthens`	for `cond` and `thens`
`graphics`	for pixel cursor-positioning, `erase-rectangle`, `line`, and `frame`
`fonts`	for named font types
`resources`	for `use-resources`
`events`	for named event types
`ids`	for `(ID)` (simple creator ID handling)

Table 2 – Library dependencies.

The application

Next, we'll delve into the source for the calculator itself; the 'glue' layer that joins the underlying Roman-decimal conversion engine and the user interface, and provides the calculator logic.

The application is presented as a file called `duco`. Forth source is written such that later code builds on earlier code. Because of this, it lends itself nicely to a top-to-bottom discussion and analysis. We'll take the code a piece at a time.

First, in the prologue, all required library files are included via `needs`, which ensures that each file is only loaded once no matter how many times the same file is requested via `needs`. (This is akin to the common practice of using `#ifndef` and `#define` in C header files to prevent them from being repeatedly included by other source files.)

```
\ duco 99.2.7 5:00 pm NAB
needs toolkit
needs core-ext
needs roman
needs textalign
needs bitmap
needs condthens
needs graphics
needs fonts
needs resources
needs events
needs ids
```

All of these files (except *roman*, as described above) are Quartus Forth library files, and are part of the distribution package. See Table 2 for more detail on the library dependencies.

The tiny 'm' on the user display, which indicates that a value is stored in the memory of the calculator, is defined as a bitmap:

```
5 7 2constant m-size
20 5 2constant m-position
m-size bitmap memory-indicator
2 base !
1110110000000000 ,
1001001000000000 ,
1001001000000000 ,
1001001000000000 ,
1001001000000000 ,
decimal
```

The bitmap is defined using the `bitmap` word from the `bitmap` library file. We take advantage of Forth's ability to work in an arbitrary numeric base; here we use base 2. Dark pixels are indicated with '1', clear pixels with '0'. Once defined, `memory-indicator` becomes an active word that can draw itself anywhere on the Palm's screen. Two double-cell constants are defined (`m-size`, `m-position`) that describe the size and eventual position of the 'm' on-screen.

Next, we define the variables the calculator uses internally:

```
\ For interim values:
0 value a
0 value b
2variable display
\ High-cell of display indicates range errors:
display constant out-of-range
0 value memory
```

The values `a` and `b` hold interim values within the calculator. A 32-bit double-cell variable `display` is defined; our calculator only works in the 16-bit range of unsigned integers, so any calculation exceeding this range will result in a non-zero value in the high-cell

of display. We name this cell out-of-range, and by checking it we can alert the user to underflow/overflow errors during normal calculator operation. The calculator's memory will be held in the value memory.

In the following, the four words get-display, set-display, zero-display, and error are used to retrieve and update the display variable (they serve as shorthand later in the code):

```
: get-display ( — u ) display 2@ d>s ;
: set-display ( d. — ) display 2! ;
: zero-display ( — ) 0. set-display ;
: error ( — ) -1. set-display ;
```

We then come to the first part of the calculator-specific logic. The add-symbol routine acts as an intermediary between the calculator and the *roman* module: it adds a new character to the end of the Roman representation of the currently displayed value, converts the resulting Roman numeral string back to an integer, and stores it back.

```
: add-symbol ( char — )
  >r
  get-display ( u )
  split-roman ( nnn 1000's )
\ Add char to end of low Roman:
  swap >roman 2dup + r> swap c!
\ Convert back to integer:
  1+ roman> ( 1000's newnnn )
\ Combine new nnn and 1000's:
  >r 1000 m* r> m+ set-display ;
```

By converting the input to a Roman representation and back after each symbol is added, we 'clean' the input and prevent the user from entering a malformed Roman number. For example, should the user push I five times, V will show on the calculator's display.

The opcodes for the calculator are defined using an enum (from the *toolkit* library file):

```
\ Opcodes:
0 enum OpType
  OpType do-nothing
  OpType add
  OpType subtract
  OpType multiply
  OpType divide

do-nothing value nextoperator
false value pendingequals

: clear ( — )
  zero-display
  false to pendingequals
  do-nothing to nextoperator
  0 to a 0 to b ;
```

A clear word has been defined; it is the action that will later be associated with the 'C' (clear) button in the GUI. The value pendingequals is an internal flag that indicates to the calculator engine whether an arithmetic operation is pending.

Next in the duco file, the refresh routine updates the on-screen display:

```
: refresh ( — )
  boldFont font drop
\ Clear Roman display:
  16 107 22 44 erase-rectangle
\ Clear Arabic display:
  14 32 24 6 erase-rectangle
\ Display Roman:
  24 45 at
  out-of-range @ if
    s" error" type zero-display 0
  else get-display dup romantype
  then
\ Display Arabic:
  0 <# #s #> 38 type.right
\ Memory indicator:
```

A thumbnail sketch of Forth

The fundamental building block is called a 'word'. This is akin to a function or subroutine in C or Basic with a key difference: a word defined in Forth becomes a 'first-order operator' – no special syntax is needed to invoke it. It becomes, effectively, a keyword in the language.

Parameters are passed via a first-in, last-out stack called the *data stack*.

A simple example

Given the example:

`3 5 + 7 * . <Enter>`

here's a breakdown of what the Forth interpreter does. A stack diagram is shown at the end of each line; these are commonly seen in Forth source code. Each stack diagram has the format (before — after) showing the contents of the data stack both before and after each operation:

Parse 3 ; translate it to binary and push the number onto the data stack. (— 3)

Parse 5 ; translate it to binary and push the number onto the data stack. (3 — 3 5)

Parse + ; find it in the Forth dictionary and perform its action, which is to pop two numbers from the data stack, add them, and push back the result. (3 5 — 8)

Parse 7 ; translate it to binary and push the number onto the data stack. (8 — 8 7)

Parse * ; find it in the Forth dictionary and perform its action, which is to pop two numbers from the data stack, multiply them, and push back the result. (8 7 — 56)

Parse . ; find it in the Forth dictionary and perform its action, which is to pop a number from the data stack, convert it to Ascii digits, and display the result on-screen with a trailing space. (56 —)

The end of the input-buffer has been reached. Display the prompt ' ok', perform a carriage return, and wait for another line of input from the user.

The screen will show: `3 5 + 7 * . 56 ok`

There are a number of standard built-in words. For example, the word DUP duplicates the top number on the data stack. Here's the definition of a word that squares the number on top of the stack, and displays the result:

```
: SQUARED ( n — n*n ) DUP * . ;
5 SQUARED <Enter> 25 ok
```

The text between the round brackets is a comment in Forth source, just as is any text on a line following a backslash.

The Complete Solution for Software Distribution

InstallShield 5.5 Professional

The complete professional installation development system.

• Windows 98/NT logo compliance • Integrated Installation Development Environment • New VB 5.0 and VB 6.0 Project Wizard • Dynamic File Linking • Advanced Media Building Technology • Function Wizard • Visual Registry and Visual Shell Objects Editor • New Templates for ODBC 3.5, BDE 5.0, PowerBuilder 6.0 and Access 97 • SMS installation / uninstallation support, Command-line build for batch processing • includes PackageForTheWeb™ 2 for creating single self-extracting EXE or CAB files

InstallShield Express 2.1

A visual, point-and-click installation development system.

• Automatic handling of common Windows 98/NT logo compliance issues • New VB 6.0 Express Setup Wizard • Point-and-Click support for installing common components • Express Extensions allow calls to external DLL functions and launch EXE's from within the installation • Language support for English, French, German, Italian, Spanish, Dutch, Finnish, and Swedish

DemoShield 5.4

A quick and easy way to give your software the competitive edge.

• On-the-Fly Editing • New Demo Wizard and Template Demos • Enhanced Multilingual Support • ActiveX controls for Web-enabled demos • Point-and-Click Design Environment and Media Libraries for simple multimedia authoring • provide links to demos, Web pages, or other applications from within your demo • Setup Wizard for creating a customised setup for distribution via CD, floppy disk, or the Internet includes built-in macro recording and screen capture tools to simulate live action in your demo

InstallShield® Software Corporation

NEW FROM INSTALLSHIELD SOFTWARE CORPORATION

InstallShield for Windows CE

InstallShield for Windows CE is a PC-based software solution for easily creating the installation files for Microsoft Windows CE-based applications and Microsoft Windows CE-based desktop companion CE applications.

InstallShield for Windows CE supports the desktop-to-device, internet-to-device, and PC Card-to-device install scenarios for delivering your application. Intelligence about specific devices and processors is built into ISCE, insulating the engineer from the particulars of the form factor(s) they are targeting.

InstallShield for Windows installer

InstallShield for Windows installer is designed to support the new Microsoft® Windows installer service. This enables an application to receive the Windows 2000 logo and to support new Total Cost of Ownership reducing features like component level install/uninstall, application advertising, auto-repair of corrupt application components, and automatic rollback after a failed installation.

It All Starts Here®

Authorised InstallShield Reseller

QBS Software Ltd.
Tel: +44 (0) 181 956 8000
Fax: +44 (0) 181 956 8010
Email: info@qbss.com
Web: www.qbss.com

InstallShield®
Software Corporation
www.installshield.com

```
memory if
    m-position memory-
    indicator
else
    m-size m-position
    erase-rectangle
then ;
```

It uses words from *graphics*, *roman*, *typealign*, *fonts*, and the bitmap we defined earlier, `memory-indicator`. Both the Roman and Arabic numbers are drawn at the appropriate places on the screen, and the memory indicator is drawn or cleared based on the presence of a non-zero value in `memory`.

Four words – `args`, `interim`, `equals`, and `operator` – provide the basic logic of the calculator itself:

```
: args ( – b. a. )
  b 0 a 0 ;

: interim ( – )
  a to b get-display to a ;

: equals ( – )
  pendingequals if interim then
  false to pendingequals
  nextoperator cond
    dup add = if
      args d+ set-display
    else dup subtract = if
      args d- set-display
    else dup multiply = if
      args drop nip m* set-display
    else dup divide = if
      a if
        args drop
        1 swap m*/ set-display
      else error
  thens drop refresh
  get-display to b ;

: operator ( opcode – )
  pendingequals if equals then
  to nextoperator
  true to pendingequals
  interim zero-display ;
```

The `equals` word performs the actual calculations based on the most recent opcode. Error checking is performed to ensure that division by zero is not attempted.

Each of the buttons in Duco's user interface has an associated button number. 'M' is button 2000; 'C' is button 2016. To make the following button dispatch code more readable, `item>button` is used to map the button numbers onto Ascii characters. It also maps the correct Roman symbols to the Roman numeral buttons on the calculator:

```
\ Button ids run from 2000 to 2016.
2000 constant first-item-id

\ Map form ctlEvents to chars:
: item>button ( id – char )
   first-item-id -
   s" MDCLXVI/*+-=wraek" drop + c@
;
```

The following `do-button` routine is the dispatch engine for pushed buttons. Every button is mapped to an action in a `case/end-case` structure. The Roman numeral buttons are handled as a group by the final `else` clause:

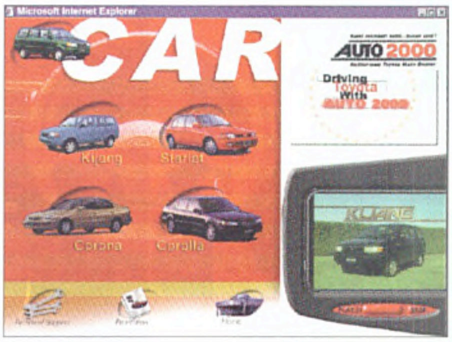
```
\ Actions for each button:
: do-button ( char – )
  cond
    dup [char] k = if clear refresh
    else dup [char] e = if \ clear error
      zero-display refresh
    else dup [char] w = if \ mc
      0 to memory refresh
    else dup [char] a = if \ m+
      get-display memory + to memory
      refresh
    else dup [char] r = if \ mr
      memory 0 set-display refresh
    else dup [char] + = if
      add operator
    else dup [char] - = if
      subtract operator
    else dup [char] * = if
      multiply operator
    else dup [char] / = if
      divide operator
    else dup [char] = = if equals
    else dup add-symbol refresh
    thens drop ;
```

There are other ways, arguably more effective, to set up a conditional structure like this. The code shown (roughly equivalent to a `switch` statement in C) works for our purposes and is conceptually quite simple. To simplify the conditionals, `cond/thens` (from the *condthens* file) is used; we can create an arbitrarily deep `if/else` structure without having to match up `then` statements at the bottom.

Next, to make use of the resources we defined earlier, the `Duco/rsrc` resource database is opened via `use-resources`. The function `show-panel` establishes the buttons and the frame around the numeric display area, as well as the line dividing the decimal and Roman display areas.

Figure 2 – Duco in action (Duco.PRC is approximately 7 KB in size, quite a reasonable figure for a Palm application).

What Good Is An Object Tool Without An Object Database?



Attracting and retaining customers requires applications that are not only entertaining and informative, but also easy to use.

Today's business information is made up of not only text and numbers, but also images, motion, and sound.

Jasmine's object technology is designed for building enticing systems that you get to market faster, develop more rapidly, and gain productivity through reuse and integration of code, data and applications.

Only a pure object database like Jasmine inherently understands how to store and process complex data. Complex queries, such as fare comparisons for specific destinations and dates, are easily implemented with Jasmine.

Jasmine Works With Your Favorite Tool

- Sun:
  - Java Studio
  - Java WorkShop
- Microsoft:
  - Visual InterDev
  - Visual C++
  - FrontPage
  - Visual J++
  - VisualBasic
  - Visual Studio
- Symantec Visual Café
- Inprise JBuilder
- Bluestone Software
- Centura
- SilverStream
- SuperNova
- Dataviews Corporation
- Trellix

Just think how much more powerful your object tool would be with an object database like Jasmine™. With Jasmine's pure, object-oriented architecture, objects defined in your application tool (C++, Java, and others) are the same as the objects managed by Jasmine.

INFORMATION IN HARMONY™

You can use native language bindings, rather than learning and coding new non-object-based data access protocols such as JDBC. And Jasmine supports large amounts of complex data so you can deliver true, next-generation applications.

With Jasmine, objects never have to be converted to relational formats, so they run much faster. And there's no need to choose between object models—Jasmine supports both COM and Java. It even lets you use different tools at the same time. Jasmine's object infrastructure also integrates external data and applications, making them accessible from any application development tool.

That includes the object tool you're using right now. So call **01753 679679** or visit **www.cai.com/ads/jasmine/dev** for your FREE, full-function Jasmine Developer Edition CD. And turn your object tool into an instant advantage.

