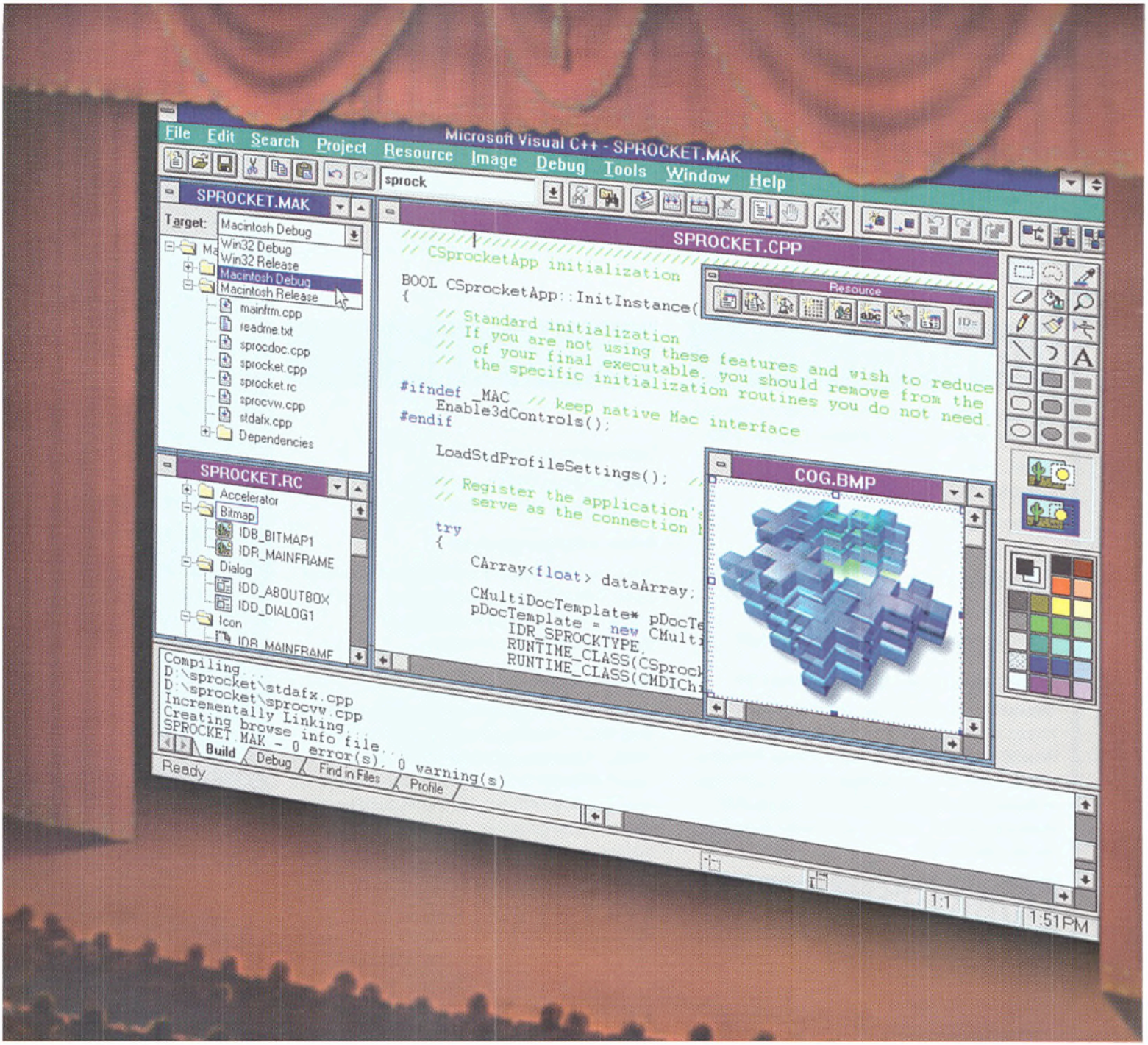
WinRegistry

A supplement to EXE Magazine

November 1994



projects to the 32-bit world.

The user interface has been recrafted. There's a new project manager, an integrated code and resource editor, a browser on demand, even an advanced debugger with dockable windows, customisable toolbars, just-in-time debugging and more.

Afraid to sacrifice speed for "friendliness"? You don't.

There's an incremental linker that links only what you change. New compiler optimisations increase application speed. And MFC 3.0 adds proven features in seconds, saving weeks of rekeying. So it's actually faster.

It's easy to create OLE 2.0 and ODBC™ – enabled applications for a 32-bit environment, such as Win 32s™ and Windows NT™ 3.5.

It's also the first step towards Windows™ 95.

If you want more, don't talk to your screen. Ask about the new versions of Visual C++ 2.0 for Intel,® MIPS® and Macintosh.™ To order contact your usual reseller or telephone the Microsoft Upgrade Centre on 081-614 8000.

Microsoft

Editorial

Our last special Windows supplement appeared more than two years ago. Things have changed considerably since then: it's now time to issue a new one.

Whilst there are still plenty of Win16 tools available, the major novelty in the Windows development scene at present is the appearance of tools aimed specifically at Win32. With these you can develop applications that will execute today on Windows NT (and if you limit yourself to the Win32s subset on Windows 3.x with Win32s installed). But most important, they will run next year on Windows 95 (aka Chicago): to wear the new 'Windows 95' logo, Microsoft requires that the software runs smoothly on NT *and* Windows 95.

Get the right tool for the right job. For rapid application development or prototyping, you'll find plenty of visual programming environments. On the other hand for more complex developments or applications which should run fast, C++ is now the language of choice. Whether to develop real object oriented code or just plain C structured code, a C++ compiler is up to the task.

Don't reinvent the wheel, life's too short! A lot of standard code has already been written and is packaged as libraries or C++ classes. Once mastered, a good library will not only reduce development time but it may also help in porting an application between several environments. MFC, for example, should be available shortly for the PowerPC architecture.

Some years ago most software development began by coding a suitable language in assembly code. Then SDKs which included basic tools became common. As can be seen in this guide, a plethora of tools are now available to help at all stages of the development process: editors and version control for the source level; memory checkers and debuggers for after the first code has been generated.

This supplement covers the tools you need to deliver 16 and 32 bit applications, with some additional features for that little extra something. With Microsoft pushing Win32 as the universal API, it's probable that next year you'll see the number of Win32 tools overcrowd the Win16 arena. Another important trend that should be more visible with the arrival of Windows 95 is connectivity. Initially applications will be integrating more network services than simply remote printing and file sharing. Eventually they should be global network aware with the ability to retrieve information directly from, say, the Internet.

One thing unfortunately missing in this issue is a list of drinks to suit your every programming need. But hopefully, with all these tools utilised to maximum effect you'll have time to find out for yourself!

DM

Contents

The EXE Enterprise

Our mission: to seek out new code libraries, to boldly go where no Software Development magazine has gone before.

Our promise: to read all your email and letters be they in Forth, LISP, BASIC, C, C++, American or just English.

Editorial ..... David Mery
Production ..... Melanie Welsh
Advertising ..... Steve Miles
Advertising ..... Marc Green
Front Cover ..... Jacqui Ramrayka
Subscriptions ..... Amina Haque (071 439 4222 ext2252)
Subscriptions ..... Fiona Mitchelson (071 439 4222 ext2212)

The WinRegistry Windows Supplement is published by Process Communications Ltd, publisher of EXE Magazine.

Process Communications Ltd, St Giles House, 50 Poland Street, London W1V 4AX.
Telephone: 071 287 5000, Facsimile: 071 437 1350, email: cliffs@dotexe.demon.co.uk

```
Sub Form_Load ()
    Show
    GetDataBaseNameFromUser
    On Error GoTo ErrorHandler
    Set dbObject = OpenDatabase(msDbName)
    FillTableList
    cboTableList.ListIndex = 0
    Exit Sub
ErrorHandler:
    msg = "Unexpected error." & Error(Err)
    MsgBox msg, 48, "Database Display"
    Exit Sub
End Sub

Sub GetDataBaseNameFromUser ()
  dlgGetDbName.DialogTitle = "Select Database For Display"
  dlgGetDbName.Filter = "Access d/b (*.MDB)|*.mdb"
  dlgGetDbName.Flags = &H10&
  dlgGetDbName.Action = 1
  msDbName = dlgGetDbName.Filename
  lblDbName = dlgGetDbName.Filetitle
End Sub

Sub FillTableList ()
  cboTableList.Clear
  For i = 0 To dbObject.TableDefs.Count - 1
    If dbObject.TableDefs(i).Attributes And DB_SYSTEMOBJECT <> 0
Then
    Else
      cboTableList.AddItem dbObject.TableDefs(i).Name
    End If
  Next i
End Sub
```

Figure 2 - How to load a form

So far, the only information the user has supplied is the location and name of the database to be opened. The application can interrogate the database's TableDefs collection to obtain information about each table object it contains. The first item used is the collection's Count property. This is almost universally implemented in collections. It enables us to iterate through all the instances of the child object within the collection. A TableDefs object has several properties, for example, whether or not it can be updated.

The application first checks the attributes property and ignores any that are flagged as being system tables (used internally by the database management system and of no use to us). Notice that when the Count property of the TableDefs collection is being referenced it is not indexed. The attributes and name property that refer to an individual TableDefs object are indexed. The name of each table which is not a system table is added to the table list box and is available for selection by the user.

A bug considered as a feature

The first time round we force this event by setting the list index property in **Form_Load**. This is a useful 'feature' in VB and one that must be used with caution as it is noted in the list of bugs that will probably be fixed with version 4. You can't count on this still working in the next release and so it should not be used in a live application.

Let's have a look at **cboTableList_click** in Figure 3. The field position counters are initialised then the **FillIndexList** routine is called. This works in exactly the same way as **FillTableList** but this time we are iterating through the Indexes collection of the selected TableDefs, populating the list box with the name property of each Index object in the collection. So the Indexes collection is a child of the TableDefs object which is itself a member of the TableDefs collection of the database object.

The last thing to do is to display the content of the database table, i.e. the Fields collection. We use this collection to obtain the name, type and size of each field in the table. The **AddNewFieldToDisplay** routine is then called to prepare a text box for each item in the collection. Notice that in this event procedure, TableDefs is referred to directly by name rather than by index. Since some of the tables were discarded, the position on the list does not necessarily represent a table's position within the collection.

The application can interrogate the database's TableDefs collection to obtain information about each table object it contains

The **SetHeightOfInnerPanel** routine deals with enabling the scroll bar where necessary. Any controls in excess of those needed for this table are made invisible and **tblObject** is set to the name from the selected TableDefs. This exposes another Fields collection, this time with the all important Value property that will allow the **PopulateFields** routine to display the actual data. The Index property of the table is set to the name from the indexes collection list box.

Where the work is done

The **AddNewFieldToDisplay** routine is called to prepare a frame control to display the field name and a text box to contain data for each table's field. Rather than having to keep track of how many controls were loaded from the last table, the application attempts to load afresh each time and handles the error condition which will result if the control is already loaded.

The row and column counters are incremented as appropriate with reference to the field's potential size. Where possible we will want to save screen space and display three small fields side by side rather than using a row for each. The new controls are positioned and sized correctly. The ZOrder method will ensure that the text box does not end up underneath the frame. The frame's caption is set to the field's name and finally controls are rendered visible.

The application must ensure that all of the controls that have been loaded onto

C, C++ and BASIC programmers, now you get much more than xBase compatible DBMS power.

Thousands of programmers have already discovered how to get dBASE, FoxPro and Clipper compatibility with their favorite language and hardware platforms. For example, one customer has C programs running on PC and Sun workstations sharing data with concurrently running FoxPro for Windows applications.

You see, CodeBase technology is simply the best way to add multi-user xBase compatible DBMS power to C, C++, Basic or Pascal.

You still gain speed & small size

CodeBase users really appreciate our small executable size. Unlike SQL engines which are a Meg or so in size, CodeBase 5.1 EXE's can be as small as 45K! You'll also like the speed–with our Intelligent Queries you get the execution speed of C plus stunning query performance from our smart use of available index information.

Now formatted data entry in Windows is as easy as point & click!

Experienced Windows programmers know **formatted** data entry is difficult to program under Windows. But with our new CodeControls, you can simply 'Point & Click' to design data entry windows for date, numeric, and character information–formatted just the way you want it.

***Introducing the new* CodeControls**, a unique set of data-aware custom controls. Now simply drop them into your Windows applications via your favorite visual interface builder.

NEW–Data-aware controls

Our new custom controls are ***data-aware,*** so now you can easily build a scrolling list box that's tied to a data file, or look up matching combo box entries–even as the user types.

Introducing CodeReporter 2.0

***Introducing the new* CodeReporter 2.0** our visual, interactive xBase report writer. We designed it with developers in mind, but end-users will love it.

Create a wide variety of reports –visually, easily, and instantly.

Use ***CodeReporter's*** new *Instant Report Wizard* to create a report–in an instant. To refine your report, simply drag and drop report objects–for data, totals, text or graphics–using the interactive layout screen.

Easily build report queries using our calculator-style expression builder. Then get your reports lightning fast through the built-in Intelligent Query Technology.

And get multi-platform portability.

Once your reports are designed under Windows, you can generate corresponding source code. Use this source code to launch reports under DOS, Macintosh, Windows, NT, OS/2 or UNIX.

Introducing CodeTranslator 3.0

Now you can automatically translate Clipper, dBASE, and FoxPro code into C++. Turbo-charge critical xBase applications, port to new operating systems, and gain the flexibility of C++.

CodeTranslator keeps your variable names and uses the CodeBase++ library–making the translated code easier for you to read and maintain.

Buy One, Get Two FREE.

Now when you buy any one of our xBase compatible library products: ***CodeBase, CodeBase++, CodeBasic*** or ***CodePascal*** (for the language of your choice), you'll get both the new ***CodeReporter 2.0*** *AND* the new ***CodeControls 2.0*** absolutely **FREE –for a limited time only.**

TO ORDER NOW CALL 071-833 1022

System Science

Your Sequiter Authorised Dealer

SEQUITER SOFTWARE INC.