COMPUTER® ASSOCIATES
Software superior by design.

Jasmine™

```
(ID) Duco (ID) rsrc use-resources
1100 constant DucoForm

: show-panel ( — )
\ The buttons:
  DucoForm ShowForm
\ The display area:
  20 154 19 3 dialogFrame frame
  39 40 19 40 line ;
```

We have defined a constant named `DucoForm` with the form number of the Duco form. This is only used once in `show-panel`. Nonetheless, it's good practice to name all 'magic' numbers in source code – it aids readability and maintenance.

Figure 3 – A new standalone application in the Application Launcher.

The `help/about` routine is responsible for displaying the 'About' box, and displaying the 'Help' text. This is the only place in the source that we directly call functions from the Palm OS ROM – `FrmAlert` and `FrmHelp`.

```
3000 constant AboutBox
3001 constant HelpString
2001 constant AboutMenuItem
2002 constant HelpMenuItem

: help/about ( — itemid )
  event >abs itemid cond
    dup AboutMenuItem = if
      AboutBox FrmAlert drop
    else dup HelpMenuItem = if
      HelpString FrmHelp
  thens ;
```

Events are dispatched in the following: menu events are passed to the `help/about` routine, button events to `do-button`.

```
: do-event ( ekey — )
  dup menuEvent = if help/about
  else dup ctlSelectEvent = if
      event >abs itemid
```

```
        item>button do-button
      then
    then drop ;
```

And this is where it all comes together. Here is the main entry point, named `go`:

```
: go
  clear show-panel
  0 to memory refresh
  begin ekey do-event again ;
```

Building on all the preceding code, `go` clears the calculator, displays the graphical user-interface, clears the calculator 'memory' feature, refreshes the display, and then drops into a simple event-handling loop.

That brings us to the end of the *duco* source. Make sure *duco* is installed as a memo in your Palm MemoPad application. Again, as with *roman*, be sure the memo starts with a backslash and a space, so that Quartus Forth will recognise it as a source file.

The app is ready to be tested within Quartus Forth. Start this by tapping on the icon from the Application Launcher. Type:

```
include duco <Return>
```

where `<Return>` is a Graffiti Return – a single stroke in the Graffiti area from upper right to lower left.

After a few seconds, and assuming no errors in the source, Quartus Forth will report ' `ok`' and return you to the input prompt. The source of Duco has been compiled into native machine code. Recall that the main entry point for Duco is a word named `go`. Type:

```
go <Return>
```

And Duco will run! (See Figure 2.)

Running the application from within Quartus Forth is fine for testing, but for a public release you'll want a standalone executable (see Figure 3), a PRC file. The registered version of Quartus Forth generates such files.

There are a number of distribution points for Palm software on the Web. Foremost among these is PilotGear HQ at *http://www.pilogear.com/*. They have an enormous collection of freeware, shareware, and commercial software. If you're so inclined, they'll take orders for your software and send you monthly cheques.

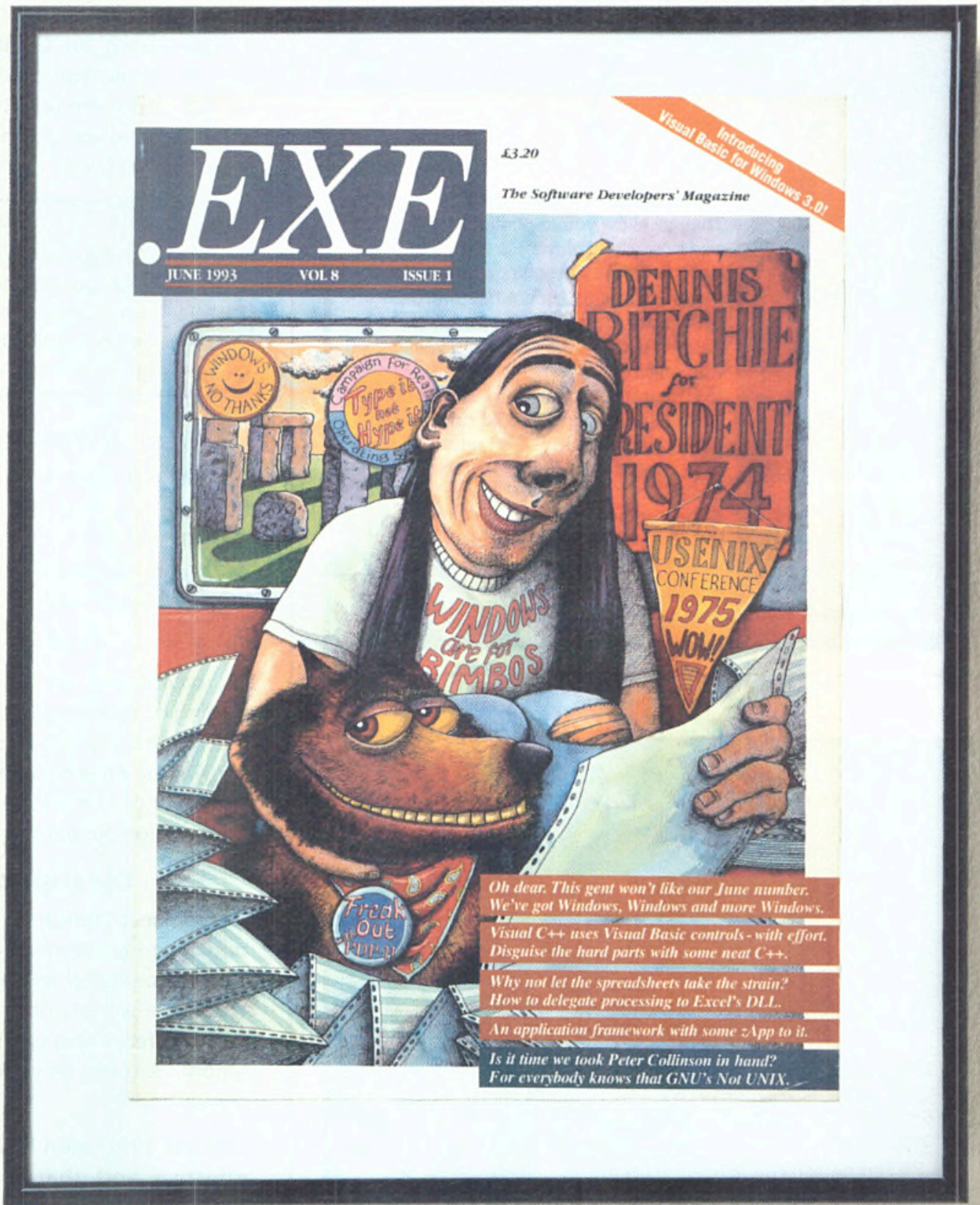
Making a product available at PilotGear is a simple matter of following the instructions they provide. Duco is already there. I look forward to seeing the Palm applications you create with Quartus Forth!

Neal Bridges is a developer specialising in handheld systems. He can be contacted at http://www.interlog.com/~nbridges/, or via email at quartus@interlog.com.

The full code listings are available at ftp.exe.co.uk/pub/exestuff/9905_forth.

Unix in perspective

On the occasion of his 100th EXE article, Peter Collinson looks at his changing Unix world. From university days with UUCP and Usenet News through the arrival of Solaris to the emergence of the World Wide Web.



This article is the 100th feature that I have written for EXE. The first one was printed in June 1990, so it's taken a little under nine years to reach this milestone. If you do the arithmetic, perhaps you'll think that there were several editions that didn't contain an article from me. However, the old .EXE publication diary was not graced with a January edition until 1995, because the editorial team liked their Christmas holidays. In the early years, writers produced eleven chunks of text per year. I've actually missed only one edition: October 1992.

A week is supposedly a long time in politics. Nine years is a *very* long time in computing. In 1990, I'd been freelance for a year, having left my tenured employment as a lecturer in Computing Science at the University of Kent. At the university, I was responsible for the development of Unix on the campus. We were the first university in the UK to use Unix as part of the Computing Service, running the first VAX 11/780 Unix system in the UK.

Due to the foresight of our boss, Brian Spratt, we had a well-developed network on the campus using Cambridge Ring technology at a time before Ethernet was commercially available. Users could go to a terminal room somewhere on campus and transparently log into one of several machines to get their work done. Machines could share printers and tape devices. We had an exten-

sive email system for all users. Many system operations were provided by client/server applications, well before that term became a marketing buzzword.

Of course, we didn't understand the novelty of what we had done, and we told no one really, because it didn't seem exceptional and it appeared somewhat parochial. It was development that solved our day-to-day problems and was not research. I don't think that I understood what had happened until years later when I attended conferences and heard papers re-solving the problems we had encountered and solved.

The need for a Sun

Latterly at Kent I'd been involved with the creation of the UK UUCP network, connecting UKC to the world via Holland and fanning the network out to both commercial and academic sites in the UK. The UUCP network supported the Usenet News system and carried email. It created the appetite that was later going to explode into the Internet.

Actually, the UUCP network wouldn't have started at all without the Joint Network Team (JNT) that provided a free backbone – JANET – to academics in the UK. The JNT supremos kept congratulating themselves on all the interactive terminal traffic that was travelling on their network. However, a goodly proportion of the bytes was UUCP traffic conveyed from machine to machine by logging on using a terminal connection. I have to acknowledge Lee McLoughlin who did the work on UUCP to make all this possible. The University spun off the network in the early '90s. It became EUNET GB and then was sold to PSInet.

However, as I planned to leave the University I couldn't face life without a Unix machine. Well, the cheapest choice was probably a version of Xenix running on PC hardware, but this lagged the developments in Unix by several years. At that time, to run anything from the Net you needed a Sun. The ability to run code from the Net was important because the X window system was the GUI of choice for us academic Unix folks. Once you had become bitten by that way of working it was hard to contemplate returning to a single text-based console supported by the extant commercial Unix offerings.

I scraped together our family resources and decided to buy a Sun386i. Its operating system was based on 4.2BSD and that was the system I was used to. It was the cheapest Sun workstation that was available at that time. It cost a little less than £10,000. I think that it had about four megabytes of memory, four gig of disk space, and a tape streamer. It was graced with a 19" monochrome monitor and an optical mouse. Prices have dropped considerably in numerical terms. I am unsure how to compare a ten-year-old price in sterling with one today.

There were many problems with the Sun386i. For example, its software was not quite 4.2BSD and its internal clock was massively inaccurate. The main problem was overheating. In the summer, my non-air-conditioned office became hot and the machine turned itself into a toaster. I remarked in an article at the time:

'It's early May and we have been having a heatwave in the UK. My Sun386i has been suffering a little from heat exhaustion and it began to weep, wail, and gnash its teeth mightily about servos not working properly. Allowing the machine to cool down meant that all was well, so I decided that I should take the opportunity to remove the filth that builds up inside the machine clogging various grilles and fans. A clean grille improves the internal airflow and cools the disk down just below marshmallow browning temperature. A user-serviceable air filter would be a good add-on for these machines. However, while working out which four screws were needed to remove the front grille, I removed the plastic skin on the back of the machine – to discover a lot of embossed signatures.' The names of all the people involved in the project had been placed permanently in the metalwork of the casing. A nice touch.

I was more than pleased when later Sun decided to get out of Intel machines and made me an offer of a cheap upgrade to a SPARCstation I. It had loads more Mips and a more 'standard' operating system. I've upgraded the machine twice since then, and it needs upgrading again.

My machine was connected to the UUCP network using a 19,200-baud link. I had two 'local connection' broadband modems at the end of a direct connection from my house to the UKC campus. The connection was 'a BT special' called an EPS8 circuit, designed to support private phone networks. There were four continuous pieces of copper that joined both ends – no amplifiers, no packet switching, just electrons. The connection speed that modem technology supported was dependent on the length of the wire, so the wire must have been less than 11 km. The line meant I was continuously connected; mail arrived with me shortly after it arrived on UKC's machines, and my mail hit the network just after I sent it. Since I was a consultant to the networking company, I was not charged either.

Going freelance

I was out of the University and started to look for things to do. I was lucky to have a 'godfather' at the time, someone who I could ring up and say 'give us a job'. I was also doing occasional technical stuff for EUNET (the network ran using loads of my software and this needed tweaking). Out of the blue, someone who I vaguely knew in the US mailed to ask if I was interested in writing regularly for a new magazine that he was starting. He'd read a couple of my pieces, which were irreverent histories of Unix (these are still available on the Web, see *http://www.hillside.co.uk/articles*). This seemed a good idea and I agreed to write for *SunExpert*. Actually, by the time I started to generate copy for the magazine there had been a palace revolution and my friend was no longer involved.

After I'd done four articles, it seemed possible to sell the articles to a UK magazine and I started to look around for one that might be interested in them. At the time, I hadn't appreciated the essential difference between the computing press in the US and the UK. In the US, there were (and are) many magazines devoted to computing that mostly contain what might be loosely termed 'technical stuff'. People who work as computing technical staff write the bulk of the magazine presenting technical information to the reader. In the UK, there were (and are) very few magazines of this type. In the UK, the computing press tends to consist of news or review material written by computing journalists. They are actually very light on material that allows you to move on. Okay, the magazines do have technical sections, but they are generally quite short pieces tucked away at the back.

I hit on .EXE, I don't quite remember how. I was lucky with the timing. The magazine started with a mission never to mention Unix.

My initial contact coincided with a new editor who was prepared to take a chance on running a regular Unix column (thanks Will Watts). Incidentally, all the early contact was done on the phone or via email. I didn't meet any EXE staff for around four years.

There were two initial articles based on material that had been published in the US magazine, but it quickly became apparent that the target reader for EXE had a different profile, and so different work was needed. Actually, looking over the list of articles, they are split into several categories. There are 'techie articles on some neat thing', usually about something I was doing at the time of writing. There are 'bits of Computer Science wrapped up in some small project', again sparked off by current work. There are 'software or system reviews', often some system or product that had popped up in my life and seemed interesting or topical. (I once tried to review something I didn't like and really could not write anything about it.) Finally, there are 'interviews'. I've done interviews with several notables in the Unix world and its fringes. In general, I've tried not to put my own spin on what people have said in interviews. These are people with things to say and it has seemed better to allow them to speak in their own words, albeit edited into a sequence by me.

Whither Unix?

I've used Unix for the greater part of my working life. Long enough to stop being a Unix anorak, anyway. Of course, it is a moving target that's changed and adapted to the requirements that have been placed on it; I used to say Unix got big by never needing to say 'no'. It has been and remains very successful (although it is common to hear people say 'Unix has never fulfilled its promise'). Its success is based in no small measure on C, the best high-level assembly language ever devised.