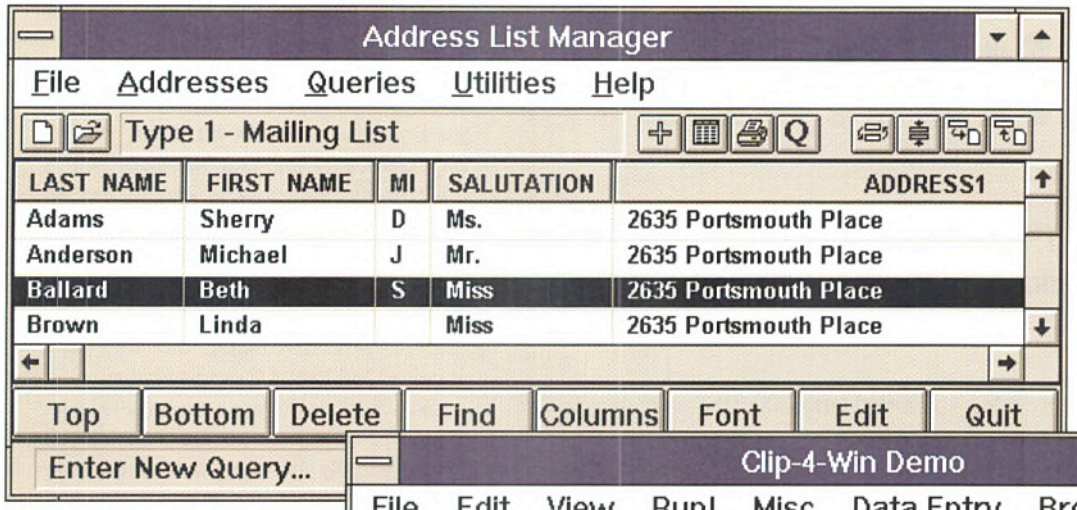
FAX 403•436•2999

UK Tel. +44-81-317-4321

P.O. Box 575 Newmarket NH 03857-0575

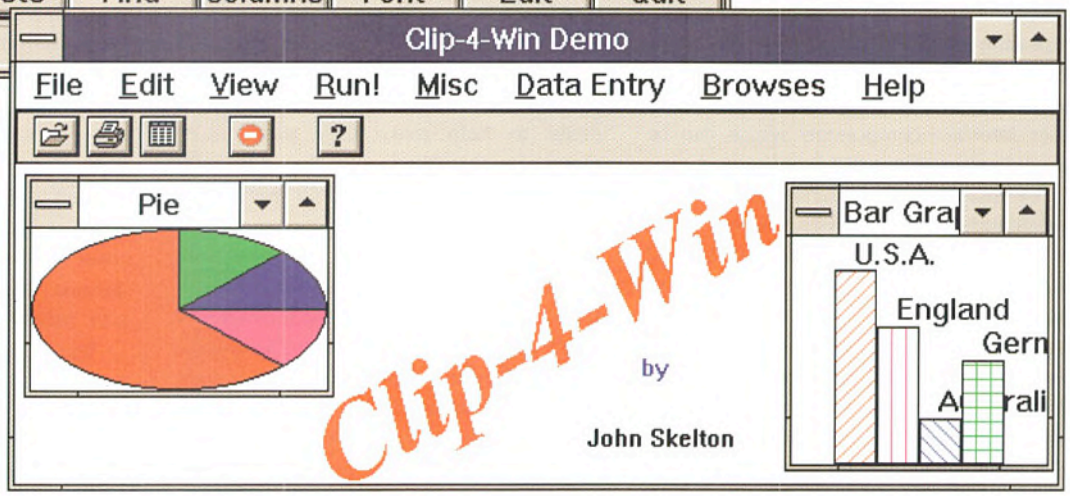
Clip-4-Win

Fast Windows Xbase Applications - NOW



Sample Application

(source code supplied)



Demonstration of Windows features

(source supplied)

Start Learning Windows

With the add-on library Clip-4-Win you can start programming for Windows in the Clipper language you already know. No need to struggle with C/C++.

Great GUI Application

Provides TrueType browsers, toolbars, menus, mouse support, etc. so you can produce stunning applications.

What about Visual Objects ("Aspen") ?

Visual Objects (aka Aspen) is the Windows Xbase product that CA are busy working on. It is not expected to be shipping for some months. You can use Clip-4-Win to develop and ship Windows applications today. The public demonstrations of Visual Objects have shown that Clip-4-Win and VO have very similar function calls (both modelled on the Windows API). You're not locked in to Clip-4- Win .

Use Lots of Memory

Under Windows you can use lots of memory without having to overlay at all. The 640K barrier is simply not there. You don't have to wonder how to squeeze your Clipper program into the 350-450K typically left after loading your network software .

Event Driven

In keeping with Windows, Clip-4-Win provides you with the functions required to write event-driven software. This gives the user much more freedom .

DeBugger Support

The Clipper debugger uses its own window.

Simple Access to Windows API

It's quite easy to use from Clipper... For example,

```
MessageBox(0, "Press Enter", "Info", MB_OK)
```

You can use menus, brushes, windows, scroll bars, icons, cursors, clipboard, resource files, DLL's, DDE, etc. very easily.

Other Add-Ons

Compatible with: Class(y), Comix, dbClass, FlexFile, FUNCky, Graphics Server, Grumpfish products, NetLib, NovLib, ObjectDB, SIx Driver, TechWriter, most other Clipper add-ons, and most DLL's.

Requires

Clipper 5, Enhanced mode Windows 3.1, and a Windows-compatible linker, such as Blinker 3, Microsoft LINK 5.30 or SLR Systems' OptLink. The Clip-4-Win Toolkit provides both linker and interactive GUI designer.

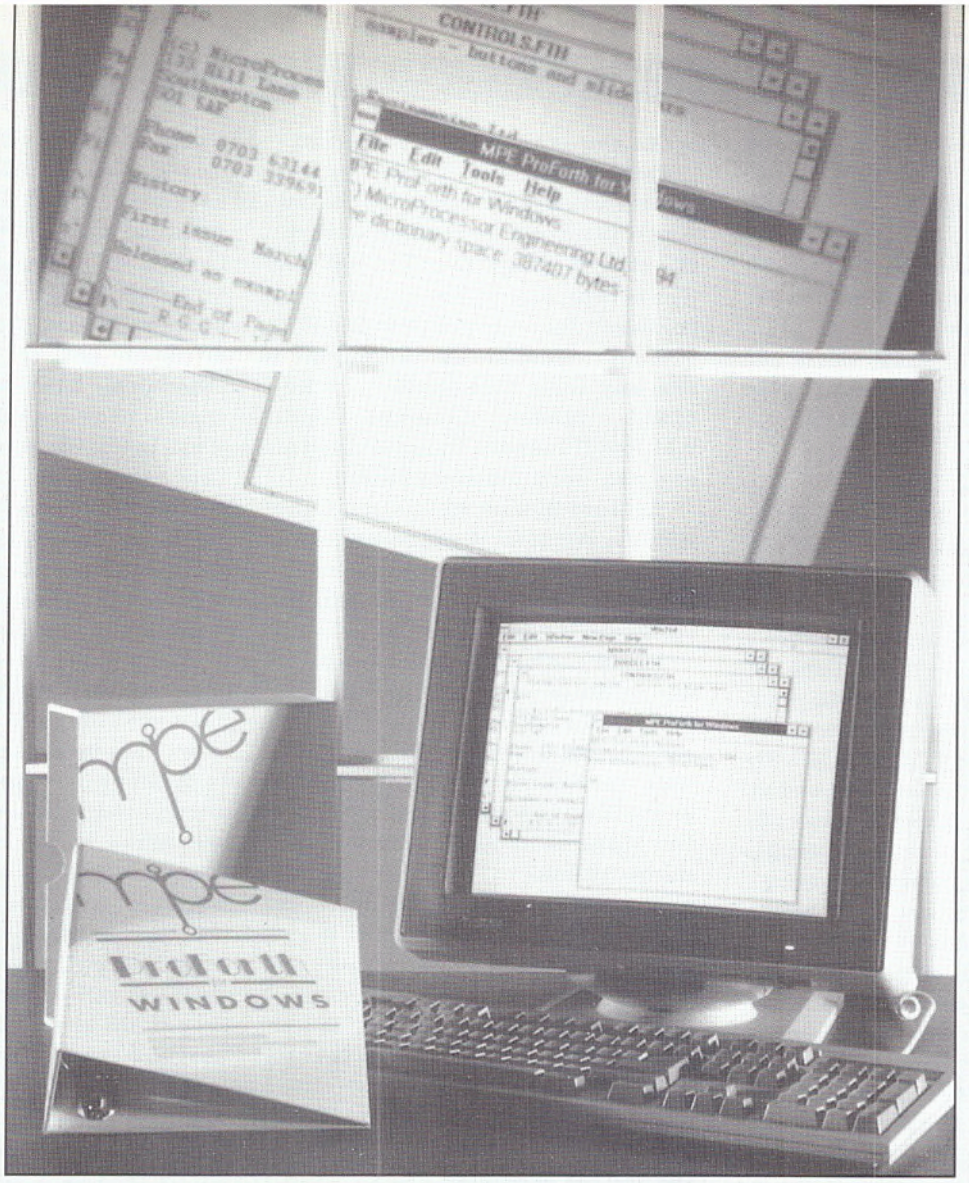
Call for a FREE demo

QBS Software Ltd
10 Barley Mow Passage
London W4 4PH
Phone: +44 81 994 4842
Fax: +44 81 994 3441
Email: qbs@cix.compulink.co.uk

QBS SOFTWARE

081 994 4842

Published by Skelton Software, Kendal Cottage, Hillam, Leeds LS25 5HP, UK.
Tel: +44 977 683296, Fax: +44 977 681650, Compuserve: 100112,3102



Faster Windows Development With Forth

- interactive Windows development and debug
- reduce development time
- handle Windows with ease
- create a window in 4 lines of code
- draw the window - ProForth Guide writes the source code
- real time capabilities
- multitasking and timers

ProForth for Windows

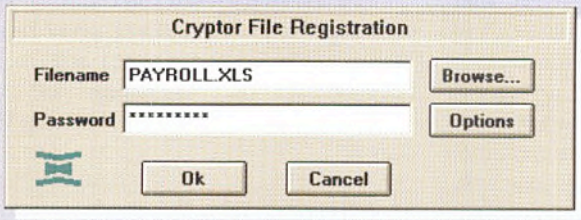
brings interactive development to Windows, and Windows brings a visual operating system to Forth. The combination is a very powerful tool, making ProForth for Windows the fastest way to develop Windows applications. ProForth is a 32 bit Forth for Windows 3.1 and Windows NT.

Microprocessor Engineering Ltd

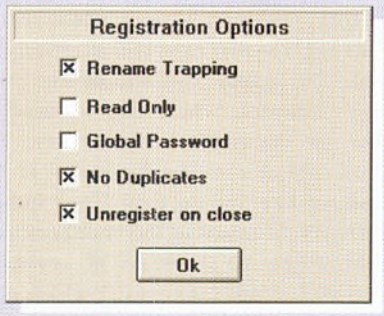
133 Hill Lane
Southampton SO15 5AF
Tel: 01703 631441
Fax: 01703 339691

XITECH CRYPTOR

Transparent file encryption for Windows applications via a software solution.
***Not* seeing is believing.**



Cryptor provides desktop security for all files under the control of a Windows application. Simply by entering filenames and passwords through the interfaced DLL provides access to those chosen encrypted files only from within the host application.



The proprietary encryption algorithms which combine uniquely for each copy provide security of the highest degree for sensitive files and data. Ideal for securing personal and financial details, satellite laptops and notebooks, or protection of electronically transmitted data. Shared access of protected files between any DLL referencing application. Specific interface for easy access within FoxPro (FLL) and for compatibility with FoxPro DOS Cryptor.

- **Supports any Windows applications that can access the DLL**
- **Eight choices of encryption levels on installation**
- **Unique algorithm derived from Serial Number**
- **Easily integrated software solution**
- **Network compliant**
- **Files always remain encrypted on disk**
- **Optional temporary file registration for further security**

Cryptor operates as a Virtual Device Driver (CRYPTOR.386) added to the Windows SYSTEM.INI file, allowing this seamless integration and operation. The DLL can be called programmatically or via an interface from within a particular application.

DLL (per user) £79, FoxPro (FLL) £299, FoxPro Runtimes £15

For more information please contact : Xitech (Europe)
Tel 0707 276637 Fax 0707 267777 Compuserve 100117,3316 CIX Xitech
The Lindens, Chantry Lane, Roe Green, Hatfield, Hertfordshire, AL10 9PH

Also from Xitech: **ReFox - FoxPro decompiler & brander**
ReSource - dBase IV 1.1 & 2.0 decompiler
FoxFix - FoxPro file repair tools & royalty free library

xitech

WINDOWS HELP FILES

ONLY £110 +VAT

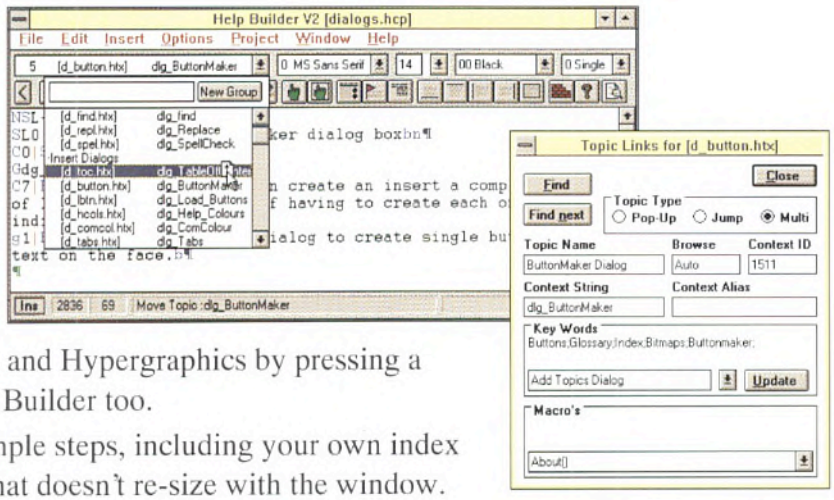
Produced *even more* Quickly, Simply and Efficiently

With Help Builder Version 2.0

The ABC Help Builder is a complete application designed to produce all the files needed by the Microsoft help compiler.
It is a simple to use tool for anyone who needs to produce on line documentation of any description, not just application help files.
Version **II** has been packed with new features to make it even easier to use and above all to help you create professional help files as simply and effortlessly as possible.
Version **II** of the Help Builder has been under development for over a year, to make it the most versatile help authoring tool you can buy.
You have complete control in all aspects of the look and feel of your finished help file.
There are a number of effects that can be created with the Help Builder that are difficult if not impossible to do with a standard Word processor and other Help Authoring applications.
The Help Builder utilises re-usable topics that can be included in other Help Projects.

The Help Builder's features include

- ❑ **Generates all compiler files**
- ❑ **Insert bitmaps and 'sheds'**
- ❑ **Multimedia file support**
- ❑ **Secondary windows**
- ❑ **Dialogs to insert HPJ Options**
- ❑ **Full featured spell checking**
- ❑ **Imposes no style restriction**
- ❑ **Group topics in logical order**
- ❑ **Simple to create contents list**
- ❑ **Glossary creation features**
- ❑ **Creates bitmaps for buttons**



☺ **Insert** Bitmaps, Audio, Video and Hypergraphics by pressing a button, view them in the Help Builder too.

☺ **Create** a glossary in a few simple steps, including your own index buttons - not a hypergraphic that doesn't re-size with the window.

☺ **Reports** window allows you to view and act on all errors and warnings produced by the Help Builder and the Help Compiler.

☺ **Dialogs** to generate the majority of the HPJ file entries, you can of course edit the HPJ file within the Help Builder to suit yourself.

☺ **Unique** Audit system to accelerate testing of your completed help file. It provides three extra buttons and a text editor in the Help window, to navigate logically through the Help file and make notes. You can then view this file in the Help Builder to make any amendments. The easiest way to proof your Help files.

HBII
Help Builder V2

Try it for yourself:
A free, fully working, evaluation copy is available on request

ABC Software

Pentecot, Windsoredge, Nailsworth, Gloucestershire, GL6 0NN
Sales: Tel. and Fax 0453 833325

Written, produced and supported in the UK

Nu-Mega
TECHNOLOGIES INC

AVOID EMBARRASSMENT!

In Windows, DOS, Windows NT, Win32s - Even Chicago... With BOUNDS-CHECKER™ Tools, You Can Develop On Five Different O/S, And Catch Program Bugs!

The BOUNDS-CHECKER Experience If you're a C/C++ programmer, it's a must! ***Here's why:*** BOUNDS-CHECKER puts you in control of software program bugs. It's that simple! No other development tool tracks down C/C++ program bugs so thoroughly and so automatically. ***Here's how it works:*** BOUNDS-CHECKER sets hundreds of breakpoints within your program to monitor its behavior. When a bug is detected, BOUNDS-CHECKER immediately stops your program and pops up with a detailed description of why the error occurred. You can then inspect your program's source, variable, stack and heap with BOUNDS-CHECKER's powerful display windows to help determine the exact cause of the problem. In addition, BOUNDS-CHECKER logs all Windows events, including API calls, API returns, and other important events. These events can be viewed with T-VIEW™, BOUNDS-CHECKER's exclusive event viewer - a tool designed to help you see the flow of your program.

BOUNDS-CHECKER - Now Multi-O/S! Nu-Mega has just the right BOUNDS-CHECKER to see you through your next release - In Windows, DOS, Windows NT, WIN32s, and coming soon - Chicago! That's right! Nu-Mega has packed all the power of our16-Bit BOUNDS-CHECKER automatic debuggers plus even more features, into a family of 32-Bit tools that programmers must have! New 32-Bit features include thread control and API compliance capabilities. So tackle NT, take-on WIN32s, and look toward Chicago!

Easy To Use There is no learning curve with BOUNDS-CHECKER. Simply select your program's name from BOUNDS-CHECKER's menu; all the rest is automatic. There's nothing to link-in and no macros to compile into your program. BOUNDS-CHECKER tools use the debug information produced by your own compiler. All this, plus the functionality of a *single comprehensive automatic bug finder!*

Stop Bugs Like These At Their Source!

- API Parameter Errors
- API Compliance Errors
- Resource Leakage Problems
- Processor Faults
- API Return Value Errors
- Data And Heap Corruption
- Memory Leakage Problems

BOUNDS-CHECKER, Your Guide To Chicago! It's the newest BOUNDS-CHECKER, and it's coming soon! If your new application has you "Chicago-bound," - take this new wonder along for the ride. It'll help you navigate through even the most complex passages of Chicago, so you can make your new release on-time and error-free!

Get It Right In Your O/S! Using BOUNDS-CHECKER during application development just makes sense. It's like a built-in first line of defense against program bugs. The fact is, using BOUNDS-CHECKER today can save you a lot of grief in the long run - Avoid Embarrassment, make sure all of your programs have been BOUNDS-CHECKED! Rely on Nu-Mega and BOUNDS-CHECKER to make your next release the best ever!

Make Your Order A Trio And Save A Bundle!

Choose any three BOUNDS-CHECKER products **(including BOUNDS-CHECKER For Chicago)** - and get the third FREE!

- ❑ **BOUNDS-CHECKER For Windows**
- ❑ **BOUNDS-CHECKER32/NT**
- ❑ **BOUNDS-CHECKER For WIN32s**
- ❑ **BOUNDS-CHECKER For DOS**
- ❑ **Chicago Reservation!** Yes, I want BOUNDS-CHECKER For Chicago Now! (Call us for ordering information)

£195 + VAT each

Special Trio Price: £415 + VAT! Single Orders: £195 + VAT each

Call System Science Now! (0171) 833-1022
Or Fax This Form! (0171) 837-6411

❑ VISA ❑ MasterCard

Card # _____ _____ _____ _____ Exp. _____

Card Holder Name (Print) _____

Signature _____

Phone: _____

OverNight Delivery! Risk=Null/30 day money-back Guarantee

Call: (0171) 833-1022
FAX: (0171) 837-6411

SYSTEM SCIENCE

RISK = NULL
30 DAY
MONEY BACK GUARANTEE

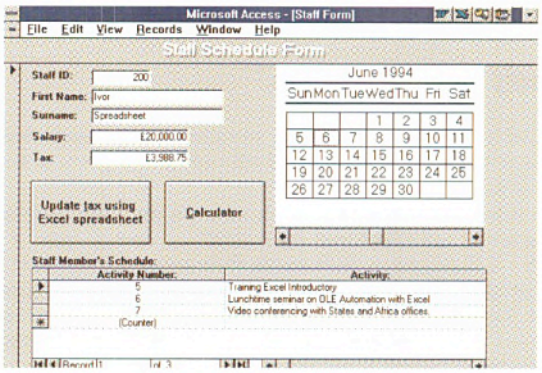
3-5 Cynthia Street
London N1 9JF

24 HOUR BBS
001-603-595-0386

Learn to develop the fast way

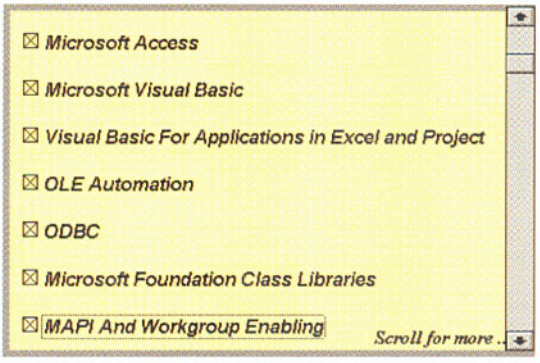
Don't waste months trying to figure it out - we can tell you what you need to know in just a few days!

Microsoft Access - master the features which make Access the hottest thing since filing began:



- Access DAO (Data Access Object) Model
- OLE Automation
- Windows API programmability
- OLE custom controls
- Include calendars and tab dialog boxes
- Create your own wizards and tooltips
- TTTs (Tips, Tricks and Traps)

Visual Basic - the strategic language of the next few years:



- DLLs and the Windows API
- Programming the jet database engine
- Object data types
- Create Windows Help files
- Master event driven programming

Database Design - create well-designed, self documenting database applications:

- Conceptual data modelling
- Asymmetrix Infomodeler
- ORM (Object Role Modelling)

Call us now for a *free* audio cassette of Access/Visual Basic tips and tricks!

BLUENECK

COMPUTER TRAINING

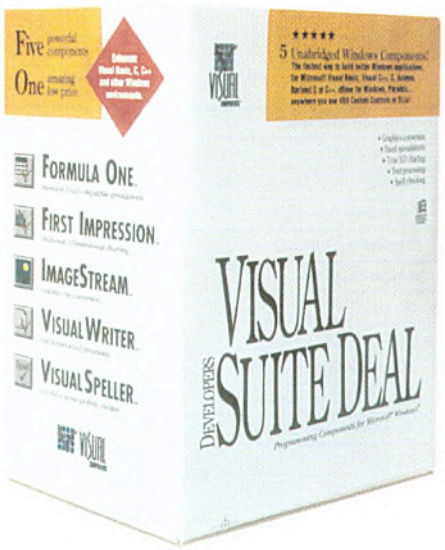
9 CHANDOS ROAD • REDLAND • BRISTOL • BS6 6PG • UNITED KINGDOM

TELEPHONE 0117 974 4416/923 7758 • FAX 0117 923 8172

Microsoft | Solution Provider
Authorised Training Centre

Special Autumn Windfall

The Visual Tools Suite - 5 superb VisBas products for the price of one!



VisualWriter

VisualWriter provides developer's with complete text processing functionality. It allows WYSIWYG display, print features, character and paragraph formatting for various presentation styles, large text storage capacity and zooming functionally that enables scaling factors to view small details or overall documents appearance.

Formula One

Formula One is a fully functional spreadsheet development tool for Visual Basic and many other Windows-based environments. Speed - Power - Flexibility - Database Connectivity - Excel compatibility - Royalty-free runtime distribution - Over 126 spreadsheet functions. In short, a splendid product which will save you money, time and resources.

VT Speller

VT-Speller is a 100,000 word spell checker which seamlessly adds spell checking functionality to your Windows-based apps. Comes as a DLL or VBX for easy integration into VB and many other Windows apps.

ImageStream

ImageStream is a complete import, export and graphic file conversion tool. It provides total and accurate translation of both vector and bitmapped images, and supports the widest variety of of graphic file formats. Also supports printing, zooming, compression and image processing.

First Impression

First Impression is a true 3-D business charting and graphing tool which produces fabulous presentation quality output. Creates understandable charts and graphs from the most complex data sets. Full object oriented design allows you to control all attributes, including rotation, perspective, fill patterns, text style and more.

All for just £225.00 !!

Save over £500

Special Offer Number Two

Dan Appleman's SpyWorks-VB PLUS Visual Basic Programmer's Guide to the Windows API by Dan Appleman

SpyWorks-VB allows you to do virtually anything in Visual Basic that is possible using other languages like C. It includes a complete set of controls for subclassing, Windows hooks, and Callbacks.

SpyWorks-VB includes a full set of Visual Basic aware debugging tools, online help, and lots of sample source code.

- Customize VB and third-party controls
- Add entries to control menus
- Create rollups and half-size captions
- Build extensions to other applications
- Create virtual forms
- Communicate between forms and applications
- Trap WM_USER and other messages
- True LostFocus, GotFocus detection
- Trap Windows API Parameter Errors
- Use Windows Callbacks
- Create system "hot-keys"

Includes details on over 700 Windows API functions!

Turn your Visual Basic knowledge into Windows programming expertise by tapping into the powerful tools of the Windows application programming interface.

- Extend the capabilities of Visual Basic Gain a full understanding of how to create Visual Basic declarations for any dynamic link library and avoid application errors.
- Deepen your understanding of such Windows concepts as event-driven programming, device contexts, and device independance.
- Create, copy, print bitmaps, metafiles, and other graphic objects.
- Use advanced printer features, including the printer setup dialog, functions to abort printing in progress, mixed portrait and landscape print mode printing, and more.
- Take control of mouse, keyboard and other system resourc.

Just £129.00 !! Save £24

Offers available for a limited time only.

QBS Software Ltd
10 Barley Mow Passage
London W4 4PH
Phone: +44 81 994 4842
Fax: +44 81 994 3441
Email:
qbs@cix.compulink.co.uk

QBS SOFTWARE

081 994 4842

AppWare™

NEW PRODUCTIVITY FROM A NEW DIVISION OF LABOUR

Commercial software satisfies business needs. If it doesn't, it may be software, but it isn't commercial. Successful, commercial software requires an effective combination of business analysis and technical skills. Developing applications without *AppWare* has typically involved compromising either the product's technical sophistication, or its usefulness. When both technical implementation and business analysis are done under the same roof, they are effectively competing *against* each other for finite resources. Using third-party libraries still requires a high degree of 3GL competence, and constrains program maintenance.

The distribution of processing across different computers on a network can lend a number of advantages to software systems. However, without *AppWare*, a specialist is almost always required to design in network benefits. Since the methods used by network applications vary widely and involve diverse and occasionally incompatible external dependencies, it is not unusual to require a product expert just to install network applications as well.

Novell's *AppWare* development system accelerates the creation of applications, improves their fit to market requirements. It requires experts in business processes to master a limited set of fundamental programming skills, but in so doing they gain effective control over their information technology. At the same time, it isolates the complexities of particular techniques and implementations, including the complexities of computer networks, by encapsulating these in *AppWare Loadable Modules*. Business applications developers benefit from the competition among technical developers. Technical developers benefit from the widest possible deployment of their expertise.

DESIGN, BUILD, AND MAINTAIN YOUR OWN BUSINESS TOOLS

Business requirements change constantly, and often quickly. If you're one of the sharpest in your field, you'll soon see how your information technology must respond to these changes. If you are using *Visual AppBuilder* efficiently, you will be able to design and implement the required changes correctly and with confidence. If necessary, you will be able to specify unambiguously your requirements for enhancements needed from 3GL or technically-specialised programmers.

Visual AppBuilder is a high-level programming tool that enables you to construct full-featured applications by linking icons that represent objects and methods exported by pre-built *AppWare Loadable Modules* (ALMs). ALMs are interchangeable software components produced by Novell or third parties. At present, Novell and third parties provide a number of ALMs including messaging and database. Using *Visual AppBuilder*, you combine and configure ALM icons into an application within a graphical environment; the structure of even quite complex application is still clearly visible. Subsequent maintenance and re-use is therefore much easier than with 3GLs.

We at **Etobico Technology** can assist you to acquire the *AppWare* skills necessary to express your business expertise in software. We can also develop with you a framework and strategy for your adoption of *AppWare*. If required, **Etobico Technology** can also develop custom ALMs which can give you a unique competitive advantage, or develop archetype applications for you to adapt and change in the future.

DEPLOY YOUR TECHNOLOGY WHERE IT ADDS THE GREATEST VALUE WITH THE GREATEST EASE

If you have best-of-breed technology that can give a competitive advantage to applications developers, you can increase your investment's value through *AppWare*. Put your product into the hands of the new breed of expert programmers using *AppWare* by releasing an ALM. **Etobico Technology** can shorten your time to market and maximise your effectiveness through targeted training and technical services. We have been working with *AppWare* throughout its Beta programme and have several more years' commercial experience in *NetWare Loadable Module* development and the design of distributed applications for NetWare.