Unix has been successful by encouraging shared developments on a wide variety of platforms and by allowing the distribution of those advances in source form, allowing porting and bug fixing. I also think that the Unix process model is a healthy way to develop new software and ideas. It's good engineering to be able to put some new development into a small container so that when it breaks and dies, as it must, it doesn't take the system with it.

One of the eternal Unix problems has been the tendency for splits into several competing camps, each with some element of the 'not invented here' syndrome. In the time that I've been freelance, two developments have eased this problem somewhat. First, we've seen the acceptance of the Posix standards so each system supports a certain base-level of functionality, both at the system call and command level. Second, we had a temporary merger between companies, which allowed the commercial systems that derived from 4.2BSD to merge with those that derived from System V. There were benefits for both sides. The System V camp obtained proper virtual memory support and some of the file system developments that had grown on the 4.2BSD-derived world. The 4.2BSD camp obtained many revised and updated utilities that the System V world started life with.

The revised system, of which Solaris is the prime example, does suffer from software 'bloat' because the kernel needs to support features from both sides. I had to add more memory to my Sparc system to run Solaris comfortably. However, in this case the ability to obtain the functionality is more important than the amount of code needed to achieve it.

The desktop battle

Unix has certainly lost the 'battle for the desktop', with Windows becoming pre-eminent. The reasons for this are many and varied. I think that the bottom line is that Microsoft has delivered applications that people want to use. I certainly run Windows NT because it supports several applications that I use on a daily basis. However, I do suspect that there is evidence that people are becoming annoyed with the lack of compatibility between successive releases of the same Microsoft product. Change for the sake of change seems to be the order of the marketing day.

It's possible that the lowest level of Unix communication, the text file, is simply not good enough as a primary entity, especially as we've moved towards more image and audio processing. The OLE system on Windows allows users to deal with objects, where the object can be text or a binary file containing images or audio. The OLE system is imperfect, as you'll find if you attempt something that's complex. I once tried and failed to convert a load of files automatically by writing a script to convert a file from one image format to another, and then applying that script to several hundred files. The problems may have been simply down to the implementation. The application that I was using doesn't seem to worry about memory usage; it just appears to get more and more without re-using it. Batch operations of this kind are rarely attempted on Windows systems – users have become accustomed to being the machine controller themselves by sitting and clicking away.

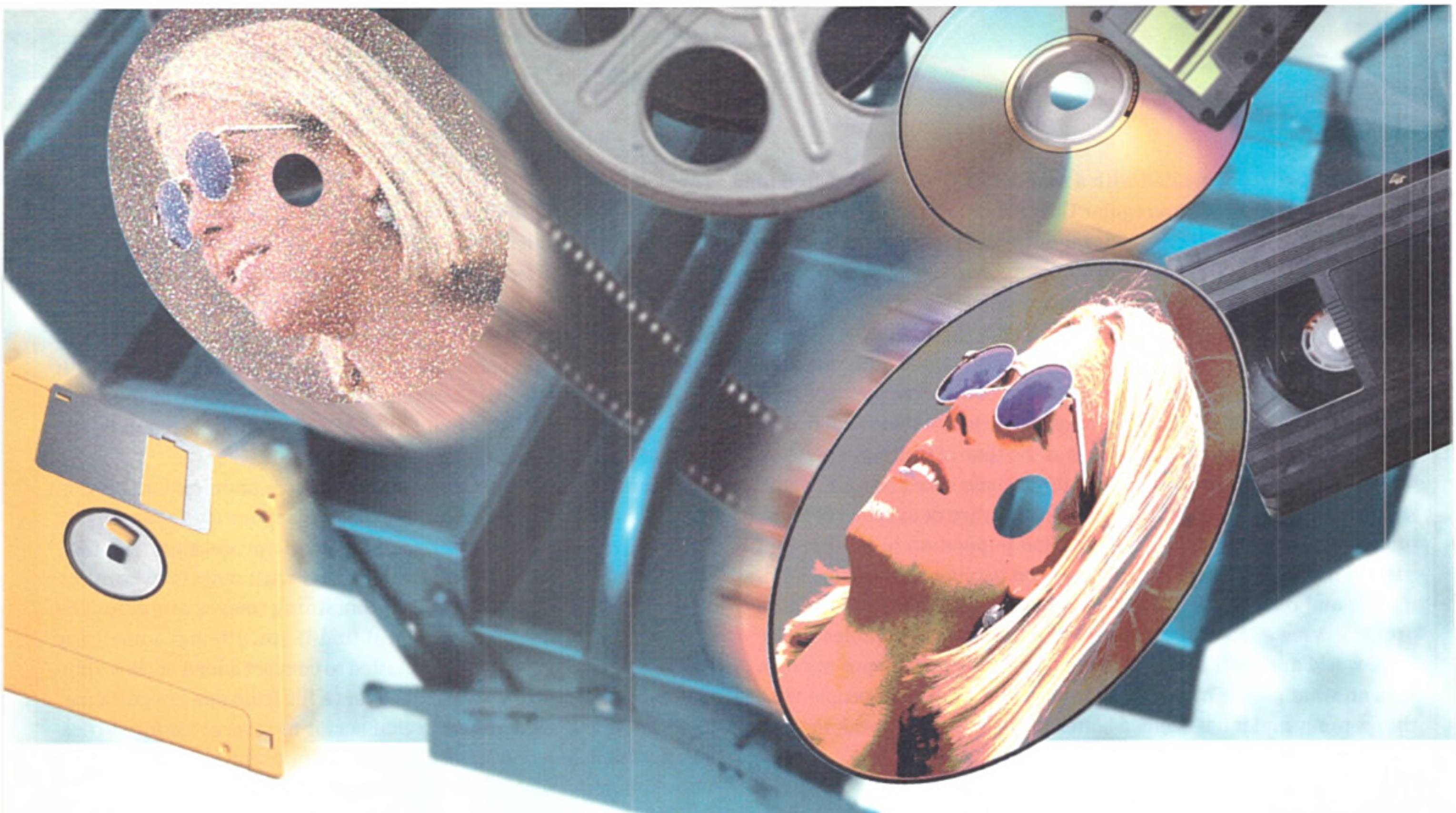
Unix probably never had a chance to provide a desktop that would be acceptable to mortals. Of course, X11 has a much harder job than its equivalent on Windows; it's much simpler to generate a windowing system on a single-user machine running a single program. However, the X window system has always been too 'technically lead' and contains too many mysterious ways to configure and use applications. The X Consortium started life trying to appease several camps in different companies each with agendas of their own. There was no remit for draconian decisions. Flexibility was the order of the day.

I am also sure that development has been inhibited because the industry standard look-and-feel, provided by Motif, demands a licence fee. Free software developers don't want to pay for their development tools. There would be a greater penetration of a standard 'look and feel' if the standard industry widget set was free to all.

It remains the case that program configuration for X11 is done using a bunch of incomprehensible control files. A few systems have tried to tackle this. The CDE (Common Desktop Environment) on my Sun allows a certain amount of tailoring supported by a GUI or two. And I notice that my newly booted Linux system has several mechanisms that attempt to make all this much easier.

The server

I suppose the jury is still out on the question of whether Windows NT will supplant Unix in its role of network server. There are questions of reliability and robustness, and I can never get a good answer on this. People who I ask about their NT server usually tell me (somewhat defensively) that they don't have problems with it. This conflicts with my personal experience of NT. I find that I have to reboot my system regularly, apparently to free up resources. I actually turn it off at night, and usually have to reboot it at least once a day to make it work



again. I suppose that I should blame the applications that run on the machine rather than the system itself. However, I must contrast this with my Unix systems that run continuously, and just sit there doing the work until I decide that I need to change something fundamental, or the power goes off.

I think that Windows NT probably has queries against its security and its proof against hackers. Its main protection at the moment is obscurity; nobody knows how it works and not many people can see the code. But in the long run, security by obscurity doesn't work. Hackers will experiment to find holes and will exploit them. The Unix industry has problems, but is, in general, very responsive to these issues.

I was beginning to wonder whether Unix had actually run out of things to do, when up popped Linux. This offers 'the Unix experience' to a whole new generation: the ability to create their own programs by reading and modifying other people's work. I've taken some small steps to get a bootable Linux system, and am pleased with what I have found.

The Web

One technology that begins to make an appearance in my articles in mid-1995 is HTML and the World Wide Web. I'd put myself 'properly' on the Internet in 1994, using ISDN with EUNET GB. When EUNET GB was sold to PSInet, it became clear that I was going to have to do something else. I started to use Internet Network Services who supply me with a leased line connection. At the time I switched, I was selling BSD/OS into Europe and we were doing a considerable amount of support for customers over the Web. I am sure that the web pages massively reduced the total number of stupid questions with which we had to deal.

The early HTML articles in EXE concentrated on how to create pages, which at that time was a matter of creating text with a text editor, rather than using some WYSIWYG tool. Actually, the WYSIWYG tools tend to create rather inefficient HTML, often using new features that are not necessarily supported by older browsers. They seem to be an opportunity for Mr Gates' company to get you to use its private technologies without you knowing. I've had a case of this recently, where some pages created by a customer killed my Netscape browser.

The Web has changed dramatically since its beginnings. The early pages were grey, with text that was large and spread impossibly widely across the page. The Web is now filled with colour, and people are designing 'small'. Typically, web pages are designed with characters that are one size smaller than the standard, which must annoy Macintosh users whose characters are smaller than those on the PC.

Interestingly, the Web is also inhibited by wars between browser manufacturers whose 'not invented here' antics cause people doing web page development considerable headaches. At the moment, I am trying to use only aspects of the Web that I know will work in

all browsers. Well, that's not quite true; I'll use some of the newer stuff for decoration as long as I know that browsers will degrade, and the decoration doesn't impact on the general accessibility of the site.

I find in my travels to the offices of clients of our web service that the larger ISPs are pretty good at giving away CDs with free software that contain ancient browsers. Some ISPs are so big that they have forced their users to use ancient browsers and tell web page providers that they had better support them. This reasoning seems kind of cockeyed to me.

I suppose another change is in the nature of our work for web clients. We find ourselves creating more and more pages from databases, allowing the client to change the data that's displayed for their site without having to mess with the actual nitty-gritty of generating pages. Web page design is more and more about real programming as time goes on.

The Web is becoming a place where you can find out useful information on everyday issues. Things like the times of trains, the cost of a stamp, the settings on a disk to allow you to add a new disk to your system, information from Mr Gates' company, and so on. I suspect that people are still cautious about supplying their credit card information, even when secure servers are involved, unless they are confident that the organisation processing the card is likely to be okay. For example, many people are prepared to book conferences with the Usenix server that I run, with a credit card. They are less happy to buy objects that need to be shipped to them. There are signs that this is changing, but on balance uptake appears to be slow.

Here's to the next 100

I should end by thanking the EXE staff for the excellent sub-editing that takes the text and improves it into the article that you see; and mentioning the feedback I receive.

Writing for magazines used to be a lonely business, but I often get email or even phone calls as a result of creating these columns. Mostly, I like this type of contact from readers. Phone calls are sometimes not welcomed, especially at the weekend, and especially when the caller has undoubtedly consumed some alcohol and won't let me off the phone. Email sometimes corrects the inevitable inaccuracy. Sometimes it is due to the sender reading what they thought I said, rather than the words that were printed. Sometimes it is abusive. But, best of all, sometimes it is laudatory. Email is always answered to the best of my ability.

Thanks, then, to all these people for comments, both good and bad. In the words of John Ebden (nearly): 'If you have been, thanks for reading.'

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.

Before you learn Java, here's a lesson in economics

Cost of attending Bruce Eckel's
"Hands on Java Seminar".................................$1500
Plus travel expenses...$????
Total...$Lots

Or

Free UK Shipping

Buy the "Hands-On Java" CD-ROM ..£39+vat

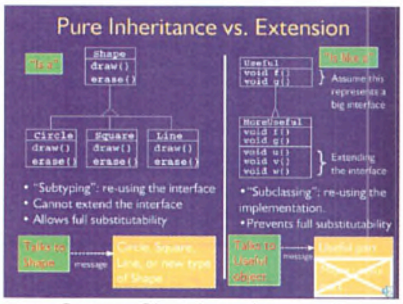
A New Standard in Computer Based Training

Bruce Eckel's Hand's-On Java CD is a professional tool for serious students of the Java programming Language. Every aspect of Java is treated in depth, not just introduced. The CD contains more text, more audio and more source code than any other Java CD-ROM. It goes beyond in-depth coverage of JAVA 1.1 to explore 1.2 topics such as Swing and new collection classes.

Course Features:

CD contains the entire contents of Bruce Eckel's acclaimed "Hands-On Java" 5-day intensive seminar.

- 14 hours of audio lectures, synchronised with 500 slides of information
- Lectures begin with Java syntax and progress to advance topics such as network programming, AWT, Beans and RMI
- Full JDK 1.1 coverage
- Over 15,000 lines of source code in 320 Java examples
- The CD contains two versions of Eckel's excellent "Thinking in Java" book:
  - A printable version
  - A screen formatted 2,500-page hyper-linked version for easy viewing. Includes 230 chapter, section and sub-heading hyper-links and 3600 index links.
- The CD contains approx. 600MB of content.



Screen Shot from lesson 3

CD Cover

About Bruce Eckel - (www.eckelobjects.com)

Bruce Eckel is an award-winning author, chair of the Java programming track at Software Development'98, and one of the most well known independent authorities on Java. Bruce, a 12 year veteran presenter, won the much coveted Software Development Magazine's Jolt Cola Award for programming book of the year in 1996.

Using "Hands-On Java" CD-ROM

Select the lesson you require, listen to the lively audio presentation whilst you observe well-constructed slides on screen. At any time you can interrupt the lesson to be taken directly to the relevant chapter in the 'Thinking in Java' book. Complete the supplied exercises to reinforce the lessons.

To order your £39 "Hands-On Java" CD
Call 0181 317 7777
sales@siliconriver.co.uk

Silicon River

Silicon River Ltd, 106-108 Powis Street, London, SE18 6LU. TEL: (0181) 317 7777, FAX: (0181) 316 7778, email: sales@siliconriver.co.uk www.siliconriver.co.uk

A word to the wise tester

Francis Glassborow shows how to test your classes by incorporating a test harness as a static function. And he highlights a must-read book.

Most of us have some idea that software should be tested. Indeed, the exponents of extreme programming (another of Kent Beck's innovations [see *www.armaties.com/extreme.htm* – Ed]) make testing a core part of their mechanism for software development. However, if you have ever watched the average programmer fixing a bug in a piece of code, you will realise that testing frequently leaves much to be desired.