ETOBICO TECHNOLOGY LIMITED
111 SOUTH CROXTED ROAD
LONDON SE21 8AX
ENGLAND

ETOBICO

INTERNET: EXE9411@ETOBICO.COM
COMPUSERVE: 100137,3456

TEL./FAX: +44-181-4731910

SYSTEM SCIENCE

See us on Stand 191 at SD'94, NEC, Nov 22-24

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

Windows Development

Doc-To-Help - help authoring	£259
Robo-HELP - help authoring	£365
InstallSHIELD - installation utility	£325
LeadTools Imaging library	£call
Smalltalk/V Windows	£295
PC-Logo Windows	£60
PI DLLs for WP & graphics import	£call
PI Document Editor - new pricing	£345
Spread/VB++ - spreadsheet function	£169
Professional Toolkit	£239
Zinc Application Framework	£call
Graphics Server SDK	£245
SmartHeap Win or DOS	£345
TCP/IP Development tools	£call
CodeBase or CodeBase++	£239

VB Value

PI for Vbx - new import filters	£call
Disctinct TCP/IP Visual Ed	£179
Spyworks by Desaware	£115
Spread/VBx	£165
TrueGrid Pro	£115
VB Assist	£139
Designer/Data Widgets	£95
VB Tools 4.0	£110
QuickPak Professional for Windows	£149
QuickPak Scientific for Windows	£125
PinPoint/VB - API debugger	£125
Grid/VBx, Tab/VBx	£call

Mathematics & Science

MathCad Plus	£365

- Mathematica - DOS, Win, Mac, Workstations, Education & Student pricing.
- Derive - school & university licensing.
- Origin new ver 3.5 with add-on modules.
- Lindo & What's Best Linear Programming
- SigmaPlot, SigmaStat, TableCurve

...many more for maths & stats.

UNIX

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

For FORTRAN

Lahey FORTRAN 90 - new	£625
FORTRAN to C translators	£call
MS FORTRAN-77 5.1 - DOS	£145
MS FORTRAN Powerstation - DOS	£235
MS FORTRAN Powerstation - NT	£415
IMSL Numerics for MS Fortran	£call

Cross Development

Hi-Tech C Cross - many tragets	£call
Aztec C for 68K, 8086 & Z80 targets	£call
Introl C Cross - many targets	£call
2500AD Cross Asm & Simulators	£call

Pharlap DOS Extenders

TNTIDOS Extender - 32 bit	£335
286IDOS Extender - 16 bit	£335

Microsoft

Visual C++ 2.0 CD, & VC++ 1.5 16-bit	£299
Visual C++ 2.0 CD upgrades MS C/C++	£149
Visual Basic Windows PE	£235
Visual Basic DOS PE	£235
Macro Assembler 6.11- NT or DOS	£110
Access 2.0 (upgrades available)	£289
Access Developers Kit	£295
Multimedia Viewer	£239
Project for Windows	£375
Windows for Worgroups, Windows NT	£call
Office and Office Professional	£call

(See FORTRAN Section)

Editors and Tools

CodeWright Editor Windows ver. 3.1	£175
CodeWright Fusion for MS VC++	£109
Multi-EDIT Pro - DOS	£129
DemoQUICK - create Windows Demos	£395
Dan Bricklin's Demo II - DOS Demos	£195
PC Lint C/C++ - Lint code checking	£149
Flowcharting & diagramming tools	£call
Track Record - bug tracking software	£110
Archimedes BugBase	£375

MKS

MKS Toolkit of Unix tools	£210
MKS RCS - Version Control	£275
MKS Lex & Yacc for C, C++ & TP	£215
Internet AnyWhere - off line news & Mail	£99

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

3-5 CYNTHIA STREET, LONDON N1 9JF
FAX 0171 837 6411

QUALITY IN CUSTOMER SERVICE AND SUPPORT

SYSTEM SCIENCE

0171 833 1022

Watcom

Watcom C/C++ 10.0 CD-ROM	£225

Printed documentation is also available

Watcom Fortran-77[32] - DOS,OS/2, NT	£375
Watcom SQL - DOS,Win,OS/2	£259

SQL Servers for DOS, Win, OS/2 & Netware

Borland

Borland C++ 4.0 - CD or disk	£295

Upgrades & Comp. Upgrades available

Power Pack for DOS	£69
Borland C++ OS/2 1.5	£249
Turbo C++ Visual Edition - Win	£85
Turbo C++ 3.0 & World C++ Video	£79
Visual Solutions Pack of Vbxs	£59
Turbo Assembler	£59
Brief Programmers Editor	£125
Borland Pascal & Turbo Pascal	£call
dBASE for Windows & upgrades	£call

Nu-Mega

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

Call for savings on Nu-Mega bumdles

PVCS Specialists

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

OS/2

OS/2 OS/2 for Windows	£49
IBM C Set++ bundle on CD	£185

Includes OS/2 Windows & Kase:Set

Visual Age	£1,295
CA-Realizer	£99

CLIENT/SERVER

ObjectView Desktop special	£169
Btrieve Development DOS or Windows	£395
Btrieve Client Engines DOS or Windows	£645
Scalable SQL for Netware	£call
PowerBuilder Desktop	£189
PowerBuilder Desktop C/S Starter	£425

With Erwin for PB, & Watcom 3 user SQL server

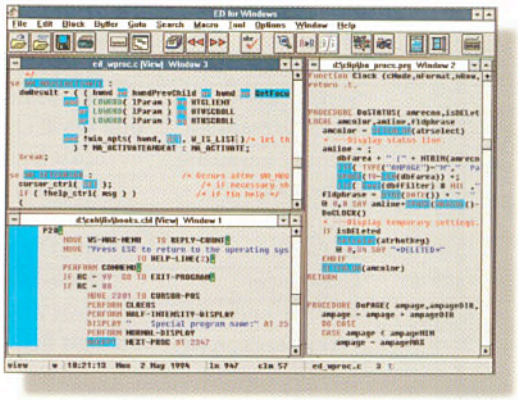
NEWS

- Microsoft Visual C++ 2.0 now shipping, with option of special offer Windows NT Workstation version 3.5.
- Btrieve Technologies demerge from Novell All Btrieve products & upgrades now available from System Science.
- EasyCASE Database Engineer module for SQL development and reverse engineering.
- BOUNDS-CHECKER Win/32s for Windows hosted 32 bit debugging.
- PowerBuilder Bundles with ER/WIN, Lotus Notes library and Funcky.
- New PVCS Tracker, Track Record and Bug-Base, for recording bugs and managing their resolution.

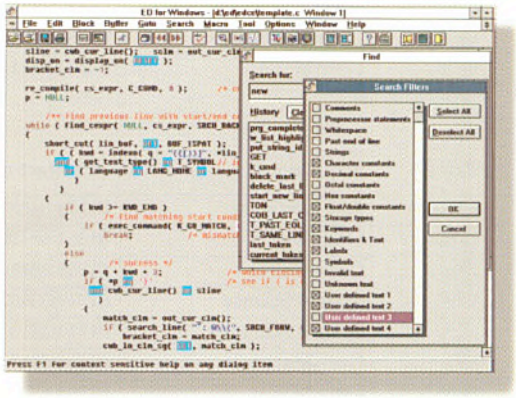
ObjectView from KnowledgeWare for Client/Server Development special £169

Is your editor cramping your style?

IT'S SURPRISING how many programmers put up with those ho-hum editors bundled with their compilers. Or even DOS-hosted monsters which to be, errm, brief, haven't seen a proper upgrade since before Thatch left the throne. If you have ever found yourself picking through megabytes of material by hand to accomplish a task that you know should be a few keystrokes, then we have a suggestion for you.



ED supports multiple windows, multiple files per window, and multiple languages too



ED's new advanced Search & Replace puts you in control

ED it

Introducing the new *ED for Windows* programmer's editor. ED is intelligent and language-sensitive - it knows all about your compiler. So whether you use C++, Clipper, Smalltalk, Eiffel or one of the other 24 languages that ED supports, you can be enjoying the benefits of colour syntax highlighting and compiler error tracking within minutes.

But it's the other stuff that makes the difference. Can't remember where you put a function? Just type its name in the File Open dialog. Lost in the on-line docs? ED will search through as many help files as you want at the click of a mouse. Like your menus and tool bars just so? A cinch with ED. There's a spelling checker that's smart enough to know about comments, a C-like macro language, and heaps more bits and pieces.

Try it

The best way to find out what ED can do for you is to have a go. And you can. For free.

Just call us up, and we'll send you a full working demo*. (If you are Wired, you can download it from any of the places listed at the bottom of this advertisement**).

...or Buy it

Already convinced? We can ship out your copy of ED for Windows 3.0 today. ED for Windows costs just £145 (+ shipping and VAT).

New Features in ED for Windows 3.0

Configurable Toolbars - Enhanced Compiler Support - New Spelling Checker - Easier Configuration - Precision Search and Replace - Improved File Open Dialog - Six More Languages including Smalltalk, Eiffel & Forth

The programmer's editor

QBS Software Ltd
10 Barley Mow Passage, LONDON W4 4PH
Phone +44 81 994 4842
Fax +44 81 994 3441
BBS +44 81 747 1979
Email qbs@cix.compulink.co.uk

QBS SOFTWARE

*Demo is fully functional, but won't let you save files > 2KB. **The ED for Windows V3.0 demo ED3DEMO.ZIP can be downloaded from the QBS BBS (tel 081 747 1979, baud rate 2400-14400, connect 8 N 1, in the DEMO section), from CIX (topic qbs/files), from CompuServe (GO CLIPPER, library 8) and from ftp site CICA (demo directory) and its mirrors (eg src.doc.ic.ac.uk, computing/systems/ibmpc/windows3).

Testing, Testing 1,2,3...

1994 BEST PRODUCTS open COMPUTING

"WinRunner
is the benchmark..."

MS-Windows
Ovum Report,
December 1993

LoadRunner
Product of the Year 1994

Client/Server
Unix World's
Open Computing,
January 1994

"XRunner
is the clear leader..."

Unix/X-Windows
Ovum Report,
December 1993

For Quality Software
Use Quality Testing Tools

mercury
interactive (UK) ltd

NT Version Now Available

Tel. 0634 262525 Fax. 0634 262828

Bounds-Checker for Windows V2.2	Debugging for VBXs	£195	System Science	0171 8331022
CCC/Manager	Source code control system	£385	Grey Matter	01364 654100
Defect Control System 2.10	Bug tracking tool	£470	Systems FX	01844 275175
LoadRunner\PC for Windows	Multi user and C/S appl. load testing tool	£8250	Mercury Interactive	01634 262828
MKS LEX & YACC 3.2 for NT	Lexical analyser & language generator	£240	Grey Matter	01364 654100
MKS Programming Platform for NT	Toolkit, RCS, LEX & YACC	£655	Grey Matter	01364 654100
MKS RCS 6.1	Version control	£295	Grey Matter	01364 654100
MultiScope Debuggers 2.0	Post mortem debugger	£279	Symantec	01628 592222
PCYACC	NT language generator	£910	Grey Matter	01364 654100
Portage Base	Software development kit for NT	£495	Grey Matter	01364 654100
PVCS Configuration Builder	Software configuration management	£260	Intersolv	01727 812812
PVCS Version Manager	Software configuration management	£400	Intersolv	01727 812812
SingleStep for Windows NT	Development and debugging for PPC & 68K	$1500	SDS	01703 643366
SlickMake 4.1	NT Make	£120	Grey Matter	01364 654100
Soft-ICE/W v1.5	Application and system-level debugger	£295	System Science	0171 8331022
SourceSafe	Multiplatform version control system	$295	One Tree Software	0101 919 8212300
Test Director	Multi user test planning, design & mgmt.	£5000	Mercury Interactive	01634 262828
TLIB Version Control 5.01	NT version control	£140	Grey Matter	01364 654100
Track for Windows 2.06	Bug tracking tool	£595	Soffront Software	0101 408 2632703
Track Record 1.5	Bug tracking tool	£149	SoftWerk	01440 730121
VERSIONS 1.1	Version control	£179	Grey Matter	01364 654100
WinRunner	Conformance testing of Windows apps.	£3000	Mercury Interactive	01634 262828
WinScope V1.2	Monitor and debugs system calls to any API	$149	Periscope Company	0101 404 8885335

Editors

Product	Description	Price	Supplier	Contact no
Codewright 3.0b	Editor	£205	Grey Matter	01364 654100
Codewright NT 3.0a	Editor	£205	Grey Matter	01364 654100
Ed for Windows	Editor with language syntax support	£149	QBS	0181 9944842
SlickEdit 2.3	NT Editor	£145	Grey Matter	01364 654100
Visual SlickEdit 1.5	Windows and NT Editor	£220	Grey Matter	01364 654100

Miscellanous

Product	Description	Price	Supplier	Contact no
ABC Help	Help builder	£110	ABC Software	01934 516714
Asymetrix InfoModeler Desktop ed	Case tool	£760	Grey Matter	01364 654100
Code.Print Pro for VB 3.0B	Visual Basic pretty printer	£85	Grey Matter	01364 654100
DemoQUICK	Create demos	£395	System Science	0171 8331022
Doc-To-Help	Creates help files with WinWord	£259	System Science	0171 8331022
EasyCASE Data Dictionary Maint.	Tool for EasyCase	£95	Grey Matter	01364 654100
EasyCASE Professional 4.1	Case tool	£665	Grey Matter	01364 654100
EasyCase System Designer	Case tool	£890	Grey Matter	01364 654100
EasyFlow for Windows	Flowcharter	£160	Grey Matter	01364 654100
FlowCharting 4	Flowcharter	£200	Grey Matter	01364 654100
HDK2 2.4	Help tool	£275	Grey Matter	01364 654100
HelpBreeze 1.6	Help builder	£240	Grey Matter	01364 654100
Help Browser 2	Indexed help browser	£49	Oxford Computer Cons	01865 793077
InstallSHIELD for Win NT 2.0	Installer	£520	Grey Matter	01364 654100
MWB - C	Large system code browser for C	£109	System Science	0171 8331022
MWB - C++	Large system code browser for C++	£119	System Science	0171 8331022
OPTLINK 5.1	Fast linker	£259	Symantec	01628 592222
PC-Install for Windows	Installer	£145	Grey Matter	01364 654100
Pretty Printer 2.0	Visual Basic pretty printer	£80	Grey Matter	01364 654100
RFFlow 3.0	Flowcharter	£75	Grey Matter	01364 654100
RoboHELP 2.6	Help builder from Workd	£370	Grey Matter	01364 654100
Select OMT	Case tool based on Rumbaugh's method	£495	Select Software Tools	01242 226553
Select OMT Professional	Superset of OMT with Perspective metod	£995	Select Software Tools	01242 226553
SmartHeap	Win 32 heap manager	£525	System Science	0171 8331022
Visual Planner	Project management and resource planning	£990	Information Builders	0181 98247000
wINSTALL	Installer	£245	System Science	0171 8331022

PCX Prog Toolkit - Win	Display, save, scale, manipulate PCX	£169	System Science	0171 8331022
PI Bitmap Import Filter	Import TIFF, PCX, GIF, BMP & EPS formats	£145	System Science	0171 8331022
PI EPS Filter	Import full vector EPS format	£1395	System Science	0171 8331022
PI Import Professional	Bitmap and vector formats filter	£1995	System Science	0171 8331022
PI Vector Import Collect	BMP, DXF, CGM, HPGL filters	£695	System Science	0171 8331022
PI Vector Import Filter	DXF, CGM, WPG, GEM, PIC, HPL & DRV	£295	System Science	0171 8331022
PICS 1.0	Screen library	£175	Grey Matter	01364 654100
Poet Client SDK 2.0	Database library	£720	Grey Matter	01364 654100
Poet Single User SDK 2.0	Database library	£360	Grey Matter	01364 654100
Professional Toolbox Win	DLL with spreadsheet, date, etc. functions	£239	System Science	0171 8331022
ProtoGen+ 4.1	Screen library	£220	Grey Matter	01364 654100
ProtoGen+ C/S Suite 1.0	Database library	£850	Grey Matter	01364 654100
Q+E DB Lib Dev System 2.0	Database library	£430	Grey Matter	01364 654100
ReportWriter Visual Coder	Database library	£109	Grey Matter	01364 654100
Rules	C++ tool for rule based programming	£5700	Ilog	01483 440388
SQLView	Database library	£395	Grey Matter	01364 654100
TE Dev Kit Windows	Text editing and WP library	£345	System Science	0171 8331022
3d Graphics Tools - C/C++	Graphic library	£245	System Science	0171 8331022
Tools.h++ 6.04	Persistent data functionality	£340	Grey Matter	01364 654100
VBtrv\C++ 2.1	Database library	£185	Grey Matter	01364 654100
Views 2.0 (single user)	Very graphical user interface in C++	£4500	Ilog	01483 440388
WinControl Library	Screen library	£160	Grey Matter	01364 654100
WindowsMAKER Professional 5.5	Screen library	£750	Grey Matter	01364 654100
WRAP	On-the-fly compression DLL	£200	Knowledge Garden	01753 833534
zAPP Windows & zAPP Factors	Portable interface development	£695	System Science	0171 8331022
Z-PHIGS Windows	PHIGS library for Windows	£1295	System Science	0171 8331022

Database languages and tools

Product	Description	Price	Supplier	Contact no
Btrieve Client for Windows	Database library	£645	System Science	0171 8331022
Btrieve Dev. Kit for Windows	Database library (needs an engine)	£395	System Science	0171 8331022
Btrieve Scaleable SQL DB Engine	Single scaleable SQL workstation runtime	£115	System Science	0171 8331022
CA-dBFast 2.0	XBase compatible development tool	£339	Computer Associates	01753 577733
Clip-4-Win 2.0	Create Win app with CA-Clipper 5	£240	QBS Software	0181 9944842
CodeBase 5.1	CodeControls and CodeReporter	£239	System Science	0171 8331022
CodeBase++	CodeWindows and Code Reporter	£239	System Science	0171 8331022
Crystal Reports 3.0	Database language	£149	Grey Matter	01364 654100
Crystal Reports Pro 3.0	Database language	£295	Grey Matter	01364 654100
Database Library	Data compression library	£179	System Science	0171 8331022
dBASE 5.0 for Windows	Database development system	£349	Borland	01734 321150
DDF Maker 1.1	Builds DDF files for Access to use Btrieve	£105	Grey Matter	01364 654100
Equinox 1.0 Single User	Database language	£279	Grey Matter	01364 654100
FoxPro 2.6 prof	Cross platfrom X-base database	£300	Microsoft	01734 270001
Greenleaf Archive Library	File compression library	£179	System Science	0171 8331022
KPWin SQLKIT	Add-in toolkit for KPWin	£300	Knowledge Garden	01753 833534
ObjectView Desktop 3.0	Database language	£190	Grey Matter	01364 654100
ODBC Sniffer for Windows 1.1	Database language	£295	Grey Matter	01364 654100
Paradox 5.0 for Windows	Database Development System	£349	Borland	01734 321150
PowerViewer	C/S queries creation tool	£140	Powersoft	01628 34500
Q+E 6.0	Access, query, analyse, integrate and report	£350	Intersolv	01707 812812
Select Schema Generator	Generates SQL from Storage Object Model	£495	Select Software Tools	01242 226553
Terrain	C/S database design	£4950	Bachman Info. Systems	01276 451007
TopWindows	Iterative development environment	£4950	TopSystems	0181 3322223
Watcom SQL Dev. Edition 3.2	Database language	£260	Grey Matter	01364 654100

Debuggers/Testers/Profilers/Source Code Control

Product	Description	Price	Supplier	Contact no
AllChange	Configuration and change mgment system	£1990	Readmar	0171 6255255
BugBase 1.64	Bug tracking tool	£360	Grey Matter	01364 654100
BugTrak	Bug tracking tool	£135	QBS	0181 9944842
Bounds-Checker32/NT v1.0	Detect most common Windows NT bugs	£195	System Science	0171 8331022
Bounds-Checker32/s v1.0	Heap checker, debug kernel, message logger	$195	System Science	0171 8331022

List & Labels for VB 3.0	Database library	£279	Grey Matter	01364 654100
MicroHelp Comms	Communication in VB	£109	QBS	0181 9944842
MicroHelp List n'Labels	Design database reports, labels, lists	£399	QBS	0181 9944842
MicroHelp Muscle	Over 600 routines to make VB progs faster	£119	QBS	0181 9944842
MicroHelp VBXRef	Cross reference and project management	£79	QBS	0181 9944842
Network Library/Win	Network library	£80	Grey Matter	01364 654100
NetPak Prof Win	Netware & WfWG functions for VB	£149	System Science	0171 8331022
PDQ Comm for Windows	Low-level control, event driven comms	£110	System Science	0171 8331022
PI Import Professional/VB	Bitmap and vector filters for VB	£375	System Science	0171 8331022
PI Vector Import Collection/VB	Import BMP, DXF, CGM & HPGL in VB apps	£295	System Science	0171 8331022
Pinpoint VB	Visual Basic API debugger	£135	System Science	0171 8331022
Q+E MultiLink/VB 2.0	Access PC and SQL databases	£270	Grey Matter	01364 654100
QuickPak Professional 3.2 for Win	30 custom controls, 400 routines	£145	Grey Matter	01364 654100
QuickPak Scientific for Windows	Scientific library	£105	Grey Matter	01364 654100
SpellPro	American English spellchecker	£119	QBS	0181 9944842
SpyWorks VB 2.0	Access most Win APIs from VB	£120	Grey Matter	01364 654100
3D Widgets 1.0	3D form designer	£89	Grey Matter	01364 654100
ToolThing	Ten productivity tools	£129	Bits per Second	01273 727119
Tools V1.0	Utilities and soure code modules	£99	The Mandelbrot Set	01941 117534
TrueGrid/VB Standard	Database program creation tool	£70	QBS	0181 9944842
TrueGrid/VB Professional	Superset of the Standard version	£130	QBS	0181 9944842
TX Text-Control & IC Image Contol	Text and image controls	£199	Grey Matter	01364 654100
VBAssist	Enhanced development environment for VB	£140	QBS	0181 9944842
VBGraphix	Charting and drawing tool	£325	Contemporary Software	01727 811999
VB/ISAM for Windows 2.3	Database library	£89	Grey Matter	01364 654100
VB/Magic Controls	Custom controls library	£120	Grey Matter	01364 654100
VB Projectworks	Project manager	£235	System Science	0171 8331022
VBTools 4.0	Custom controls library	£115	Grey Matter	01364 654100
VBtrv for Windows 2.1	Database library	£185	Grey Matter	01364 654100
Visual Instrumentation Panel Control	Technical library	£150	Grey Matter	01364 654100
VSVBX 4.0	Custom controls library	£54	Grey Matter	01364 654100
Visual/db 2.0	Database library	£115	Grey Matter	01364 654100
Visual/db Network db system 2.0	Database library	£235	Grey Matter	01364 654100

Code Libraries

Product	Description	Price	Supplier	Contact no
AccuSoft Image Format Library/NT	Image library	£850	Grey Matter	01364 654100
Btrvgen++ 2.51	Database library	£185	Grey Matter	01364 654100
CA-Common View	C++ application framework for GUI dev.	£399	Computer Associates	01753 577733
Canvas.h++ 1.0	Graphic library	£340	Grey Matter	01364 654100
CDR Import Filter	Import Corel Draw graphics in Win apps	£695	System Science	0171 8331022
COMM-DRV 14.0	Communication library	£145	Grey Matter	01364 654100
Control Palette/NC 2.0	Screen library	£135	Grey Matter	01364 654100
C++/Views 3.0	Screen library	£585	Grey Matter	01364 654100
DataBoss for Windows 1.0	Database library	£410	Grey Matter	01364 654100
DB Link	Connect C++ applications to relational db	£1700	Ilog	01483 440388
FTS for C/Windows	C full text search capability	£145	QBS Software	0181 9944842
GraphiC/NT 7.0	Graphical library	£370	Grey Matter	01364 654100
Graphics Server SDK 2.5	Graphics SDK	£205	Grey Matter	01364 654100
Greenleaf CommLib 5.0	Communication library	£235	Grey Matter	01364 654100
Greenleaf PowerComm Toolkit Win	Communication library	£130	Grey Matter	01364 654100
GUI Assist	GUI tool	£350	Grey Matter	01364 654100
Integra VDB C++ C/S 2.1	C++ encapsulation of ODBC	£509	Grey Matter	01364 654100
Integra VDB C++ Desktop 2.1	C++ encapsulation of ODBC	£209	Grey Matter	01364 654100
Linpack.h++ 1.1	OO interface to linear algebra	£515	Grey Matter	01364 654100
M.4	Expert system building tool	£685	Grey Matter	01364 654100
Math.h++ 4.1	Mathematical library	£340	Grey Matter	01364 654100
Money.h++	NT decimal fraction library	£340	Grey Matter	01364 654100
NeoImage API Pro	Image processing library	£210	QBS Software	0181 9944842
NeoImage API Std	Image processing library	£160	QBS Software	0181 9944842
NeoImage API Appl	Image processing library	£245	QBS Software	0181 9944842
ObjectGraphics C++ 1.0	Graphic library	£149	Symantec	01628 592222
Object Graphics 1.0 Turbo Pascal	Graphic library	£149	Symantec	01628 592222

386 Prolog for Windows	Prolog	£745	System Science	0171 8331022
Universal Cross-32 Meta-assembler	Cross assembler	£175	Grey Matter	01364 654100
Visual Ada	Ada	£480	Grey Matter	01364 654100
WinForth Explorer	Forth	£125	Grey Matter	01364 654100
WinForth Professional	Forth	£345	Grey Matter	01364 654100

Development systems/Prototypers

Product	Description	Price	Supplier	Contact no
Application Framework 4.0	Cross platform developement tools	£530	Zinc Software	0181 8559918
APS for Client/Server	Graphical 4GL	£3500	Intersolv	01707 812812
CA-Visual Objects	Active OO repository based native compiler	£750	Computer Associates	01753 577733
CASE:W Corp.	Code generator	£695	System Science	0171 8331022
CASE:W VIP	Code generator	£1595	System Science	0171 8331022
DataFlex for Win 3.1b/Phase II dev	4GL	£695	Data Access	01923 242222
Enterprise Developer 1.0 Solo	Distributed C/S database development tool	£589	Symantec	01628 592222
Enterprise Developer 1.0 Team	Distributed C/S database development tool	£1319	Symantec	01628 592222
KnowledgePro Windows NT	KPWin for NT	£400	Knowledge Garden	01753 833534
KPWin	OO, event-driven language	£250	Knowledge Garden	01753 833534
KPWin++	KPWin superset, generates C++	£600	Knowledge Garden	01753 833534
Lotus Notes ViP for Windows	Visual programming environment for Notes	£895	Lotus	01784 455445
NewEra development licence	Graphical OO C/S development environment	£3200	Informix Software	01784 240244
Open Insight 2.0	Repository-based development environment	£645	Revelation Technologies	01908 233255
PowerBuilder EE 3.0A	Complete entreprise edition	£3355	Powersoft	01628 34500
PowerBuilder Desktop 3.0	Development env. With ODBC	£695	Powersoft	01628 34500
PowerMaker	C/S development tool	£250	Powersoft	01628 34500
Prominare Development System 2.1	Application development tool generating C++	£695	Microtransfer	01869 350340
Select C++ Generator	Generates Visual C++ code from OMT	£495	Select Software Tools	01242 226553
Synchronicity 2.0	Persistent object mapping and business obj.	£4250	Easel	01344 304595
Systems Engineer 5.0	Client builder	£500	LBMS	0171 6364213
Together/C++ Personal Edition	OO modelling and design tool	£790	Object UK	01703 399990
Together/C++ Team Edition	Multi user version of Together/C++ PE	£2930	Object UK	01703 399990
Visual AppWare Builder	Cross platform visual development tool	£340	Novell	01344 724000
WindowsMAKER Prof	Code generator	£750	System Science	0171 8331022
XVT C Development Solution	C visual application framework	$1950	Personal Wkstation Soft.	0171 2310333
XVT C++ Development Solution	C++ visual application framework	$1950	Personal Wkstation Soft.	0171 2310333
XVT C Development Solution/NT	C visual application framework	$1950	Personal Wkstation Soft.	0171 2310333
XVT C++ Development Solution/NT	C++ visual application framework	$1950	Personal Wkstation Soft.	0171 2310333
Zinc Engine 3.6	Multi platform development tool	£305	Grey Matter	01364 654100
Zinc Windows Activation Key 3.6	Windows (includes NT & Win32s) key	£305	Grey Matter	01364 654100

Visual Basic add-ons - VBXs

Product	Description	Price	Supplier	Contact no
AccuSoft Image Format Library/VBX	Image processing, scanning, special effects	£430	Grey Matter	01364 654100
Aware/VBX	Collection of data aware custom controls	£110	Contemporary Software	01727 811999
CCF Cursors	Create cursors	£30	QBS	0181 9944842
Code Basic 5.1	Access dBase, FoxPro & Clipper databases	£140	Grey Matter	01364 654100
Code.Print Pro	Maintain printed source code	£85	QBS	0181 9944842
Communications Library/Win 2.10	Communication library	£115	Grey Matter	01364 654100
Compression Plus	File compression library	£199	System Science	0171 8331022
Controls for Btrieve 2.0	Database library	£185	Grey Matter	01364 654100
Custom Control Factory for VB	Create custom button controls & toolbars	£40	QBS	0181 9944842
d-Barcode VBX2.2	Barcode management	£94	Grey Matter	01364 654100
Data Widget 1.0A	Custom controls library	£99	Grey Matter	01364 654100
Distinct TCP/IP for Win - SDK 3.2	Custom controls library	£185	Grey Matter	01364 654100
FXTools/VB Professional Edition 1.5	Sundry tools	£260	Grey Matter	01364 654100
FXTools/VB Standard Edition 1.5	Sundry tools	£149	Grey Matter	01364 654100
HighEdit for Visual Basic 2.0	Word processing functions in a VBX	£165	Grey Matter	01364 654100
ImageKnife/VBX 1.3	Image handling	£175	Grey Matter	01364 654100
ImageMan/VB 3.0c	Image manager	£239	Grey Matter	01364 654100
ImageStream/VBX 1.01	Graphic library	£220	Grey Matter	01364 654100
Integra VDB for VB, C/S Edition 2.1	ODBC database control	£509	Grey Matter	01364 654100
Integra VDB for VB, Desktop Ed. 2.1	ODBC database control	£209	Grey Matter	01364 654100

Windows tools for developers

A good workman never blames his tools. A good developer needs good software tools. Here's a guide of over 200 tools right for the job.