Having located (hopefully correctly) the cause of a bug, many programmers make modifications, recompile the code, and run it to see if they have fixed the problem. If they have, they then go on with the next code fix, blithely unaware that they have not finished the current job. It is not enough to test that you have fixed the bug; you must also test that the rest of the code's behaviour has not been changed.

We need to make this final process easier and more accessible so that there is no excuse for the accidental introduction of changed behaviour during the process of developing or maintaining code. I want to suggest a couple of ideas that might make this easier. For the rest of this section I focus on C++ though most of the material will easily translate to Java and to other class-based languages.

Every class has a main

I suggest every class should have a member function called `main`. Well, you could call this special member function `test` but calling it `main` attracts me. Every program has a `main`, so why shouldn't every class have one? The purpose of `main` is to test the class's public behaviour. At the very least every public member function should be called at least once within `main` with the results written to a file. The last action of `main` should be to compare this newly written file with one that contains the designed output. Of course, the two should always match.

Originally, I used to teach programmers to write a test application for each class that they developed. It was the job of the test application's `main` function to test the class behaviour. But it is very easy to detach the test harness from the class. It is also easy to loose track of the latest version of your test harness, and so omit tests for later features.

Under my new scheme you still start with test harness, class definition, and class implementation files, but as soon as the class begins to settle down you move the test harness into the class as a static function. Initially, you have three files: `test.cpp` (which contains the test harness), `myclass.h` (which contains the class definition), and `myclass.cpp` (which contains the implementation).

In general, `test.cpp` will contain a single function called `main`. When ready, cut this from `test.cpp` and paste it into `myclass.cpp`. Then use your editor to prefix `main` with `myclass::`. Next, you add: `static int main();` into the public part of your definition of `myclass`. Finally, you add a very simple definition of `main` into `test.cpp`:

```
int main(){
   return myclass::main();
}
```

At this point I can just hear the code-efficiency fanatics complaining about the test harness being included in their production code. Of course, if you had a good quality linker (such as that which used to ship with TopSpeed C++) implementation code would only be included in the final executable if the function was called. As this was standard in that development environment almost a decade ago it is a sad reflection that it is still not the case for most other development environments. However, you can easily avoid fattening your code by using the preprocessor to conditionally include the implementation, if `TESTING` is defined.

This mechanism for testing involves one small irritation: either everything goes into `myclass::main` or you need other functions. For a class of any size, other functions are almost certainly needed. Then there is the issue that you really should be testing the protected and private member functions. Your `static main` in `myclass` scope has access to these, but any other functions you need for testing do not. This leads me to an extra level: provide an entire test class and make it a friend of the relevant class.

I guess some of you are going to object to seeing the following declaration in `myclass`: `friend class myclass_tester;` on the grounds that it provides an access route to your class's implementation details. True, but access constraints never have protected against deliberate cheating, only against accidental abuse.

This means I can outline a typical set of files for developing a class with proper testing. The file `myclass_tester.cpp` contains:

```
#define TESTING
#include "myclass.h"
int main (){
   return myclass::main();
}
```

Note that this is the complete content of this file. The only variant is the class name. A good editor should allow you to generate this file semi-automatically. The file `myclass.h` includes:

```
#ifndef MYCLASS_H
#define MYCLASS_H
class myclass {
public:
   static int main();
   friend class myclass_tester
// the rest of the class definition
};
#endif
```

The file `myclass.cpp` includes:

```
#include "myclass.h"
#ifdef TESTING
#include "myclass_tester.h"
int myclass::main(){
   // test routines
}
#endif
// rest of the implementation of myclass
```

The file `myclass_tester.h` includes:

```
#ifndef MYCLASS_TESTER_H
#define MYCLASS_TESTER_H
#include "myclass.h"
class myclass_tester {
```

```
// declarations of members
};
#endif
```

and finally, `myclass_tester.cpp` includes:

```
#include "myclass_tester.h"
// all implementation of
myclass_tester
```

This means we have five files where the traditionalists make do with one and the modernists use two. However, what we now have is definition, implementation, and testing packaged in a standard form.

The only warning that I would give is that testing the correct behaviour of code is not the same as testing that an application delivers on its specifications. The former should be done as part of the process of code writing. The latter should be the task of a distinct group tasked with testing deliverables.

I have outlined my approach without using namespaces in order to keep the code as simple as possible. In reality, I would always create classes encapsulated in namespaces. A group of related classes will all be in the same namespace, but unrelated ones will not be. One advantage of namespaces is that you can use them to encapsulate many features that have previously been placed in classes. For example, early coding guidelines suggest that `enum`s should always be defined in a class scope in order to avoid polluting the global namespace. This was an excellent precept, but I think that moving them out to an encapsulating namespace is better. Among other things, it allows users of your code to strip off the scope qualifiers if they wish through `using` declarations.

I am wondering whether static members should not also be moved out of the class into the enclosing namespace. I think this depends on whether the static member requires privileged access and whether it needs to be inheritable. Certainly a strong case can be made for moving some private static members into the enclosing namespace in the implementation file – do not clutter your class interface with non-essentials.

A must-read book

A copy of *The Practice of Programming* by Brian Kernighan and Rob Pike (ISBN 0-201-61586-X) landed on my desk a few weeks ago. It is exactly what you would expect from two such experienced programmers. It is full of good sense distilled into highly readable text. Of course there are a few places where I would disagree with them and sometimes I wonder if they have actually read all the titles in their end of chapter supplementary reading lists. In other words, you should read the book thoughtfully and avoid simply assuming that every book in a suggested reading list will be of equal quality. Having said that, I think readers of this column should make an effort to read this book. It would be a rare programmer who gained nothing from such an effort.

Conferences

During the last couple of months I have attended two excellent conferences. They were very different in style and in content but both deserve to be added to your list of events worth attending in future. DevWeek continues to be an excellent conventionally organised conference. It is multi-track – covering C++, Java, Delphi, and VB. I remain of the opinion that running a straight C++ track in parallel with one essentially about C++ based components is an error, but only because there is a considerable overlap of interest between the two.

If you want presentations (well, really large group tutorials) from world class experts, this event delivers.

The second event was OT99 (it used to be called Object Technology). This is a completely different style of event centred on a largely residential base of attendance. This is important because the result is that the organisers provide a complete programme for attendees right through to Birds of a Feather meetings between 10.30 and 11.30 in the evening. The daytime items ranged through case studies, tutorials, keynotes, goldfish bowls, and workshops. One item was even based on a play. Not only is the mix impressive, the result is impressive too. You do not always learn what you expect to, but I never came out of a session without having learnt something. The moderators, presenters, etc, were all good and most were very good or excellent.

The evening programme ranged through such interests as writing Haiku, Japanese gardening, and drumming. In other words, the event addresses the whole person, not just the programmer in you.

Though relatively expensive, you get excellent value for your money. If you are a serious software developer, this event should be on your list of ones to try to attend.

Last month's problem

Examine the following piece of pure C code and determine why it should not compile without a diagnostic. When you have done so, decide whether the equivalent code in C++ (using `iostream` objects) should compile and execute correctly.

```
#include <stdio.h>
#include <ctype.h>
struct char10 { char data[10]; };
typedef struct char10 mystring;
mystring uppercase(mystring item){
   int i;
   for (i=0; i<10; ++i) item.data[i] = toupper(item.data[i]);
   return item;
}
int main(){
   mystring hello = {"Hello"};
   printf("%s World", hello.data);
   return 0;
};
```

In the hurry to finalise last month's column I left out the intended problem but (fortunately) introduced another completely silly one. Let me deal with the latter before allowing you to focus on the intended problem. A certain carelessness in writing the C++ grammar means that the language should diagnose null statements at global or namespace scope. I have no idea if this applies to C as well, and even less whether it will apply to the soon to be finalised (well, possibly) new standard for C.

A strict C++ compiler should issue a diagnostic for that semicolon after the closing brace of `main`. I have no problem with a compiler issuing a warning for an empty statement at any scope, but I think most of us would be unhappy if it proceeded to refuse to generate object code.

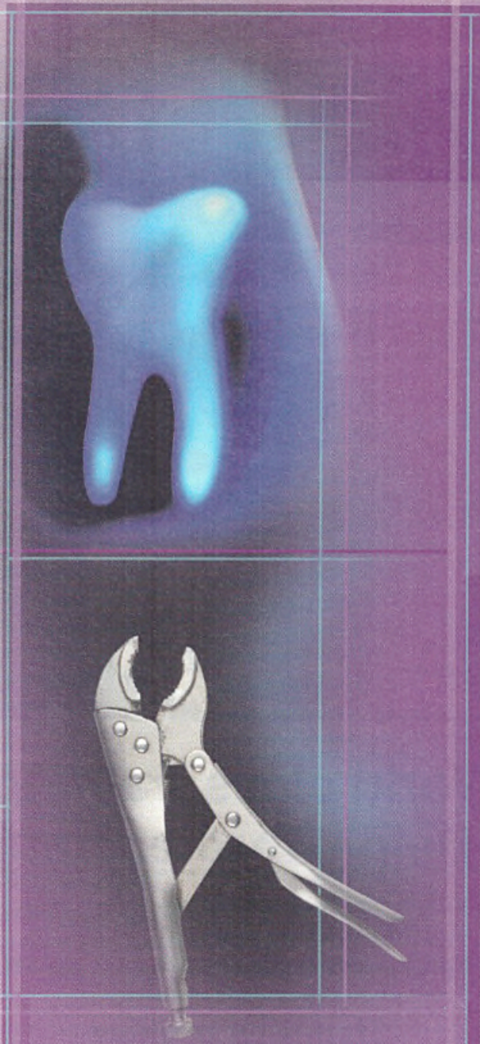
This month's problem

Consider this version of `main` (which is the one my compiler saw):

```
int main(){
   mystring hello = {"Hello"};
   printf("%s World", uppercase(hello).data);
   return 0;
}
```

Association of C/C++ Users subscriptions: individual £15, student £7.50, corporate £80, Overload & C++ SIG £30 (including ACCU membership). For further information and application forms write to Francis Glassborow, 64 Southfield Road, Oxford, OX4 1PA, ring 01865 246490, or email chair@accu.org.

WOULD YOU PULL YOUR OWN TEETH?



Then why write your own security code?

For developers, adding software security code to your product can be a source of acute pain. Rainbow's easily deployed license management solutions provide world-class software protection that won't hurt a bit. The SentinelSuperPro™ delivers low profile, low cost security. SentinelLM™ harnesses the power of the Internet to provide remote electronic license control. Pain-free protection, secure distribution, 100% user registration and more from the software protection industry leader—Rainbow.

http://europe.rainbow.com/uk

Developers — contact us to see if you qualify for a free developer's kit!

RAINBOW TECHNOLOGIES

Changing the way the world secures business.

Sentinel House, 4 The Forum, Hanworth Lane, CHERTSEY, KT16 9JX
Tel: 0800 579200 Fax: 01932 579243

Let docking commence

Mark Smith shows how you can use docking windows within your Delphi applications, while avoiding some annoyances demonstrated by the IDE.

One of the most visible differences between version three of Delphi and version four is the use of docking windows in the integrated development environment. While it might have seemed like a really good idea to the Delphi development team, most programmers have found the docking interface a mixed blessing. The Delphi IDE often fails to detect the difference between attempts to stack windows using tabs or to display them side-by-side. Sometimes it positively refuses to do one or the other of these operations and you end up dragging windows around wildly while the IDE draws docking rectangles around all of the other forms, visible or not. This month I'll look at how you can use docking windows within your own applications, while avoiding some of the more obvious annoyances demonstrated by the Delphi IDE. I'll also look at the different options for docking and floating toolbars.

The basics

With only one example program and less than three pages of printed documentation, it is no surprise developers haven't rushed to adopt docking into their applications. Table 1 shows the main classes, properties, methods, events, and messages used to provide docking facilities, built by trawling the VCL. As you can see, docking has been added to the VCL in `TControl` and is therefore available to anything you can put on a form, as well as the form itself. The `TControl` descendants can be docked to other controls while `TWinControl` descendants can accept other controls docked to them. For `TControl` the two most basic properties are `DragKind` and `DragMode`. If a control has a `DragKind` of `dkDrag` (the default), it means the control can take part in drag and drop operations. However, if `DragKind` is `dkDock`, the control takes part in docking instead. You need to set `DragMode` to `dmAutomatic` to get the docking operation to begin without writing code, but if you leave `DragMode` set to `dmManual` you can begin a dock operation in code by calling `BeginDrag` in a mouse down event. The `TWinControl` descendants have a Boolean `DockSite` property, which if `true` means that a control can accept other controls docked to it. You can choose to accept a control or not by adding handlers to the `DockOver` and `GetSiteInfo` events, setting `Accept` to `false` if you choose not to accept a control.

Many classes have extended support for docking operations. The `TControlBar` class automatically creates a wrapper for controls docked to it, while `TPanel` has special support for docking, acting to tile the controls docked onto it. The `TPageControl` automatically creates a new `TabSheet` for every control you drop on it, but does not always label the new tab. To overcome this, you need to add a handler for the page controls' `DockDrop` event.

A sample application

The sample docking application, DockDemo (see EXE OnLine), shows how to customise docking in the following areas: beginning a docking operation, controlling the appearance of a control while it is floating, and ending a docking operation. The main form contains several controls, all dockable, though some have `DragMode` set to `dmManual`, requiring the application to begin the docking operation in the `MouseDown` event for the control. The event handler begins a drag operation if the user holds down the Alt key to prevent the dragging/docking behaviour from interfering with normal use of the mouse. The manual approach gives some degree of customisability over how far the mouse has to move before a dragging or docking operation will begin by setting the drag threshold.

While controls can be described as floating, they never form windows in their own right – Delphi creates a new window for the control to sit in while it is floating. This window is called the floating dock site, and the default floating dock site is a form of type `TCustomDockForm`. Unfortunately, `TCustomDockForm` has a few undesirable features that would make it look odd in a real application. It does not support having more than one control docked to it, which would prevent the most sensible use for docked controls. It does not allow one to have other controls placed on it that are not docked, such as a panel for holding buttons, and it sets its caption to the text in the control, which is often empty or meaningless. Fortunately, it is very easy to replace the default form with one of your own creation.

For the demonstration application the form `TFloatingDockSiteClassForm`, in module `frmFloatingDockSite.pas`, takes steps to overcome the problems listed above, and is shown in action in Figure 1. In the demo app, we assign a new floating dock site to all of the controls with which we wish to use the dock site. When one of these controls is floating, it manufactures one of our forms unless it is about to be docked onto an existing floating dock site, which can accept many controls. You can choose whether the docked controls are tiled onto a panel or presented as tabs underneath a toolbar. Finally, the form has a go at producing a reasonable caption for each of the controls docked to it.

You can go a lot further in customising how docking works. Classes descended from `TWinControl` have a property `DockManager` that determines how a control handles docking. If you do not provide a new value for `DockManager`, Delphi will use the default one, `TDockTree`. This class manages a tree structure of dock zones (regions into which other zones or controls are docked). Looking at the code for `TDockTree` and `TDockZone` shows these are fairly complex classes. If you want to customise the way docking works, it's probably best to leave the default dock manager in place and modify the way controls respond to docking instead.

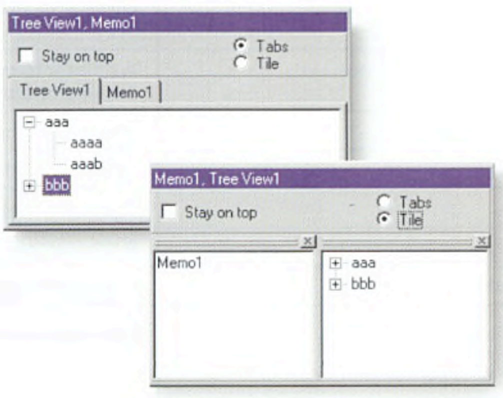


Figure 1 – The demonstration floating dock site form in action.

Toolbars

There seems to be two schools of thought regarding how toolbars should look and behave: the Internet Explorer style, where you can move toolbars around at the top of your form, or the Office 97 style, where toolbars can be dragged away from the application window completely. The toolbar project in this month's sample application shows both sorts of toolbar. The form captioned 'Cool Bar Form' is built using the standard windows `TCoolBar`, while the 'Control Bar Form' is built with the Delphi `TControlBar` control acting as the host for the two toolbars.

First, let's look at the `CoolBar` solution. When you run the application, you get two toolbars labelled 'Text' and 'File', each with a single sizing/positioning bar on the left. The neat feature of cool bands is the way that you can stack many toolbars onto a single row and then dismiss them by clicking on the line on the left of the toolbar. This is the way toolbars work in Internet Explorer 4. You have the advantage of being able to specify what the user needs to do to call up and dismiss toolbars using the `BandMaximize` property, which can be set to click, double-click, or not enabled. The text label shown at the left of each toolbar is actually the name of the `CoolBand` that holds it. Dragging a toolbar off the `CoolBar` results in the destruction of the `CoolBand` that hosted the toolbar, and by default `CoolBand`s do not show a caption. You need to do a little work to get the name of the band to re-appear when you re-dock the toolbar to the `CoolBand`, as demonstrated in the demo application.

The `TControlBar` controls offer the more traditional look of the MS Office suite of applications, with toolbars that do not overlap and that can be dragged off the palette and moved to wherever is most convenient. You select, move, and drag a toolbar around using the two bars to the left of the toolbar. Like `TCoolBar`, the `TControlBar` uses a list of internal objects (in this case a private list of records of type `TDockPos`) to manage the collection of docked objects. You don't have a direct interface to the collection, but you can manipulate it using the `OnBandDrag`, `OnBandInfo`, `OnBandMove`, and `OnBandPaint` events, which give you control over how a band is positioned and presented.

By default, a toolbar component has a border width of zero, which means you can grab it only where there is a control that does not accept focus, such as a separator. Setting a `BorderWidth` of `1` makes it possible to click on the edge of the control to initiate docking, as shown on the File toolbar of the form captioned 'Control Bar Form'. You can drag the toolbars from this form and drop them onto the other form, but the file toolbar will be easier to remove. Another property you will want to use is `Autosize` to remove any excess space around the contents of the toolbar while it is floating.

Let docking commence

Overall, I'm surprised at the amount of work Borland did to add docking to the VCL. I hope this article has encouraged you to try and find a use for it in your applications. ■

Mark Smith is a Delphi contractor, currently working for an investment company based in Westminster. You can contact him by emailing msmitha@cix.co.uk, or say hello at a Borland User's Group meeting. Telephone 01980 630032 for BUG details.

Item	Purpose
`TControl DockOrientation`	Used to specify the orientation of controls in a docking zone. Values are `doVertical`, `doHorizontal`, or `doNoOrient`.
`TControl DragKind`	Whether a control takes part in drag/drop or drag/dock operations.
`TControl FloatingDockSiteClass`	Specifies the type of class used to hold a control when it is floating.
`TControl ManualDock, ManualFloat, Floating`	Use these functions to make a control dock to another one, to undock the control, or to tell if the control is floating already.
`TControl HostDockSite`	The control that is the dock site for the current control.
`TControl TBDockHeight, LRDockWidth`	Height and width of the control last time it was docked.
`TControl UnDockHeight, UnDockWidth`	Height and width of the control last time it was floating.
`TWinControl DockSite`	If true, then the control can have other controls docked to it.
`TWinControl DockManager (protected), UseDockManager`	An object of type `IDockManager` used to control how docking is handled. If `UseDockManager` is true, a new instance of `TDockTree` is created to act as the dock manager.
`TWinControl OnDockDrop, OnDockOver, OnDockOver, OnUnDock, OnGetSiteInfo`	Events called when you try to dock a control to one that is acting as a dock site.
`TWinControl DockClients, DockClientCount`	Gives access to the controls docked to this one.
`TForm DockManager (public)`	A pointer to an object of type `IDockManager` responsible for managing the dynamics of a docking operation.
`TCustomDockForm`	The default implementation of a form used to hold floating controls.
`IDockManager`	Defines an interface for managing docking operations.
`TDockTree`	The class that provides the docking management, as specified in the `IDockManager` interface.
`TDockZone`	A `TDockTree` contains a linked list of `TDockZone` objects that represent the various dock zones.
`TDragDockObject`	Intermediate object that holds information for the `TWinControl OnDock` operations.
`CM_CONTROLLISTCHANGE, CM_DOCKNOTIFICATION, CM_UNDOCKCLIENT, CM_VISIBLECHANGED`	Messages sent to a `TCustomDockForm` object in response to docking actions.

Table 1 – Classes, properties, and methods involved in docking.


As one of the country's leading independent Microsoft training companies we choose our 200 lecturers carefully. Not for their pretty faces, but for their real-world experience, business knowledge and technical experience. That's because at QA Training we offer the best training for doing the job, in addition to getting the certificate.
In fact, over 5,000 Microsoft solution developers can testify to the success of the unique QA Training approach in the past year alone. With the widest choice of Microsoft courses available, using our unique certification packages and flexible SkillsCard options, you won't be pulling your hair out to get the course you want, when you want.
QA Training.
Certifiably the best for MCSD.
MCSD
Certification
ARE YOU CERTIFIABLE?
TAKE QA TRAINING'S NEW MCSD PROGRAMME AND YOU SOON WILL BE
TO FIND OUT MORE INFORMATION CALL:
01285 883388
OR SEND YOUR BUSINESS CARD TO: QA TRAINING,
CECILY HILL CASTLE, CIRENCESTER, GLOUCESTERSHIRE GL7 2EF
FAX: 01285 883399 OR EMAIL: mcsd@qatraining.com
OR VISIT OUR WEB SITE AT: http://www.qatraining.com
Part of the QA Group
Microsoft Certified Technical Education Centre
Microsoft Certified Solution Provider Partner
QA Training
© 1999 QA Training acknowledges that all product or service names mentioned herein are trademarks of their respective owners.

The state-of-art of XML

Tom Guinther continues his investigation of XML by presenting an example client/server system based upon real world usage.

Last month's overview of XML focused on the basic syntax and a subset of the technologies surrounding XML, including a general discussion of the XML parser/processor technology and the APIs used to traverse the XML document structure. This month I'll talk more about the HTML/XML Document Object Model (DOM) and what you can expect to find in the variety of available XML technologies.

XML technology choices

On a recent project one of my product-development teams began looking for effective ways to use XML technology to create a fast, robust, and easy to deploy client/server architecture. Along the way we learned a lot about the current state-of-the-art of XML: what you can and can't reasonably do at this point in time. When we started the project Sun had not officially released Project X so it wasn't a viable option although it was considered. Instead we looked to take advantage of the built-in XML support in Microsoft Internet Explorer version 4.0 and/or version 5.0. What we would get from using the Microsoft technology was easy deployment (the end-user didn't need to download an XML command processor), a high degree of quality, and the price was right (ie it was free).

What we found was that IE 5.0 provided an advanced set of XML technologies, including an XSL processor. IE 5.0 would also allow the server to embed XML within HTML templates (XML data islands) – the client processes the data via the browser's object model using JavaScript. For the actual presentation of the XML data we would use XSL (XML stylesheet language) templates to automatically generate the major UI components such as graphs and tables. XSL templates provide a clear separation between the format of the XML data and how it is displayed. We could dynamically alter the look-and-feel of the client simply by changing the underlying stylesheets. Really, it is a great solution; there is no fuss over choosing and deploying XML technology, the XML processor is fully DOM compliant, web developer tasks are simplified by using web scripting languages instead of Java, and it is the only viable XSL solution we could find at the time.

Unfortunately, it was a solution we did not feel comfortable choosing because IE 5.0 was three months away from being released and all of our users had IE 4.0. Because this product was being developed in a large corporate environment it was unlikely we could convince the system integrators to upgrade our large client base. That would be like trying to roll a sleeping 800 lb gorilla to the other side of the bed. It just wasn't going to happen.

Instead, we were faced with two basic options, neither of which was the preferred solution. The first was to deploy the IE 5.0 XML processor with our application. As it turns out, our tests show this is feasible but we were worried about side effects it might cause with other applications since the IE 4.0 and 5.0 technologies are not compatible. This left only one option: use the XML services provided by IE 4.0. What we lost in the process was XSL (not supported by IE 4.0), and we had to work with an XML processor that was highly correlated with the XML DOM, but not source code compatible. We also couldn't use XML data islands, which meant major aspects of our IE 5.0 architecture would need to be completely redesigned. One final note: it is possible to move a majority of the processing to the server side where technology choices have little or no impact on the end-user. The downside to this is that the client application becomes much less intelligent and a disproportionate amount of the processing load is shifted to the application server, which could result in scalability problems.

The XML client/server architecture

Figure 1 illustrates the overall architecture of the system we were dealing with. For historical reasons performance was a key issue. This means most of the information in the backend database is ready for immediate presentation and the SQL necessary to extract information from the database (via JDBC) is so simple it practically defies optimisation. The application server is a Java console application invoked by the CGI command processor. In the production environment it is actually a Windows NT system service (using Microsoft technology available in the Java SDK 3.1), which provides higher reliability and flexibility for system operators.

One thing that may surprise you is that the application server doesn't use any of the XML implementations that I have mentioned previously. The reason for this is that the server's implementation of XML is so straightforward that it wasn't necessary to use anything all that sophisticated. The server's primary responsibility regarding XML is to generate XML for the client to process. Because our XML format is kept conceptually simple (never more than two nested levels) writing the XML data was as simple as opening an `OutputStream` and writing to it. I should mention that we did try to use the IE 4.0 technology on the server side (for writing XML documents) but got strange results.

On the other hand, the client needs to process the data from the application server so it should use some type of XML processor technology to make life easy. In the general scenario, the client requests a particular HTML page from the server. The server responds by grabbing an HTML template for that particular page. The template contains all the commands and logic required to both request and present the details of the page. The client is responsible for providing the individual parameters necessary to drive the data gathering. For example, the client is

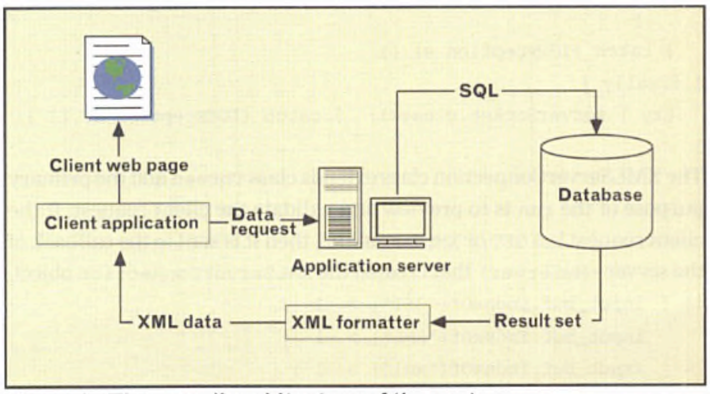


Figure 1 – The overall architecture of the system.

Oracle 8i includes a JVM so that you can write stored procedures in Java, as well as have the ability to get results in XML format.

producing web reports so it needs to provide an 'Account ID', an 'As Of Date', and the period type of report (monthly or year-to-date.) These parameters are provided to the client software via an easy to use dialog box. The code behind the dialog box also uses XML to generate a complete list of available accounts and the corresponding dates available for the accounts. This level of intelligence makes the client software very easy to use and prevents many of the mistakes end-users typically make.

The application in detail

The system goes like this. The client application is an HTML page that contains a frameset to display reports dynamically. The frameset allows the main HTML page to be persistent so that context information is always available. The client application requests the report templates (they are HTML) from the application server and loads them into the frame. The report templates contain JavaScript, which is executed to access the parameter data from the context and make further data requests to the application server. This monitors a 'well-known' port for requests from the client. The client requests are formatted very similar to a CGI request. Finally, the application server parses the client request and fires events to different Java-based command processors to handle the request.

The following code fragment shows the primary thread loop for the application server. It creates the socket connection and waits for a connection, where each connection is handled by a new instance of an `XMLServerConnection` object that handles the client requests.

```
public void run() {
   ServerSocket serverSocket;

   try  {
      serverSocket = new ServerSocket(mPort, 2000);
      mStatus = true;

      while (mStatus) {
         Socket socket = serverSocket.accept();
         new XMLServerConnection(this, socket);
      }
   } catch (IOException e) {}
} finally {
   try { serverSocket.close();  } catch (IOException e) {} }
}
```

The XMLServerConnection class extends class `Thread` and the primary purpose of the `run` is to preview and validate the client request. If the client request is a `GET` or `XML` command, then it is sent to the callback of the server `(MyServer)` that created the `XMLServerConnection` object.

```
if ( input_buf.indexOf("GET") > -1 ||
     input_buf.indexOf("XML") > -1 ||
     input_buf.indexOf("xml") > -1 )
{
     s_ret = MyServer.process(input_buf);
}
```

The base class `XMLServer` further validates the client request and if it is syntactically correct (not including parameter validation) then the request is sent to the `processXML` method. If the request is not well formed, an HTML error message is sent back to the client. In order to do any useful work a class must be derived from `XMLServer` that overrides the `processXML` method.