Mission impossible: to create a complete directory of Windows development tools. Microsoft used to attempt this daunting task but gave up a long time ago. That didn't stop us trying though. We faced several problems on the way. First, there are so many tools that it would need a whole book, or better, a CD-ROM to publish all of them. So which tools should we include in this guide? To fit as many products as possible we had to be very succinct. Then, how should we present the information? By category, by API supported, by editor, by distributor? We chose the first solution. When there were several configurations, we usually described only one, such as single user license price only. But we still tried to give version numbers.

So this directory, or 'registry', is not comprehensive but it should help you find a product and give you a general idea of what is available - today. Operating systems evolve and development tools with them, so it's a good idea to have a point of reference in this fast moving industry.

The rules for giving the source of a product are as follow. If the product is manufactured in the UK or comes from a company with a UK presence, then that's the contact I have tried to give. If the product has a sole distributor then that's the one listed. Otherwise, I quoted the dealer who first provided me with the information in the clearest form. Apologies to those who were omitted.

Most development tools are available from more than one source and often at slightly different prices, so shop around. Also, prices obtained directly from software editors are almost automatically retail prices which are always quite different from 'street' prices.

Languages - Basic

Product	Description	Price	Supplier	Contact no
CA-Realizer	Structured BASIC visual dev. tool	£99	Computer Associates	01753 577733
Visual Basic Windows 3.0 prof	Visual Basic development environment	£230	Microsoft	01734 270001
Visual Basic Windows 3.0 std	Visual Basic development environment	£96	Microsoft	01734 270001

Languages - C/C++

Product	Description	Price	Supplier	Contact no
Borland C++ 4.0	Complete C++ Visual Dev. Environment	£349	Borland	01734 321150
C/386 Developer	C Compiler	£395	Grey Matter	01364 654100
High C/C++ 386	DOS & Windows compiler and SDK	£555	Grey Matter	01364 654100
Visual C++ 1.0 prof	C++ visual environment	£230	Microsoft	01734 270001
Visual C++ 1.0 std	C++ visual environment	£65	Microsoft	01734 270001
Visual C++ 1.5 prof	C++ visual environment	£299	Microsoft	01734 270001
Visual C++ 2.0 std	C++ visual environment	£300	Microsoft	01734 270001
Symantec C++ Professional 6.1	Standard plus Win32s & MFC 2.0support	£369	Symantec	01628 592222
Symantec C++ Standard 6.1	Integrated development and debugging env.	£99	Symantec	01628 592222
Watcom C/C++ 10.0	Multiplatform optimised C++ compiler	£395	Watcom	0101 519 8863700

Languages - Others

Product	Description	Price	Supplier	Contact no
FORTRAN	32-bit FORTRAN development system	£300	Microsoft	01734 270001
IntegrAda for Windows	Ada compiler	£740	Grey Matter	01364 654100
IntegrAda Win. Dev package	Includes IntegrAda & Visual Ada	£885	Grey Matter	01364 654100
Macros Assembler 6.11	Intel assembly language development set	£130	Microsoft	01734 270001
MPE ProForth for Windows	Forth	£930	Grey Matter	01364 654100
Pascal 7.0	Complete Pascal Development Environment	£299	Borland	01734 321150
PC Logo for Windows	Logo	£60	System Science	0171 8331022
Personal REXX - Windows	REXX	£125	System Science	0171 8331022
RM/COBOL-85 Windows	COBOL	£1045	System Science	0171 8331022
Smalltalk/V for Windows 2.0	Smalltalk	£335	Grey Matter	01364 654100
Smalltalk/V for Win32 2.0	Smalltalk	£905	Grey Matter	01364 654100
Stony Brook Pascal+	Pascal	£279	System Science	0171 8331022

```
static UINT BASED_CODE indicators[] =
{
  ID_SEPARATOR,
  ID_INDICATOR_OVR,
  ID_INDICATOR_CAPS,
  ID_INDICATOR_NUM
};
```

Figure 6 - indicators[] declaration

The BARS application implements the floating palette in PALETTE.CPP and PALETTE.H. The declaration, in PALETTE.H, derives the CPaletteBar class from the CToolBar class. In PALETTE.CPP, we have some `CPaletteBar()` constructor code which initialises the class. The most important code here is that which sets the number of columns for the bar. The `m_nColumns` member variable contains the number of columns which should be displayed in the bar.

We override the `SetSizes()` function so that it accepts a parameter which indicates the number of columns the bar will have. If the user of the class needs to change the number from the default setting of two, coded in the constructor, they should call this function. This calls `RecalcLayout()`, a complex-looking set up that simply recalculates the spacing information to make sure that the bar paints correctly the next time it is drawn. The `SetSizes` call finishes by calling the `CToolBar::SetSizes()` function so that it can do the work necessary to chop up the bitmap of the tools on the bar.

`CPaletteBar` also overrides `SetButtons()`. Like `SetSizes()`. This function also calls `RecalcLayout()` to recompute the size of the buttons, then calls the CToolBar implementation of the function to do the rest of the necessary work.

The brunt of the work in the CPaletteBar class happens during the processing of the `DoPaint()` function. The function draws a frame for the window, which is actually borderless and would be difficult to pick out otherwise. Then, the function builds a false caption.

Since the caption isn't really there, we need a handler for the Windows WM_LBUTTONDOWN message. The handler for this function, implemented in `CPaletteBar::OnLButtonDown()`, checks to see if the mouse pointer is in the region of the false caption. If it is, the function calls `SetCapture()` to be advised of mouse movement as it happens. When the mouse moves with capture to the window, Windows sends WM_MOUSEMOVE messages to the window so it can update its position. While the user drags the CPaletteBar window, the `CPaletteBar::InvertTracker()` function draws a grey frame. The `InvertTracker()` code seems a little complicated, but it just erases and redraws the rectangle for the window as it moves. It does this with several `PatBlt()` calls to achieve maximum performance.

To manage the cursor in the `OnMouseMove()`, `OnLButtonDown()`, and `OnLButtonUp()` functions, the `OnLButtonDown()` function sets the `m_bTrackMove` member variable to TRUE so that the handler for these other messages can react properly. In the case of `OnLButtonUp()`, the flag is checked to see if capture should be released and the window finally redrawn at its new position.

If the user presses the left button over the client area of the window, and not over the false caption area, the `OnLButtonDown()` lets the call fall through to the CToolBar implementation of `OnLButtonDown()` which does the calculations required to locate the click, then activate and repaint the button bitmap.

It turns out that the CPaletteBar class also implements a `HitTest()` member function to override the internal `CToolBar::HitTest()` function. This function is used to analyse mouse clicks to see which button they hit.

```
// toolbar buttons - IDs are command buttons
static UINT BASED_CODE buttons[] =
{
  // same order as in the bitmap 'toolbar.bmp'
  ID_FILE_NEW,
  ID_FILE_OPEN,
  ID_FILE_SAVE,
  ID_SEPARATOR,
  ID_EDIT_CUT,
  ID_EDIT_COPY,
  ID_EDIT_PASTE,
  ID_SEPARATOR,
  ID_FILE_PRINT,
  ID_APP_ABOUT,
};<F255D>

if (!m_wndToolBar.Create(this) ||
   !m_wndToolBar.LoadBitmap(IDR_MAINFRAME) ||
   !m_wndToolBar.SetButtons(buttons,
                  sizeof(buttons)/sizeof(UINT)))
   {
    // error TRACE
   }
```

Figure 7 - Toolbar initialisation

Be lazy

This article has examined many of the more interesting, advanced Microsoft Foundation Classes which can really spruce up your application (although we have not touched many others such as ODBC or OLE support).

By using the features implemented by these classes, and other classes in the MFC library, you will find that implementing your own classes in Visual C++ is significantly easier when compared to reinventing all of this functionality in C. You can often 'rob' functionality that has already been implemented. You should try to gain as much as possible from the existing classes: let's make life easier for ourselves.

This article was taken from a presentation that was given at MS TechEd 94. Mike Blaszczak is a Software Design Engineer in the AFX Group at Microsoft and works on the OLE 2.0 Custom Control Development Kit. He spends some of his time writing technical books and articles.

ID_value	indicator
ID_INDICATOR_EXT	extended selection indicator
ID_INDICATOR_CAPS	cap lock indicator
ID_INDICATOR_CAPS	num lock indicator
ID_INDICATOR_SCRL	scroll lock indicator
ID_INDICATOR_OVR	overtype mode indicator
ID_INDICATOR_REC	record mode indicator

Figure 5 - Special, automatic indicators

ing may take place. If the flag is TRUE, the status bar will be immediately updated. If the flag is FALSE, changes to the text will be made but the pane will not be invalidated after the update, deferring the painting until later. The FALSE setting is appropriate when more than one pane will be updated at a time. You can make your screen updates faster and more appealing by calling the first updates with FALSE settings and TRUE with the final change.

You may wonder how the status bar updates the indicators for the overwrite, NumLock, and CapsLock keys. It turns out that the ID_values for these indicators are special; they're predefined in AFXRES.H. The status bar recognises them and appropriately initialises and updates the content of the fields throughout the life of the status bar (see Figure 5).

While this method provides a high value, it has the side effect of requiring you to provide string resource IDs exactly matching the particular AFX-defined ID_ number in your resource file. Because some of the indicators do not directly reflect the status of the machine's hardware (like the record mode indicator or the extended selection indicator), the programmer must provide callbacks which the status bar can use to interrogate these values.

The Toolbar

Many modern applications implement toolbars, which allow the user to select a command by clicking on a pushbutton with a graphical representation of the command. The toolbar remains on the screen at all times (though the user may remove it to increase the amount of space available to the application in its client area). Some toolbars are more robust, including controls like drop-down list boxes or spin controls to allow the user to select or change values from a list or within a range.

The Microsoft Foundation Classes offer support for toolbars via the CToolBar class. When you include a toolbar in your application, AppWizard adds code to create it as your application creates its main frame window. The MAINFRM.CPP module contains the `CMainFrame::OnCreate()` function, which initialises the status bar and toolbar windows for the application.

The CMainFrame class, as defined in MAINFRM.H, has an instance variable named m_wndToolBar which is a class CToolbar. As with other window types in the Microsoft Foundation Classes, this class contains the variables and functions necessary to manage a toolbar. But creating an instance of the class does not create a visible instance of the toolbar. Instead, the application must call `m_wndToolBar.Create()` to create the window. This function accepts a parameter: a pointer to the CWnd-derived parent where the toolbar will live.

One toolbar, one bitmap

If the `Create()` function is successful, the toolbar is ready to draw itself and place itself within the client area of the main form. It continues by loading the bitmap. Each toolbar is driven by a single bitmap; this conserves space and resources within Windows. At default setting the button faces are 16 pixels high by 15 pixels wide. If your buttons are a different size, you will need to call the `SetSizes()` function to make sure that the bar is measured and painted correctly.

Finally, to complete the initialisation of the toolbar, the application calls `SetButtons()`. This parameter takes a pointer to an ordered list of command IDs, which will associate each button with a menu command that the button will fire when pressed. The array can also include ID_SEPARATOR entries to put a small gap between groups of buttons. These gaps are not present in the resource bitmap used to draw the buttons.

The array created by the AppWizard for the default button bar and the toolbar initialisation code within `CMainFrame::OnCreate()` are shown in Figure 7.

Unfortunately, toolbars are relatively limited. In the current version of the Microsoft Foundation Classes, toolbar buttons can't be disabled. As a result, command-handling code needs to make sure that the action the user requested can indeed be carried out.

The stock version of toolbar used here is handy for creating toolbars which are anchored to the window they are in, and which contain only pushbutton controls. A more complicated class, CDialogBar, described next, can be used to create more robust toolbars which can host any type of Windows control.

Dialog bars

Dialog bars are implemented by a class called CDialogBar. They are similar to control bars in that they live within a frame window and paint themselves within the confines of the window, but they also provide a host for a dialog window. That is, if you wish to place controls other than pushbuttons in a control bar, you can do so by using a CDialogBar instance.

The sample application uses a CDialogBar to place a drop-down list box and some check boxes at the left edge of the client area in the main window. The code simply calls CDialogBar's `Create()` member function, passing it a reference to the parent window, the template name of the dialog resource to be created, and a control ID for the bar. The function also accepts a style word which determines the alignment of the bar within the window. An overloaded version of the `Create()` function exists which accepts a resource ID rather than a resource name.

When examining the application, you will quickly notice that there is a second toolbar in the client area of the window which contains pushbuttons, but also has a combo box. This control is created by `CMainFrame::CreateStyleBar()` function, and is dynamically sized and placed in the bar after it is created. This is the obvious disadvantage to using a CToolBar class when you have controls: each one must be dynamically placed and measured. CDialogBar allows you to create all of the items from a normal dialog box template in a resource file.

Make it float!

While the Microsoft Foundation Classes don't directly support floating tool palettes, the CToolBar class can easily be subclassed to provide the functionality that we need. It turns out that the code which manages the bitmap within the toolbar can be modified to paint more than one row of controls. C++ language lets us use inheritance to reasonably manage all of our changes.

of message maps. While the algorithm in the function is centralised around a linear lookup in the list of AFX_MSGMAPs, a simple cache mechanism makes the most recently used messages more readily identifiable. Further, non-portable versions of the library have a short hand-tuned assembly language routine in place to do the lookup faster.

Command messages differ from other Windows messages most predominantly in that they return only a TRUE or FALSE Boolean code to the caller. Command messages also don't call a default procedure when they are not processed by the application. These differences aside, the `OnCmdMsg()` function implemented in CMDTARG.CPP is largely the same as its counterpart in WINCORE.CPP.

Easy maps

Examining the message map and command routing architecture at this level is rarely necessary. It is, of course, important to understand how to use the macros and functions in question. But our tour through the mechanical aspects of the process is really best left to the library.

Best of all, even declaration and maintenance of message maps isn't a requirement for programming in the Microsoft Foundation Classes. The ClassWizard features the ability to construct, modify and view message maps. The funny comments (with double slashes and double curly brackets) serve as markers for the ClassWizard as it processes the code in your application to find the maps. Be careful when modifying your code, as disrupting these markers can confuse the ClassWizard!

Status and symbols

It seems like every application these days has a status bar and a toolbar. Some even have floating tool palettes, allowing the user to place the icons they're working with on an unused portion of the screen. Because the Microsoft Foundation Classes form an application framework rather than a class library, classes which implement these kinds of windows are readily available in the MFC libraries.

Our sample application, ToolBox, contains a simple status bar, a toolbar and a floating tool palette. While each of these features are implemented almost completely within the class libraries, let's look at how we should use the correct classes to get the job done. It's worth noting that ToolBox weighs in at less than 1,000 lines of Visual C++ source code, an indication of exactly how much work the MFC classes are doing. If you have written toolbar or status bar code for Windows in C, you know that doing the maths to measure sizes, manage fonts and store information is a considerable amount of work!

Data type	Member name
UINT	nMessage
UINT	nID
UINT	nSig
AFX_PMSG	pfn

Figure 3 - AFX_MSGMAP_ENTRY

nSig Value	Return type	Parameter list
AfxSig_vv	void	void
AfxSig_bD	BOOL	CDC*
AfxSig_is	int	LPSTR
AfxSig_vbWW	void	BOOL, CWND*, CWND*

Figure 4 - nSig common values

The status bar at the bottom of ToolBox's main window is managed by the CStatusBar class. An instance of this class is created very early in the application's life, after the CMainFrame window is created. Just like every other Microsoft Foundation Class which implements a window, there is a big difference between creating an instance of the object which will manage the class and creating the actual window on the screen. The CStatusBar class is no different.

The CStatusBar data member of the CMainFrame class, as declared in the MAINFRM.H header, is called `m_wndStatusBar`. While many nuns have made it a habit to wear white and black, many Visual C++ programmers have made it a habit to begin names of member variables with '`m_`'. This readily identifies the names as being members variables, discerning them from global variables. It further avoids collisions between the names of member variables and the names of function parameters which is often a problem when writing constructors. The `CMainFrame::OnCreate()` function calls the `CMainFrame::CreateStatusBar()` function to do the actual work involved in creating the status bar. So, the in-memory copy of the class is created whenever the user creates an instance of the application.

`CreateStatusBar()` first calls the `CStatusBar::Create()` function, passing a reference to the CMainFrame object. This actually creates the status bar window and lets it know who the parent window is. At this point the status bar is ready to paint itself: it will measure the parent window and place itself at the bottom.

The `CreateStatusBar()` function continues by calling `SetIndicators` and passing the `indicators[]` array to the function. The `indicators[]` array is declared at the beginning of the MAINFRM.CPP module as shown in Figure 6.

Let the pane take the strain

These precompiler constants tell the status bar that there will be four panes in the status bar. The first pane will effectively be blank. The second pane will be the 'overwrite' indicator, which will be used to show the status of the Insert key. The 'Caps' indicator shows the status of the CapsLock key, and the final 'Num' indicator shows the status of the NumLock key. Each of the ID_INDICATOR constants matches a string resource ID which contains the text to be displayed when the indicator is active. The CStatusBar class erases the text when the status is inactive.

The `CreateStatusBar()` function sets up the balance for the space in the status bar by calling `GetPaneInfo` for the first pane of the status bar, like this:

```
m_wndStatusBar.GetPaneInfo(0,
nID, nStyle, cxWidth);
```

This loads the nID, nStyle, and cxWidth functions with id, style, and width (respectively) of the first pane: which I declared to be a separator. The `CreateStatusBar()` function concludes its work by changing the style of the first (that is, the zeroth) pane: `m_wndStatusBar.SetPaneInfo(0, nID, SBPS_STRETCH, cxWidth);`

This causes the first status bar pane to stretch out to fill the available space in the status bar, effectively making subsequent panes right-justified. Other styles include disabled, pop-out (the 3D borders go 'backwards' making the pane appear to 'rise' from the bar), and SBPS_NOBORDERS, which makes the pane borderless and at a level with the rest of the bar.

Update now, paint later

Throughout the application, we may set the text in the status bar's first pane by calling `m_wndStatusBar.SetPaneText()`. This function takes the index of the pane we are interested in, the text we wish to set, and a flag to indicate when the paint-

that we saw in Figure 2 to indicate the end of the message map. The values all indicate what the message handler function, generically pointed to by the pfn element, will be passed as parameters and what it will return as a result code. Common values are listed in Figure 4.

The full realm of signal values and message map declaration functions can be found in the AFXMSG_.H header file.

Six ways to code...

The AFX_MSGMAP_ENTRY is used to send every conceivable type of message and can contain information about a wide variety of events.

For *normal control notification messages*, such as a the parent of a list box being sent a LBN_SELCHANGE message, the nMessage field contains the notification code (for example, LBN_SELCHANGE). The nID field contains the control ID, and the signature type is AfxSig_vv since the LBN_SELCHANGE message handler takes no parameters and returns nothing.

The member function which handles the LBN_SELCHANGE message is pointed to by the pfn element, and would be declared as `afx_msg void OnChangeSelection(void);` to match the nSig value. Of course, the function name is user-specified in the `ON_COMMAND()` declaration for the message.

For *update command UI messages*, which are sent when a menu is to be updated, the nMessage element is -1, the nID function identifies the menu item and the signature type is AfxSig_vv. The member function would have the same declaration: `afx_msg void OnUpdateUI(void);` to match the nSig value.

Messages sent when a *menu item or a keyboard accelerator is activated* include the nMessage field set to zero. The nID is the identifier of the accelerator, while the signal in nSig is always AfxSix_vv. The pfn pointer must point at a function declared as `afx_msg void HandleItem(void);`

Predefined Windows messages are sent with the nMessage element set to the constant which names the message, while the nID entry is always zero. The signature type is dependent on the message being processed. For example, WM_SETCURSOR is AfxSig_bWww requiring a `afx_msg void BOOL(CWnd *, UINT, UINT);` handler declaration, while WM_PAINT is AfxSig_vv and declared as: `afx_msg void OnPaint(void);`

Registered Windows messages are created with the Windows RegisterWindowMessage() API. They require special handling since the messages are user-defined, and the handler is a special function.

Variable Control Notification messages set the nMessage value to the notification message, the nID value to the control id, and the nSig field to the AfxSig_vv constant. The function points to the handler for the message. Variable Control Notification Messages are required by VBX-style custom controls.

`END_MESSAGE_MAP()` simply closes the message map's punctuation. It also marks the end of the message map by creating an easily recognisable null entry in the list of message map entries. This allows the functions which traverse the message map to recognise quickly when they have finished message handling.

Getting the message?

We've discussed how the message map is built: how the application enters a message loop which eventually begins pumping messages to the rest of the application. However, until now, we have not examined what happens when a message is actually processed.

The default window procedure given to Microsoft Foundation Class windows, inspired by CDialog or CWnd, includes its own `WindowProc` function. This function dispatches messages to the appropriate command function and is therefore heavily reliant on the AFX_MSGMAP structures just described. The default window procedure starts out by checking to see if the received message is a WM_COMMAND message. If it is, the `WindowProc` function passes control immediately to a function called `OnCommand()` which checks to see if the ID passed to the command window is a valid child window. If it is, it handles the parent notification message passing itself. Otherwise, since the object receiving the message is a command target, the function calls the appropriate `CCmdTarget::OnCmdMsg()` member function for the object. The return from the `OnCmdMsg()` function is passed back to Windows as a result of the message procedure. This way, the developer can return whatever information is necessary to Windows when the message is processed.

When the message is not a command, the procedure executes by looking at all of the entries in the class message map, which is found by calling the `GetMessageMap()` function implemented by the `DECLARE_MESSAGE_MAP()` macro discussed earlier. If the message is not found in the map, the pointer to the message map in the parent class is retrieved and that table is examined. If this process continues to the base class and no message mapping is found, the `WindowProc()` function calls `DefWindowProc()` to handle the message.

Unbeatable portables

Once the actual address of the routine is resolved, the function encounters a huge case statement which dispatches the proper parameters and function calls to get the message processed. The case statement is a simple jump table, so it executes very quickly. The beauty of this mechanism, however, is that it takes care of decomposing the message's low-level parameters. This means that the library itself is responsible for accounting for the differences between Win32 and Win16 message parameters. Essentially because of these fixes, Windows programs based on the Microsoft Foundation Classes are inherently more portable than applications which are not built around the classes. And, soon enough, implementations of the MFC and Visual C++ itself will exist on non-Intel platforms.

Looking up a message function in the message maps for every single call can be very time consuming, especially for deeply nested classes or those which involve thick hierarchies. The WindowProc has to make sure that processing doesn't become bogged down with the overhead

```
#define BEGIN_MESSAGE_MAP(theClass, baseClass) \
   AFX_MSGMAP* theClass::GetMessageMap() const \
        { return &theClass::messageMap; } \
   AFX_MSGMAP AFXAPP_DATA theClass::messageMap = \
   { &(baseClass::messageMap), \
     (AFX_MSGMAP_ENTRY FAR*) (theClass::_messageEntries) }; \
   AFX_MSGMAP_ENTRY BASED_CODE \
      theClass::_messageEntries[] = \
                {
#define END_MESSAGE_MAP() \
        { 0, 0, AfxSig_end, (AFX_PMSG)0 } \
        };
```

Figure 2 - AFXWIN.H

Advanced MFC Architecture

Now that you've switched to C++, Mike Blaszczak explains how to use the MFC efficiently.

The Microsoft Foundation Classes provide functionality for the developer with the incarnation of utility classes for collections and abstractions of Windows graphical objects. They also embody a large number of user interface components which can be used directly, or modified and reused.

By studying and understanding the implementation of the classes, you can come to understand the 'philosophy' behind their design. This understanding will make your programs fundamentally better for several reasons. You will not need to develop more code than you should, since you can inherit the functionality in the classes. While saving you work, this will also inherently improve your overall code quality. Since many components of your user interface will come from the foundation classes, generally your application will fit the Windows user interface standard more closely. Further, your code will work more harmoniously with the rest of the classes, thereby increasing your efficiency relative to someone who doesn't fundamentally understand the libraries.

A note about the sample code

The code for this article is based heavily on the CTRLBARS sample which ships with Visual C++ Version 1.5. (If you did a default installation, you should find this sample in \MSVC\MFC\SAMPLES\CTRLBARS.)

Pass the message

Most developers know by now that Windows programs work by processing messages sent to the application by Windows. The messages describe various actions or tell applications that Windows needs some kind of action performed.

Messages, academically speaking, are a request from one object for another object. Object A might request that object B return its current cash value, for example. Messages are really nothing more, in object-oriented terms, than the act of one class requesting another class to execute a member function. While C++ implements member functions, the way Windows expects things to work is a little different. In order to help make C++ mesh more closely with Windows, the Microsoft Foundation Classes provide a mechanism to handle messages cleanly.

After a C++ program is loaded into memory and all of the static objects it contains are constructed, control flows to the application's `Run()` function. Eventually, after performing some amount of initialisation, the `Run()` function calls another function called `PumpMessage()`.

This function grabs a Windows message using the familiar Windows `GetMessage()` API. It then sorts through the tree of windows, from the target up to the desktop, to see if there is any work to be done on the message before it is translated. This is implemented in the `PreTranslateMessage()` function of the CWnd class. By default, this feature does absolutely nothing; however, the implementer of any CWnd-derived class is welcome to override the function to perform message checking. This may swallow the message, thereby short-circuiting its processing, or it may allow the message to pass through the normal channels.

Message mapping

This is all fine and good, but it sounds terribly like normal message processing in a Windows application. In C++ programs for MFC we know we have to make a message map which relates a function in a class to a windows message or command message. How does a code like the one in Figure 1, seen so often in MFC applications, ever become invoked?

To answer this question, let's first dissect exactly what is being done. We can guess that `BEGIN_MESSAGE_MAP()`, `ON_COMMAND()`, and `END_MESSAGE_MAP()` are actually preprocessor macros, and are declared in AFXWIN.H and AFXMSG_.H. The code for `BEGIN_MESSAGE_MAP()` and `END_MESSAGE_MAP()` is in Figure 2. These header files can be found in the \MSVC\MFC\INCLUDE directory of a standard MS Visual C++ installation.

So, when we declare a message map for CBarsApp (which derives from CWinApp), we're actually building a message map data structure and initialising it. The `BEGIN_MESSAGE_MAP()` also declares a member function for the class which returns a pointer to the message map implemented in the class.

Message maps are threaded to each other via the first entry in the list, which is defined and initialised by the `BEGIN_MESSAGE_MAP()` macro.

The data structure in question is the type AFX_MSGMAP, which is defined in the AFXWIN.H header. This type contains two pointers. The first is a pointer to the base class message map, the second is a pointer to the array of message map entries. This enables the framework to pass messages to the base class if the target class does not map the message.

The array of message map entries is of type AFX_MSGMAP_ENTRY. This data type is also a structure. The description of this structure is in Figure 3.

The nMessage entry holds the value of the message ID; for example, WM_PAINT for a paint message. The nID field contains the ID of the control in the window which will receive the message. If the Windows message is not intended for a control then it will contain a 0.