In general, the derived class further parses the request, extracting any parameters passed and storing them in a `Dictionary` object using well known keys. When the proper method for the request is executed the parameters do not need to be passed via the method. Instead, each method extracts the parameters from the `Dictionary` and performs any necessary validation. This allows a generic parameter parser to be used and the methods are not parameter type dependent. The following code fragment shows the basics of this scheme for a very simple request, `xml_GetFactors`.

```
// core code of parameter parser
while (TokenGenerator.hasMoreTokens()) {
   String operand1, operand2;
   operand1 = TokenGenerator.nextToken();
   if (TokenGenerator.hasMoreTokens()) {
      operand2 = TokenGenerator.nextToken();
   } else {   break; }
   Dic.put(operand1.toUpperCase(),
            HtmlEncoder.decode(operand2));
}

// typical command execution, pulling parameters from
// a dictionary
if (command.equalsIgnoreCase("xml_GetFactors")) {
   String  account_code    = (String) Dic.get("ACCOUNT_CODE");
   String  benchmark_code = (String)
                                  Dic.get("BENCHMARK_CODE");
   String  start_date      = (String) Dic.get("START_DATE");
   String  end_date        = (String) Dic.get("END_DATE");
   return  GetFactors(account_code, benchmark_code,
                        start_date, end_date);
}
```

Most requests in this particular application server end up as SQL requests via JDBC to an Oracle database. JDBC returns a result set that includes the requested data as well as the metadata that describes it. Using the metadata, we are able to determine the name and type of each column in the result set. Using this information the application server is able to convert the results of the query into XML for processing by the client. It should be noted that Oracle version 8i includes a JVM so that you can write stored procedures in Java, as well as have the ability to get results in XML format. Unfortunately for us, we had to use version 7.33 and were unable to take advantage of these advanced features.

The new Java

XML is a great technology and should help change how we approach traditional application development, and of course, web development. The current state of XML technology has a lot of trade-offs but it improves everyday. Soon, the use of XML will be as widespread and as commonplace as Java is now.

Tom Guinther is a freelance software developer specialising in systems software and internals. He can be reached at tomg@nh.ultranet.com.


SOFFRONT ®
Enterprise Solutions
DEFECT TRACKING
WEB-ENABLED HELP DESK
SALES AUTOMATION
KNOWLEDGE MANAGEMENT
Defects submitted and closed over time
100% SATISFACTION GUARANTEED
"The ability to get users up and running with TRACK in a short period of time is invaluable."
Greg Bryant- Principal Software Engineer, NEC Technologies, Inc.
"It has also improved our customer because we have a more accurate way of tracking customer concerns."
Ed Mischkot- Software Engineering Mgr. Atlantis Aerospace Corp.
"TRACK automates interactions with related tools."
Peter Coffee- PC Week, March 4, 1996
"Wins hands down in power, flexibility and ease-of-use."
Data Management Review, January 1996
"You can create custom screens and views to tailor TRACK to your environment."
Micheal Deigman- Windows NT,October 1998
Soffront Software, Inc.
Phone: 800-763-3766
Outside USA: 408-263-2703
Fax: 408-263-7452
E-Mail: info@soffront.com
WWW.SOFFRONT.COM

<HELP> VB 6.0 </HELP>

Visual Basic 6.0 provides direct support for HTML-based help files. Jon Perkins shows how you can construct them using the tools that come with Visual Studio.

Writing help files for applications can be a rather fiddly affair if you choose to use the standard Microsoft Help Workshop in preference to one of the third-party tools. The original help compiler was written to understand RTF (Rich Text Format) files. This means that the authoring process can involve writing your help file source in Microsoft Word, adding a curious combination of footnote characters, and then saving the file in RTF format in order to achieve the desired result.

As a natural successor to this technology Microsoft has developed a new help system that is centred around the HTML format, which of course means that you can use your favourite HTML editor and save the result in native format. This month's column complements the article by Dave Jewell (*Help me in HTML*, EXE, October 1998) in which he reviewed the relative merits of three third-party HTML Help development tools (RoboHELP Office, Doc-To-Help, and HDK). However, I'm discussing the creation of HTML-based help files via the Microsoft HTML Help Authoring Kit that ships along with Visual Studio, and which is available for free from the Microsoft website.

At first glance it's not immediately obvious that Visual Basic 6.0 directly supports HTML-based help files. In order to associate a help file with a project it is necessary to select the Project Properties dialog and then raise the Browse dialog next to the Help File Name field. The resulting common dialog displays, by default, a file type of 'WinHelp Files (*.hlp)'. However, opening the file type drop-down list box also displays 'HtmlHelp Files (*.chm)'. Once a suitable file has been identified, associating the `HelpContextID` value with each form in a project works in exactly the same way.

In order to install the HTML Help Workshop from the Visual Studio disks you will need to run the installation program from the *HtmlHelp* directory on disk 1. Since this is theoretically quite an important part of the overall development studio, it is curious that it isn't included in the main Visual Studio setup program. The kit supplied on the CD is version 1.1, but the Microsoft website offers version 1.2 for free download.

Sadly, this newer version doesn't seem any more robust than its predecessor; while it is quite usable it does have a tendency to forget to repaint parts of its main form from time to time. And most of the dialogs have What's This? help buttons, but very few controls actually have an associated help topic. Oh, and it crashed on me the other day after I'd typed in – and before I'd saved – a whole load of entries into the Table of Contents definition area. Having said all this, it is stable enough to develop with, and any prolonged use will soon condition you as to where the specific foibles are, so you'll quickly find yourself working around them as a matter of course. Sad, but true.

Creating a help file

In order to create a new HTML-based help project a rudimentary wizard will ask you a few basic questions before providing you with an empty project file. As with any system, you really need to plan your layout before you start authoring. The simplest way to get results is to have one HTML file for each topic. These files can be created in any HTML editor in the same way that you would create a web page. Authors of the old-style help files will be pleased to realise that you can just insert image files directly onto the design page, so there's no more messing about with `{bmc}` macros.

As you create each HTML file, you need to introduce it into the `[FILES]` section of the project. In the Workshop environment make sure the Project tab is selected and press the Add/Remove Topic Files button. This will display a standard File Open dialog, which allows you to build up the list of files quite quickly, particularly because it allows you to select multiple files in one go. A couple of other tabs allow you to create a Table of Contents page and an Index page.

In order to make the help topics available for API calls you need to provide a bit of lookup information, entered into the project file manually or included as C-style header (`.h`) files. One file should provide a list that maps the *symbolic ID* of a help topic to a *topic ID*, for which I've shown an example in Listing 1. A second file provides an alias lookup for each of these symbolic IDs to the actual source files that they represent, this being illustrated in Listing 2. Note the use of the `IDH_` prefix in front of the symbolic IDs. If this prefix is used, then the workshop will perform a cross-reference check between the topic IDs in the compiled help file with those mapped to numeric values in the project file. It will warn you of any inconsistencies found.

```
#define IDH_Home 2000
#define IDH_Topic1 2001
#define IDH_Topic2 2002
```

Listing 1 – Mapping a reference to an ordinal number.

```
IDH_Home=Home page.htm
IDH_Topic1=Topic 1.htm
IDH_Topic2=Topic 2.htm
```

Listing 2 – Mapping the reference to the source file.

```
.topic 4000
Applies changes to database

.topic 4001
Cancels changes
```

Listing 3 – Providing the text for What's This? help.

As an aside, one useful feature that the Workshop does provide is a `Decompile` feature. This allows you to convert an existing `CHM` file back into its constituent components, so you can grab your favourite resources from other help files and use them in your own... no, forget I said that!

Programmatic access

The software developer gains direct programmatic access into the HtmlHelp system through the `HHCTRL.OCX` file. From its name one would expect to set a reference to it through the Components dialog under the Project menu. Not so! The file should be treated as a standard DLL, for which you make a function declaration as follows:

```
Declare Function HtmlHelp Lib "hhctrl.ocx"
Alias "HtmlHelpA" (ByVal hwndCaller As Long, _
                   ByVal pszFile As String, _
                   ByVal uCommand As HH_COMMAND, _
                   dwData As Any) As Long
```

where

- `hwndCaller` is the handle of the calling window, or Null
- `pszFile` is the `CHM` file to call upon
- `uCommand` is the action to perform (note that `HH_COMMAND` is an enumerated type list that provides the added benefit of allowing the Visual Basic editor to display a popup list of possible values as you type the function)
- and `dwData` is additional data, the nature of which depends upon the value of `uCommand`

In order to invoke a help file, with the context of a Table of Contents view, you would make a call as follows:

```
Const HH_DISPLAY_TOC = &H1
Dim lReply AS Long _
lReply = HtmlHelp(Me.hWnd, _
                  App.HelpFile, _
                  HH_DISPLAY_TOC, _
                  ByVal 0&)
```

These direct calls are required only for menu or command button invocations, otherwise you can just set the `HelpContextID` value for each form to match the value of the topic ID declared for each help file. It feels a little strange having such a new technology as this being presented to the Visual Basic developer as a DLL-style call, but of course this will change. Version 2 of HTML Help will be COM-based.

Context-sensitive help

What's This? functionality is a very useful form of help, but sadly tends to get missed out of many developments, particularly those performed in-house (as opposed to commercially available products, which seem to fare a little better). To create a set of text popups you need to add the values to a separate text file and then add it into the help project. The format of the entries for this text file is:

```
.topic topic_id
topic_text
```

as illustrated in Listing 3. This file should then be added into the help file project definition via the Text Popups tab of the HtmlHelp API Information dialog. To then tie this in to a Visual Basic form you need to set both the `KeyPreview` and `WhatsThisHelp` property values to `True`, and then set an appropriate `WhatsThisHelpID` value for each control. Setting the `WhatsThisButton` value to `True` then displays the What's This? question mark button on the form caption bar, but only if you've set the border style as either `Fixed Single` or `Fixed Dialog`.

There are a couple of issues to be aware of when you are programming for a What's This? implementation. The first is that a form with the `WhatsThisHelp` property set to `True` will take F1 key presses as invocations of the same kind of help. If you want the user also to have access to the full help file, then you need to find another way to offer it other than the F1 key. I would suggest that in this case either menu access or a dedicated Help command button would be the most appropriate way. The second issue is that you need to provide slightly different definitions of the application help file depending upon whether you are providing What's This? help or normal, F1-style help. To define the help file for the normal F1 kind of help the definition is along the lines of:

```
App.HelpFile = "Example.chm"
```

However, to define the help file for use with a What's This? implementation you need to specify the source file that contains the What's This? source text:

```
App.HelpFile = "Example.chm::/popup.txt"
```

If you attempt to leave this second format in place while attempting the normal F1 help, then the help file will open but an error message will be displayed.

Alternatively, trying to activate What's This? help with the help file set in the first format will merely display a message box that states 'Cannot open the file Example.chm'. Therefore, if you are designing an application that will need to have a mix of the two different kinds of help, I would suggest that you include the relevant `App.HelpFile` declaration within the `Form_Activate` method of each form.

Paper format

Microsoft Press publishes a book called *The Official Microsoft HTML Help Authoring Kit* (by Steve Wexler of WexTech Systems, ISBN 1-57231-603-9, £37.49). The content is well presented and it would certainly be of use to folk who need to write HTML-based help files that go beyond the basics. Somebody with previous experience of the Help Workshop would probably pick up the mechanics of the new HTML Help Workshop fairly quickly and so would be unlikely to need it for basic help files. However, it does show you what else can be achieved, such as using the 'Training Card' mode, which allows for a greater amount of interaction between an application and the help file.

The author takes care to present sample code for both the Visual C++ and the Visual Basic camps, but there is no getting away from the fact that the HTML Help programming interface is really written with the C++ programmer in mind. One word of warning about this book: the Visual Basic content relates to version 5, which didn't natively support HTML Help. Therefore, there is a discussion concerning the implementation of What's This? help through window subclassing, which is obsolete. At the time of writing, I could find no indication of any planned updates to *The Official Microsoft HTML Help Authoring Kit*.

These points aside, I would still recommend the book to anybody who intends to get serious about HTML Help development. For that matter though, I would also seriously recommend that such a person take a good look at the products reviewed by Dave Jewell a few months back. ■

Jon Perkins is a freelance Visual Basic developer and a Microsoft Certified Solution Developer. He is a contributing author of Advanced Microsoft Visual Basic 6.0 by The Mandelbrot Set, published by Microsoft Press. Contact him at http://www.jonperkins.com.

EXE DIRECTORY

Please call Sarah Horsley on 0171 970 6545 for details

ACTIVEX / JAVA COMPONENTS

ProtoView UK Ltd
26 Offington Drive, Worthing
West Sussex BN14 9PN
Tel:01903 538058 Fax:01903 538068
more.info@protoview.co.uk
www.protoview.co.uk
ProtoView UK offers a large selection of quality feature rich ActiveX and JavaBean components.

APPLICATION INTEGRATION

Frontec AMT UK
Unit 2, Wallbrook Business Centre
Green La, Hounslow, Middx TW4 6NW
Tel: 0181 814 4700 Fax: 0181 570 6760
info@frontec-uk.com
www.frontec.com
Cross platform intelligent messaging solutions for application integration and e-commerce. Full capabilities for the World Wide Web, EDI, SAP R/3, RDBMS, IBM MQ Series, VANs.

DEVELOPMENT TOOLS

Inprise
8 Pavilions, Ruscombe Business Pk
Twyford, Berks RG10 9NN
Tel: 0118 932 0022 Fax: 0118 932 0017
Customer Services: 0800 454065
www.inprise.co.uk
Inprise is the leading provider of development solutions from desktop to enterprise level. Inprise's award-winning tools are supported by a network of VARs, partners and resellers in the UK.

ComponentSource
30 Greyfriars Road
Reading, Berks RG1 1PE
Tel: +44 (0)118 958 1111
Fax: +44 (0)118 958 9999
sales@componentsource.com
info@componentsource.com
www.componentsource.com
Over 1200 components can be unlocked instantly 24 hours a day, via the Internet, the Website, a private dial-up network or the telephone.

CYRANO (UK) Limited
Devon House, Park Street
Slough, Berkshire SL1 1PX
Tel: 01753 516500 Fax: 01753 516441
khall@cyrano.com
www.cyrano.com
Leading developer of automated software quality tools for financial, telecommunications, government, utilities and other markets. More than 1800 customers worldwide.

GigaSoft Inc
696 Lantana, Keller, TX 76248, USA
Tel: 001 817 431 8470
Fax: 001 817 431 9860
Email: info@gigasoft.com
www.gigasoft.com
Graphing/Charting functionality for all versions of VB, Delphi, VC Builder, and others. Download a demo or evaluation edition at our website.

Intasoft Ltd
Tresco House, 153 Sweetbrier Lane
Exeter EX1 3DG
Tel: 01392 217670 Fax: 01392 437877
sales@intasoft.co.uk
Intasoft is the supplier of the popular configuration management tool AllChange. Our product is backed by high quality support, training and consultancy services.

Popkin Software & Systems
St Albans House, Portland St
Leamington, Warks, CV32 5EZ
Tel: 01926 450858 Fax: 01926 311833
sales@popkin.com
www.popkin.com
Popkin Software & Systems is the manufacturer and vendor of System Architect, the market leading suite of Analysis & Design Modelling tools.

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

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

SELECT Software Tools plc
Westmoreland House, 80-86 Bath Rd
Cheltenham Glos GL53 7JT
Tel: 01242 229700 Fax: 01242 229701
heidik@selectst.com
www.selectst.com
SELECT Software Tools provides a business-driven solution to companies adopting component-based development. Through its market-leading products and services, SELECT enables the modeling, management and assembly of enterprise-class applications from components.

Stingray Software, Inc
9001 Aerial Center, Suite 110
Morrisville, NC 27560 USA
Tel:001 919 461 0672/001 800 924 4223
Fax: 001 919 461 9811
sales@stingray.com
www.stinray.com
Stingray create best of breed object orientated developer tools for Windows Programmers. Products can be sold direct and through Resellers and Distributors.

Superbase Developers Plc
14 Regent St, Cambridge CB2 1DB
Tel: 0118 944 8962 Fax: 0118 954 0760
Contact: Cathy Rowley
106253.3627@compuserve.com
www.superbase.com
Superbase 3.2 provides a powerful development environment for data based applications with modern features and graphical 2-way development tools including source code generation.

System Science
1-6 Bradley's Close, White Lion St
London N1 9PN
Tel: 0171 833 1022 Fax: 0171 837 6411
www.SystemScience.co.uk
Sales@SystemScience.co.uk
The UK Specialist for all Software Development tools, representatives of Pervasive (Btrieve), InstallShield, RoboHelp and many more US vendors.

Take Five Software Ltd
The Surrey Technology Centre
40 Occam Rd, Surrey Research Park
Guildford, Surrey, UK
Tel: ++44 1483 295050
Fax: ++44 1483 573704
Email: info@takefive.co.uk
http://www.takefive.com
SNiFF+ The integrated cross-platform programming environment for C/C++, Java, IDL, Fortran and other languages.

VBxtras
1905 Powers Ferry Road Ste 100
Atlanta, GA 30339 USA
Tel: 770-952-6356 Fax: 770-952-6388
www.vbxtras.com
The VBxtras Catalog is The Ultimate Source of Tools and Information for the Visual Basic Developer Worldwide. Visit the VBxtras Web site to get the latest VB info or to download & buy VB tools instantly off our ESD Web site. VBxtras has a 30-day money back guarantee and the lowest prices anywhere!

FORTRAN AND C++

Salford Software Ltd
Adelphi House, Adelphi Street,
Salford M3 6EN
Tel: 0161 834 2454
Fax: 0161 834 2148
Sales@salford-software.com
www.salford.co.uk
Fortran and C++ compilers with unique CHECKMATE switchable run-time diagnostics. Low cost student editions available.

HELP SYSTEMS

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

LIBRARIES

Wintertree Software Inc.
69 Beddington Ave. Nepean,
Ontario, Canada K2J 3N4
Tel: 001 613 825 6271
Fax: 001 613 825 5521
sales@wintertree-software.com
www.wintertree-software.com
Spelling checker and thesaurus components for C/C++, MFC, Visual Basic, or Delphi applications.

MFC LIBRARIES

Hypercube, Inc.
Los Angeles, CA, USA
Tel: +1 310 559 2354
Fax: +1310 559 2357
harlan@hcube.com
www.hcube.com
MFC Extension Libraries. HyperView++: VBA-like embeddable forms environment. HyperDraw++: VISIO-like diagramming / drawing. Free demos on www.hcube.com. Buy direct and save.

MIDDLEWARE

OpenLink Software
Amy Johnson House
15 Cherry Orchard Rd
Croydon, Surrey CR9 6BB
Tel: 0181 681 7701 Fax: 0181 681 7702
Contact: www.openlink.co.uk
OpenLink Software is an industry-leading developer and deployer of secure, high performance database connectivity technology, independent operating system, network protocol and underlying database engine.

PROGRAMMING TOOLS

ZAC CATALOGS
www.InstantSoftware.com
Huge selection of visual and internet development tools! Complete technical information, downloadable demos, on-line ordering and instant on-line delivery. 30 day money-back guarantee.

PUBLISHING

Addison Wesley Longman (Addison-Wesley)
Edinburgh Gate, Harlow
Essex CM20 2JE
Contact: Customer Info Centre
Tel: 01279 623928 Fax: 01279 414130
enq.orders@awl.co.uk
www.awl-he.com/ computing
A wide range of books on software development, object-oriented technology and programming from authors such as Watts Humphrey, Erich Gamma, Fred Brooks, Donald Knuth & Bjarne Stroustrup.

Digital Press
Butterworth-Heinemann, Linacre Hse,
Jordan Hill, Oxford OX2 8DP
Contact: Sophie Foster, Product Manager
Tel: 01865 314456 Fax: 01865 314572
www.bh.com
sophie.foster@repp.co.uk
Leading Publisher of books for professional programmers and software engineers. Digital Press publishes high-quality, leading edge books on topics such as Microsoft Exchange, Windows NT Server, Intranet Development, Microsoft SQL Server, Open VMS, X-Window Systems, UNIX, Linux, Lotus Notes, Web Servers.

Prentice Hall Europe
Campus 400, Marylands Avenue
Hemel Hempstead, Herts, HP2 7EZ
Contact: Customer Services
Tel: 01442 881891 Fax: 01442 882288
ibd_orders@prenhall.co.uk
We are the publisher of books and CD-based training materials in all areas of computing - and distribute leading imprints such as QUE and SAMS.

O'Reilly
Sheridan House, North Way
Andover, Hants
Tel: 01264 342832 Fax: 01264 342761
Contact: Customer Servs Dept
e-mail: itpuk@itps.co.uk
Leading publisher of books for Perl, C++, UNIX, Open Systems, Windows NT and the Internet.

RECRUITMENT

LeadPeople Limited
Newstead House, 107 Eastfield Road
Peterborough PE1 4AU
Tel: 01733 315510 Fax: 01733 314857
email: admin@LeadPeople.com
www.LeadPeople.com
I.T. Recruitment specialists covering London City, UK and Ireland.

SECURITY PRODUCTS

Aladdin Knowledge Systems UK Ltd
1 William St, Windsor, Berks SL4 1BB
Tel: 01753 622266 Fax: 01753 622262
sales@aldn.co.uk www.aks.com
Aladdin is a leading supplier of advanced software security (HASP) and smart card development tools (ASE) for software developers.

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

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

Rainbow Technologies Ltd
4 The Forum, Hanworth Lane
Chertsey, Surrey KT16 9JX
Tel: 01932 579200 Fax: 01932 570743
sales@uk.rainbow.com
Only Rainbow delivers leading edge technology and ISO certified quality for software protection and license management.

Sheriff Software
Acudata Limited, Clarendon House
Shenley Road, Borehamwood
Hertfordshire, WD6 1AG
Tel: 0181 905 2552 Fax: 0181 959 8583
sales@sheriff-software.com
www.sheriff-software.com
The Sheriff SDK provides secure and flexible copy protection without the need for dongles or royalty payments. Includes classes and demos for Visual C++, Visual Basic, Visual FoxPro and Delphi.

Softlok International
Softlok House, 14 Bark Street East
Bolton BL1 2BQ
Tel: 01204 436000 Fax: 01204 436025
sales@softlok.com
www.softlok.com
Protect your future, protect your software. Established in 1987, Softlok specialises in software piracy protection.

Superbase Developers Plc
14 Regent St, Cambridge CB2 1DB
Tel: 0118 944 8962 Fax: 0118 954 0760
Contact: Cathy Rowley
106253.3627@compuserve.com
www.superbase.com
Superbase 3.2 offers powerful data encryption and an RSA encryption library with user-definable key lengths and built-in email support for secure communications and data management.

TRAINING

Brooks Associates
Bismore, Gloucestershire GL6 7DG
Tel: +44 (0)1452 770060
Fax: +44 (0)1452 770078
Email: bbrooks@cix.co.uk
Contact: Bob Brooks
World class training for Delphi professionals.

CRaG Systems
178 Bath Road, Thatcham,
Berks RG18 3HJ
Tel: 01635 873670 Fax: 01635 868557
exe@cragsystems.co.uk
www.cragsystem.co.uk
CRaG Systems provides training and consultancy for both Object Oriented and Structured Analysis and Design of either business or real-time systems. Object Oriented techniques follow the Unified Modeling Language (UML) and Structured techniques the Yourdon method with Ward-Mellor real-time extensions.

Database Programmers Retreat Limited
The Old Fleece, Bisley Street,
Painswick, Stroud, GL6 6QQ
Contact Christine Shakespeare
Tel: 01452 814 303 Fax: 01452 813 918
100710.303@compuserve.com
www.dp-retreat.com
DPR is a Training, Consulting and Development Company offering scheduled, customised, 1-2-1, on-site education. All our courses are based on practical, hands-on exercises and we aim to teach programmers to develop database applications in Delphi, Visual Basic, Access, Clipper and VO.

Learning Tree International Ltd
Mole Business Park
Leatherhead, Surrey KT22 7AD
Contact Jan Mott
Tel: 01372 364600 Fax: 01372 364611
uksales@learningtree.com
www.learningtree.com
Training for IT Professionals, Totally integrated training solutions from the worlds finest independent training company.

QA Training
Part of QA Group Ltd, Cecily Hill
Castle, Cirencester, Glos GL7 2EF
Contact: Customer Service Team
Tel: 01285 883388 Fax: 01285 883399
responsecentre@qatraining.com
www.qatraining.com
QA Training is the UK's largest IT training company with operations worldwide. We have unrivalled experience in technical IT training and continue to pioneer the development of new courses.

The Object People Limited
Epsilon House, Chilworth Science Pk
Southampton, SO16 7NS
Tel: 01703 769996 Fax: 01703 766066
ukinfo@objectpeople.com
www.objectpeople.com/uk/
The Object People have a world-wide reputation in assisting clients adopt and make successful progress with object technology.

Valtech Ltd
Corinthian House, St Giles Circus,
279 Tottenham Court Rd
London W1P 9AA
Tel: +44 (0) 171 307 2300
Fax: +44 (0) 171 307 2301
Web : http://www.valtech.com
training@valtech.com
Valtech is an international training and consulting group offering a wide range of courses, to transfer the expertise to develop new multi-tier systems. Valtech has trained 5000 people throughout Europe in 1997, and is recognized as the European leader in Object Technology Transfer.

WEB APP DEV

Unipalm
1 St Marys Court, High Street
Newmarket, Suffolk CB8 8HQ
Tel: 01638 569669 Fax: 01638 569601
Email: exe@unipalm.co.uk
Specialist Intranet/Internet software distributors. Cold Fusion Application Server, Cold Fusion Studio for rapid deployment of database driven Web applications. Netscape - award-winning client, servers and development tools for Web, messaging and crossware applications.

RWJ **Robert William James & Associates**
Professional Search/Contract Staffing

Searching for highly desirable professionals...

C++ SOFTWARE ENGINEERS £20 k - £32k (dep on experience)
Ref: IC100/S To participate in on-going product development and customer bespoke solutions, working on one of several new projects at initial stages of development. Experience of: C++ Development (Windows NT/Unix), OOD & development essential. Windows MFC, Client/Server development, COM /DCOM or CORBA advantageous.

PROJECT LEADER - SENIOR SOFTWARE DEVELOPER £33k+
Ref:IC101/S Experience of managing staff and involvement in all areas of the product life-cycle essential. The successful candidate will be experienced in the following: maximising productivity of teams within budget customer support, Design, Development, QA procedures VB5, SQL, SDDM

PRODUCT SUPPORT MANAGER £Neg
Ref:IC102/S This candidate will plan and direct the support team and all associated activities, including documentation, testing, quality assurance and customer support. Working knowledge of following would be advantageous: Software release strategy and awareness of AVR2000 3.1 VB5, SQL

SOFTWARE ENGINEERS (Junior & Senior) £16k - £32k
Ref:IC103/S Primary Requirements: At least 2 years software development experience with an imaging/electronics background. Essential requirements: Visual C++within NT/95/98 environment. Programming for real time applications would be advantageous.

For more information please contact us or simply send your C.V. to: Ian Coles at RWJ & Associates, 2nd Floor, Haymarket House, Wote Street, Basingstoke, Hampshire, RG21 7NL.
Tel: 01256 818833 Fax: 01256 816144
e mail: ianc@meritr.demon.co.uk

In and Around West Yorkshire

We have clients currently seeking the following:

Applications Trainer............. **circa £20,000 + international travel**
Perl/UNIX Programmers................................ **£18,000 to £23,000**
Visual BASIC Developer/Team Leader for small development team .... **£20,000 to £25,000**

'C++' Developer, 1 year's experience **£16,000**
'C' or 4GL Programmer for ANADATA development .... **£20,000 to £25,000**

Junior Games Programmer, 'C' or 'C++' .... **£12,000 + bonus**
'C++' Developer (at least 2 years strong skills)......... **£20,000 to £25,000 (includes some trips to Monaco)**
MS Access/Visual BASIC Developer for project management systems................................. **circa £20,000**

VBA/MS Access Developers (rural location)..................... **£15,000**
Good Programmer (any language) to work in Visual BASIC or 'C++' or DELPHI or ORACLE (training given)................... **£20,000**
Visual BASIC/SQL Developers (includes some help desk work).................................... **£15,000 to £20,000**

For your next career move around West Yorkshire telephone **Vincent Atherton** on **Leeds (0113) 250 4560** or write to:
AIREDALE RECRUITMENT
Realtex House, Micklefield Lane, Rawdon, Leeds LS19 6AX
email: airedale@btconnect.com

Airedale Recruitment

THE SOFTWARE DEVELOPERS' MAGAZINE

EXE

Let EXE help you to promote your company

Take advantage of our Reprint Service for your own promotional purposes. We can print up articles which mention your company (as they appear in the magazine) as separate brochures and leaflets. We can also add your company information or an advert, depending on layout.

Please call Tarina Starkey on 0171 970 6542 for details and costs.

Read all about iT

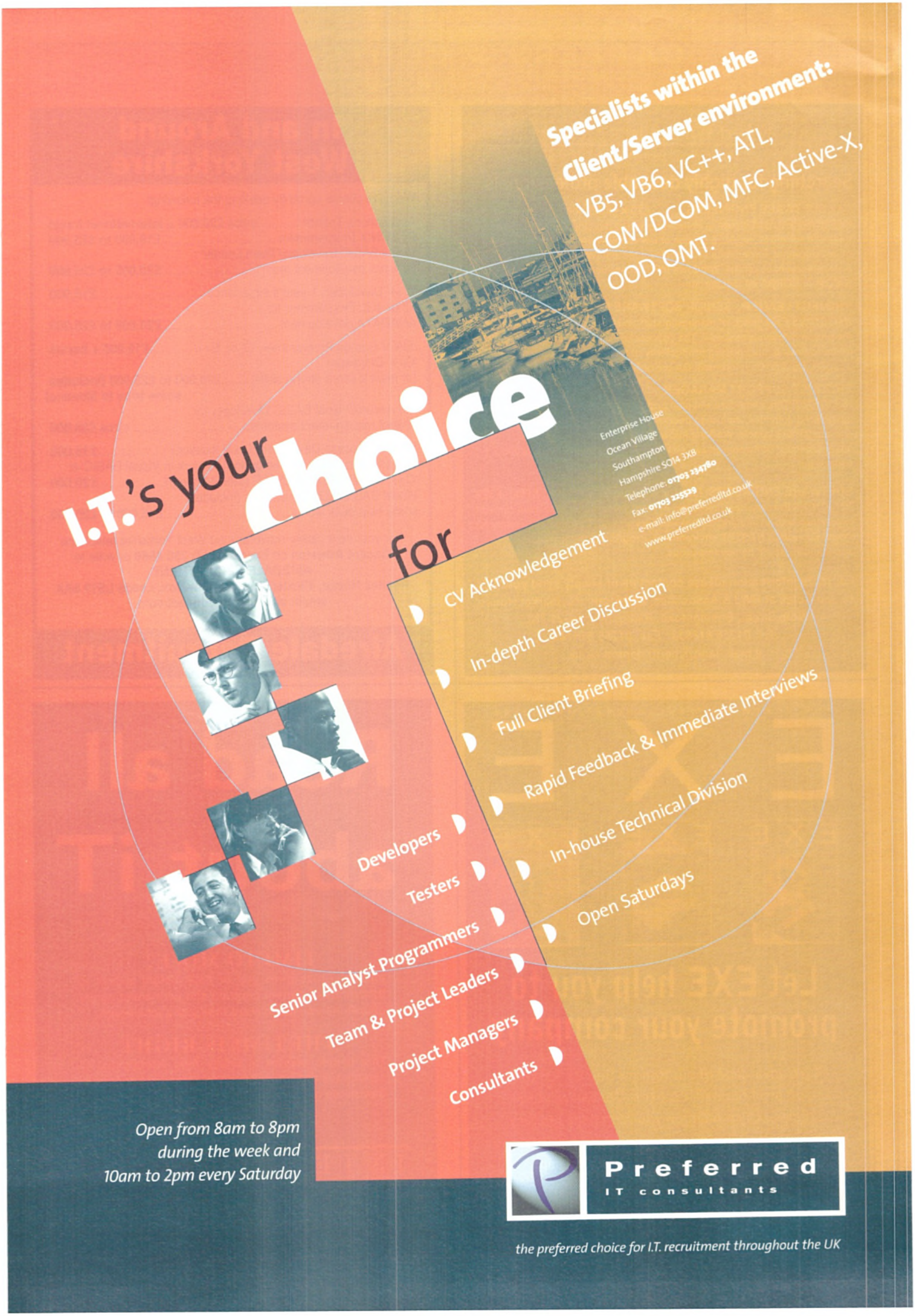
TIMETO MOVE?

Internet Times gives you 24 hour instant access to thousands of current contract and permanent career opportunities - much more than just a jobs board.

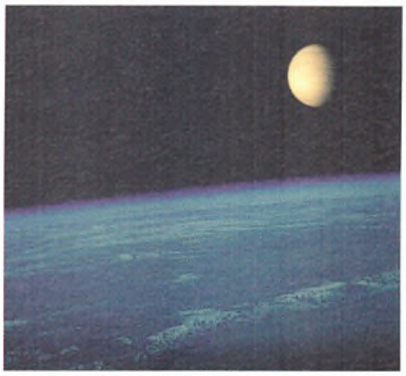
THETIMEIS RIGHT

This unique service is totally free, confidential to candidates and offers easy and immediate access to thousands of career opportunities.

INTERNET TIMES
www.internettimes.com
Freephone: 0800 0831216


Specialists within the
Client/Server environment:
VB5, VB6, VC++, ATL,
COM/DCOM, MFC, Active-X,
OOD, OMT.
I.T.'s your choice for
Enterprise House
Ocean Village
Southampton
Hampshire SO14 3X8
Telephone: 01703 234780
Fax: 01703 225529
e-mail: info@preferredltd.co.uk
www.preferredltd.co.uk
CV Acknowledgement
In-depth Career Discussion
Full Client Briefing
Rapid Feedback & Immediate Interviews
In-house Technical Division
Open Saturdays
Developers
Testers
Senior Analyst Programmers
Team & Project Leaders
Project Managers
Consultants
Open from 8am to 8pm
during the week and
10am to 2pm every Saturday
Preferred
IT consultants
the preferred choice for I.T. recruitment throughout the UK

A long time ago, in a galaxy far, far away...



Fans are already starting to queue outside theatres for tickets to see the latest installment of the *Code Wars* trilogy, *The Cobol Menace*. The new film, which stars Open Source demigod Linus Torvalds as Obi-Wan Kernel, Sun Microsystems' cross-platform guru James Gosling as JDBC Master Ace Windows, Oracle head honcho Larry Ellison as Supreme Chancellor Valium, and world's richest man and Microsoft CEO Bill Gates as the evil Emperor Palpitations, is set some twenty-five years before the founding of the C++ Empire. It follows the adventures of young hacker Anaemic Heapwalker as he begins his fall to the Closed Side of The Source. He meets a young Obi-Wan Kernel, who notices that all his code compiles first time and realises the Source is strong with him.

Schooling him in the Open ways of the JDBC Knights, Kernel teaches Anaemic to release early and often, and never to be tempted by the Copyright ways of the Closed Side.

Meeting up with JDBC Masters Key-Pair Jini and Ace Windows, the two become involved in a brutal war for control of the planet Gnu. As robot programmers from the Cobol Federation lay waste to the streets of Gnu, the four must struggle to survive against the onslaught of the Dark CGIs and their evil master, Emperor Palpitations, Lord of the Closed Side.

Asked if six weeks was too long to queue for a film, one moviegoer commented: 'Absolutely not. I've got my palmtop here, so I can still work on my revolutionary new version of `vi`. Would you like a copy?' At this point, we made our excuses and left.

Ask Dr Pentium II Bunny Person

EXE's own latex-suited chip-making dude solves your problems, technical *and* personal.

Q I'm deciding on a web application system for my corporation's e-commerce site. It has to be up 24/7, and make me look really cool and sexy, because at the moment I'm having tremendous problems getting a girlfriend. The trouble is, I don't know what server-side scripting technology to specify, and I'm so depressed.

A Yo, dude! You could be having sooooo much more fun with a Pentium II Processor™. It's great for games and stuff, and like, you can surf the Web *while* you surf a tube with a Pentium II-powered laptop. It's awesome! Whoa! Oh, yeah, use ASP, dude. It's a total babe-magnet. You'll have to beat them off with a stick.

Q Is it just me, or does software have a lot more bugs in it these days? It hurts when I cough and I'm sure that Office 2000 is to blame. Perhaps it's that Melissa virus everyone's talking about. What should I do?

A No way, dude! It's not just you. Software **does** have a lot more bugs in it today. But I don't think Office 2000 is to blame for your, like, respiratory problems, because it's optimised for the Pentium II Processor™, which is great for office stuff as well as games! You've probably been drinking too much Jolt, dude! Time to cut back, relax, strut your stuff – I recommend *Rollercoaster of Love* by the Red Hot Chili Peppers. Hang on a minute – I've just gotta go make another annoying ad...

Dr Gerbil is on holiday.

Return of the Great Yellow Hat

As you may remember, way back in the dim and distant days of February, when winter was still upon us and it got dark at 4 o'clock, we rather rashly organised the Great EXE Yellow Hat Competition. I say rashly, because we asked the question: *why are manholes circular?* without taking into account the possibility that there might be more than one answer. Which of course there is. Not only that, but as one entrant helpfully pointed out, not all manholes are actually round. Some are square. Some are triangular. There are even a few ellipsoid ones out there. Doh!

So the Great EXE Yellow Hat Competition Judging Panel sat in session long into the night debating, and we decided to assume that the manhole in question was in fact round, and accept either of the following answers (one of the entrants handily supplied both, thus earning double brownie points):

- *So they won't fall down the hole*
- *It's easier to fit a circular cover because it doesn't have corners*

The correct answers were then put into the Yellow Hat, and a winner and a runner-up drawn out by His Editorial Highness. So congratulations to our winner, Mr Ian Butterworth of London, who gets the Yellow Hat, and our runner-up, Mr Jason Judge of Whitley Bay, who gets a slightly tacky Intel Pentium II Bunny Person doll of his very own.

If you didn't win this time around, don't be disheartened. There'll be another chance to win stuff that we don't want anymore, soon.

Open® warfare

'"If you ever again... insult me," trilled Eric [Raymond] to his old Open Source pal [Bruce Perens], "and jeopardize the interests of our entire tribe... I will find a way to make you regret it. Watch your step." Bruce promptly contacted the police, and warned everyone he knew ("Because I know Eric is a firearms enthusiast").' – *Need To Know*

I was down in the west suburbs on one of those jobs that you know isn't going anywhere, like a random rush-hour shortcut in a strange city. Violets M'Gee had called up and asked me to rustle him up a few components, and just to be cute I had said I would. So I mooched away the morning, doing about as much work as a paid-by-the-hour contractor on an already-cancelled project, and about lunchtime I somehow found myself near the Slashdot, so I pushed the door open and went in.

Slashdot isn't the sort of joint where you would want to take yours or anyone's mother. The place was full of the usual crowd, elderly geeks in elderly T-shirts, arguing and bickering about nothing at all ('stuff that matters') in that sluggish, twitchy/bitchy way that they do. A pack of maggots crawling over a five day old mouse and making a big deal about it. I like it there. On a stage at the back there was the musical entertainment: a darkly-bearded hippyish type produced a kind of nasal keening noise with words. If you asked him, he'd probably have told you he was singing a protest song. 'Protest' was right, anyway.

I slunk in and sat down at a corner table and set fire to a Camel and got out a Windows mag and propped it against the wall. This was a mistake because almost at once a very large bouncer-shaped creature came over and used up all my reading light. To say his face was meaty would be to undersell the case. It looked like the whole dead cow. He leaned close enough that it felt like he was going to fall on top of me, and I noticed that MODERATOR was printed on the undercurve of his belly, which seemed like a long word for this kind of person, even on a shirt. He smelled of pizza. He poked at my magazine with a finger.

'You Borg?'

'No, me Klingon. You Sontaran?'

He shook his head, but whether in denial or in puzzlement or just because he had an itchy scalp I couldn't tell. He repeated his question.

'You a Microsoftie?'

I had just opened my mouth to say something stupid when there was a shot and a scream and some glass breaking and the smell of cordite, and the bouncer leapt away with an alacrity I would not have believed possible. A stranger stood in the doorway, like in every Western you ever saw. He had a Luger in his right hand, pointing upwards at the light fitting he had just shot. The front sight was filed off, which meant he was good, or thought he was. I recognised him from the Wanted pictures on the Web, of course. It was Eric 'Murder in the Cathedral' Raymond.

'Hey you. Come over here.'

He was beckoning to me with his left hand. It seemed sensible to do what he wanted. My titanium-lined combinations were in the wash that day. I got up and walked towards him until I stood about four yards away. The gun was still pointing skywards, which was better than having it pointed at me, but still not a clean compile by any stretch. 'Cathedral' growled at me.

'Where's Brucey?'

'Who's Brucey?'

As a stall it was much less convincing than a 16-year-old flunking her first ever hill-start.

'Who do you think, Billie-Borg? Where's his Debs hiding? In the john?'

'He's not here, Cathedral. This thing has gone far enough. Why don't you be sensible and go home quietly?'

A neatly dressed young man with glasses and a thoughtful face and a Scandinavian lilt had stood up. At first glance he wasn't obviously insane. 'Cathedral' swung around.

'Oh yeah? And take my medicine like the Gnome guy?'

The week before a group of marauding KDEs had personally flamefested a rural Gnome-aloner with a tommy-gun. I could tell 'Cathedral' wasn't in a reasoning mood.

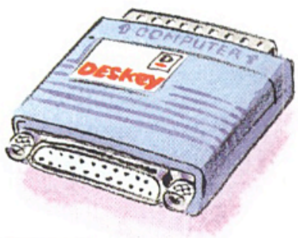
As he started to bring down his gun arm I charged him on his blind side and kicked him in the back of his knees. A gun went off – not the Luger, another gun – and out of the corner of my eye I saw the Scandinavian man fall down and the bouncer run past me with something black in his hand. I fell down myself in a heap with 'Cathedral' on top and on the way down something like a table edge hit me on the back of the head and I began to lose consciousness.

A woman's voice said, it seemed from a long way away, 'Oh my God. They did it. They only went and shot Linus.' And then, just as I blacked out, another voice, an even fainter male voice, said, 'This time the FreeBSD has gone too far.' ■

One makes software theft illegal, the other makes it impossible.

If you would rather take the law into your own hands, the DESkey range of products have security designed into the hardware. ASICs and microprocessors running proprietary algorithms provide real protection. A comprehensive range of drivers and our software protection utility DESlock, work to bring the highest level of security with the minimum of effort.

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



Don't just Dongle it – DESkey it

PC MAC PC Card UNIX etc

Designed for Microsoft Windows 95

PC CARD

Member of FAST Promoting the legal use of software

Quality Assured Firm ISO 9001 SGS

ACS ASIC

Mac OS

Ready for OS/2 WARP

DESkey

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