The nSig element is very interesting: it determines what type of function call must be made. There are about 40 possible values, including the special AfxSig_end

```
BEGIN_MESSAGE_MAP(CBarsApp, CWinApp)
    //{{AFX_MSG_MAP(CBarsApp)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

Figure 1 - MFC message map

```
Sub cboIndexList_Click ()
  tblObject.Index = cboIndexList.Text
  PopulateFields
End Sub

Sub vsbInnerScroll_Change ()
  pnlInner.Top = -vsbInnerScroll.Value
End Sub

Sub Form_Resize ()
  pnlStatus.Top = pnlStatus.Parent.Height _
                - pnlStatus.Height _
                - 390
  pnlWindow.Height = pnlStatus.Parent.Height  _
                   - pnlHeader.Height _
                   - pnlStatus.Height  _
                   - 450
  SetHeightOfInnerPanel
End Sub

Sub cmdChangeDb_Click ()
  tblObject.Close
  dbObject.Close
  GetDataBaseNameFromUser
  Set dbObject = OpenDatabase(msDbName)
  FillTableList
  cboTableList.ListIndex = 0
End Sub

Sub cmdNext_Click ()
  If Not tblObject.EOF Then tblObject.MoveNext
  If Not tblObject.EOF Then
    PopulateFields
    pnlMessage = ""
  Else
    pnlMessage = "No more records"
  End If
End Sub

Sub cmdPrevious_Click ()
  If Not tblObject.BOF Then tblObject.MovePrevi-
ous
  If Not tblObject.BOF Then
    PopulateFields
    pnlMessage = ""
  Else
    pnlMessage = "No more records"
  End If
End Sub

Sub cmdExit_Click ()
  On Error Resume Next
  tblObject.Close
  dbObject.Close
  End
End Sub
```

Figure 4 - A few more experts

`cmdChangeDb_Click` command button reproduces the common dialogue code from `Form_Load`.

The next and previous buttons will allow navigation from record to record. Note that it is necessary to check the `BOF` or `EOF` status before and after moving. If the record pointer is already at the end of the file then the `MoveNext` method will generate an error.

After the move next, if `EOF` is true, then any reference to the Field objects will also result in an error. The `Form_Resize` event makes sure that at least height-wise the display expands or contracts correctly. Finally the `cmdExit_Click` button closes the database and table tidily before the application ends.

Collections in less than 160 lines

And that's all. In 153 lines of code we have an application that can display data from any Microsoft Access database; it can be tested on the BIBLIO.MDB database supplied with VB (see Figure 5). Of course it might be a good idea to tighten up on the error handling, improve on the formatting of numeric fields, perhaps adjust the width of the display controls according to the actual contents of each field and allow for password protected databases. But it does serve to illustrate the use of objects and collections within VB.

The next step is to control objects in other applications from within VB. Have you ever considered programming a print preview utility as part of a report option? Just send the report to Word and use the function there. Have you attempted to provide even simple spreadsheet functions in VB? Excel has the function and your application can remain in control.

Whilst this might sound a little too much like evangelising for Microsoft, the technology and techniques are here and now, and there are a growing number of third party products that offer the same OLE2 functionality. If you are using Visual Basic you are about to be put in control of it all.

Steve Moulton is a Solution Developer at Datasure Ltd. A software house developing applications for the insurance market. If you would like the code from this article please send a disk and SAE to EXE Magazine marked 'DBAccess'.

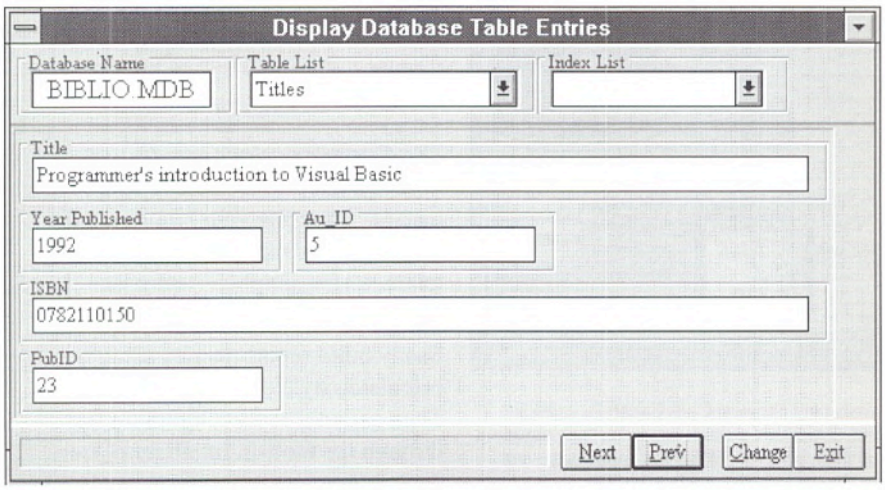


Figure 5 - The Access database viewer application

```
Sub cboTableList_Click ()
mnColumn = 0
mnRow = 1
FillIndexList
For i = 0 To dbObject.TableDefs _
 (cboTableList.Text).Fields.Count - 1
  msFieldName = dbObject.TableDefs _
    (cboTableList.Text).Fields(i).Name
  mnFieldType = dbObject.TableDefs _
    (cboTableList.Text).Fields(i).Type
  mnFieldSize = dbObject.TableDefs _
    (cboTableList.Text).Fields(i).Size
  AddNewFieldToDisplay (i)
Next i
SetHeightOfInnerPanel
If dbObject.TableDefs(cboTableList.Text). _
   Fields.Count < mnNumOfFields Then
  For i = dbObject.TableDefs(cboTableList.Text). _
   Fields.Count To mnNumOfFields - 1
    fraField(i).Visible = False
    txtField(i).Visible = False
  Next i
End If
mnNumOfFields = dbObject.TableDefs _
  (cboTableList.Text).Fields.Count
Set tblObject = dbObject.OpenTable(cboTableList.Text)
tblObject.Index = cboIndexList.Text
PopulateFields
End Sub

Sub FillIndexList ()
  cboIndexList.Clear
  For i = 0 To dbObject.TableDefs(cboTableList.Text). _
    Indexes.Count - 1
    cboIndexList.AddItem dbObject.TableDefs _
     (cboTableList.Text).Indexes(i).Name
  Next i
End Sub

Sub AddNewFieldToDisplay (nFieldNumber As Integer)
On Error GoTo ErrorLoading
  Load fraField(nFieldNumber)
  Load txtField(nFieldNumber)
  If (mnFieldSize > 18 And nFieldNumber > 0) Or  _
     mnColumn = 3 Then
    mnColumn = 1: mnRow = mnRow + 1
  Else
    mnColumn = mnColumn + 1
  End If
  If mnFieldSize < 18 Then
    fraField(nFieldNumber).Move  _
      30 + ((mnColumn - 1) * 2490), _
      15 + ((mnRow - 1) * 645), 2400
    txtField(nFieldNumber).Move  _
      150 + ((mnColumn - 1) * 2490), _
      240 + ((mnRow - 1) * 645), 2100
  Else
    fraField(nFieldNumber).Move  _
      30 + ((mnColumn - 1) * 2490),  _
      15 + ((mnRow - 1) * 645), 7380
    txtField(nFieldNumber).Move  _
      150 + ((mnColumn - 1) * 2490),  _
      240 + ((mnRow - 1) * 645), 7080
    mnColumn = 3
  End If
  txtField(nFieldNumber).ZOrder
  fraField(nFieldNumber) = msFieldName
  fraField(nFieldNumber).Visible = True
  txtField(nFieldNumber).Visible = True
  Exit Sub
ErrorLoading:
  If Err = 360 Then  ' control is already loaded
    Resume Next
  Else
    msg = "Unexpected error whilst loading  _
          new control "  & Error(Err)
    MsgBox msg, 48, "Loading controls"
    Exit Sub
  End If
End Sub

Sub SetHeightOfInnerPanel ()
  pnlInner.Top = 15
  pnlInner.Height = 45 + (645 * mnRow)
  If pnlInner.Height > pnlWindow.Height Then
    vsbInnerScroll.Max = pnlInner.Height  _
      - (pnlWindow.Height - 30)
    vsbInnerScroll.Height = pnlWindow.Height
    vsbInnerScroll.Top = pnlWindow.Top
    vsbInnerScroll.Value = 0
    vsbInnerScroll.Visible = True
  Else
    vsbInnerScroll.Visible = False
  End If
End Sub

Sub PopulateFields ()
  For i = 0 To mnNumOfFields - 1
    txtField(i) = IIf(IsNull(tblObject(i)), " ", tblOb-
ject(i))
  Next i
End Sub
```

Figure 3 - Get the filed's content

the inner panel can be displayed, which it does in **SetHeightOfInnerPanel** by extending the height as necessary. Scroll bars properties are set according to the relationship between **pnlInner** and **pnlWindow**. This routine is also called from the **Form_Resize** event.

The actual piece of code that makes it all worthwhile, and probably the simplest element of the application is **PopulateFields**.

Having established what the table contains and prepared a control for each field, a quick loop through the Table object's Fields collection will populate the controls. It is important to allow for null entries from the database.

Rather than use the full syntax for the collection and object's statement I have chosen to use the most abbreviated one. In common with VB controls, most collections and objects have a preferred or default property. The default property of a field object is its value property, the default collection of a table object is its field's collection. So, **tblObject(i)** could equally have been written as **tblObject.Fields(fraField(i).Caption).Value** but why bother wearing out fingers unnecessarily?

Just a few more events

All that's left is the code for a few simple events (see Figure 4). The **vsbInnerScroll_Change** event enables the main data panel to scroll up and down. It does this by setting **pnlInner**'s top property to the negative of the scroll value so that as the scroll bar's value increases, the top of the inner panel gradually moves upwards. Because it is wholly contained within **pnlWindow**, it effectively disappears at the top and reappears at the bottom. The **cboIndexList_Click** routine resets the index of the currently selected table while

Objects and Collections

On the road to OLE2? Steve Moulton leads the way towards grasping the concept of objects and collections in Visual Basic.

Visual Basic is gaining an increasing presence within the software development world. Microsoft has introduced its Visual Basic Applications edition in Excel 5, Project 4 and has been incorporated into the 'Office' suite. Visual Basic's full potential, as a language for producing integrated desktop applications, can now begin to be realised.

OLE2 functionality is being included in all areas: VB3 was the first Microsoft tool to be OLE2 enabled. At the time there was very little for an application to interface with and so this facet was largely ignored. Times change and increasingly OLE2 objects will become available in much the same way as custom controls (VBXs) have proliferated. OLE2 automation, whereby one application controls another, relies heavily on the concept of an OLE2 object having one or more collections of child objects which might in turn have their own collections. Properties and methods of the object application are controlled by reference to those collections.

Thus, the code `XLObject.Workbooks(1).Worksheets("Sheet2").Range("A1")` in VB refers to the first cell in the worksheet named Sheet2 in the first workbook of the instance of Excel to which a VB application has previously connected. Simple? Well yes, in fact with a little practise it is, and the principle of using collections can be illustrated without having to add the complication of OLE.

Instead, we will put together a simple application using the database object and its child collections that will allow it to open and display any Access database without knowing in advance what the database contains. Of course you could use Data manager to perform the same function but as a bonus you will get to know the details of accessing a database from within VB, dynamically generating controls at runtime and creating a scrollable window within an application, all within 160 lines of code!

The demonstration application requires VB Professional edition version 3.

Of window's sizes

The demonstration application has a tool bar containing `lblDbName` to display the database's name, `cboTableList` a drop down list box for the names of the tables and `cboIndexList` for the indexes of the current table. Each of these is contained in a frame to provide a field caption. There is also a common dialog box, `dlgGetDbName`, that is invisible whilst running.

In 153 lines of code we have an application that can display data from any MS Access database

The form's main area is filled with two panels, one inside the other. It's important to create these correctly to enable the window to scroll. Draw `pnlWindow` first on the form to fill most of the central area, height 2655, width 7485, leaving a little space for the scroll bar. The panel should be inset. After that, `pnlInner` is drawn inside the first panel and is initially positioned left 15, top 15 and sized 30 smaller than the panel. Then a frame (`fraField`) and text box (`txtField`) are drawn onto the second panel.

Position is not important but the index on each must be set to zero to create a control array. This will enable the application to load as many text boxes as there are fields in the database table. The height of the frame is 645 with visible property set to false. Finally, four command buttons and a message panel are added to the form's bottom and the scroll bar is placed next to `pnlWindow`.

The database and table objects are declared using the special type declarations (see Figure 1), the other variables should be self explanatory; their uses are detailed below.

Populating the fields

Until we get some user input there is no information on how the fields are to be populated so the `Show` method is used to paint an empty form immediately followed by a call to the `GetDataBaseNameFromUser` routine. This makes use of the VB common dialog control; an extremely useful utility that saves having to write many lines of code.

As can be seen in Figure 2, all that is required is to pass a title for the dialog box, set a filter so that it only looks for files we are interested in and send an action code of one for file open. The flags option enables Help functions which are available for free, so might as well be used.

The utility routine returns both full path and title of the selected file. `Form_Load` then opens the database, fills the list of available tables by calling the `FillTableList` routine and calls the click event of the table list by setting the ListIndex property. This in turn loads and populates the controls containing the data.

```
Dim dbObject As database, tblObject As table
Dim msDbName As String, msFieldName As String
Dim mnFieldType As Integer, mnFieldSize As Integer
Dim mnNumOfFields As Integer
Dim mnColumn As Integer, mnRow As Integer
Dim i As Integer

Const DB_SYSTEMOBJECT = &H80000002
```

Figure 1 - Objects declaration

VISUAL BASIC ADD-ONS

Royalty-free VBXs from Bits Per Second developers of the GRAPH.VBX in VB Pro

GraphicsServer

If you are still using the GRAPH.VBX in VB Pro, you are missing out on Graphics Server 2.50

- 'Hot' graphs for EIS-style interaction
- Bubble, tape and 3D area graphs
- Bitmap and metafile backdrops
- Printing into the VB device context
- Additional X-axis control (min, max etc)
- 'Null' data not treated as zero
- And a simplified run-time array access

Coming after Xmas - Graphics Server 3.0

Interactive toolbar to access the property pages at design and run-time, **Rotatable True3D** so you can view your graphs from any angle, **Real-time graphing** with more than one data-set in a scrolling window, and an **OLE Custom Control** - future-proof when Visual Basic goes 32-bit.

ToolThings

10+1 productivity tools

Improved forms creation

Tools for interactively designing, sizing, positioning and ordering those controls.

Better code management

Tools that comment or un-comment blocks of code, indent lines, remove unused routines and find user-defined functions and global/constant declarations.

Re-use existing code

CodeBrowser searches external source code files, selects the fragment you specify and inserts it into your document. Re-using code is fine in theory but no use if you can't find it - now it's easy and fast. Plus tools to generate forms, message boxes and common Windows dialogues.

CodePrint.Pro

The printing tool

Print what you want

- Print from within the VB environment
- Print only files that have changed
- Print MAK files without running VB
- Print highlighted sections
- Sort first, or print to a file

Print how you want it to look

- Index forms, objects and sub-routines
- Select distinctive fonts for code and comments
- Take advantage of bold, italic etc
- Use headers and footers
- Print code with line numbers
- Set margins and spacing
- Create a consistent style

And print when you want

Don't hang around waiting for a free printer. Create batch files of one or more print jobs and print them after hours or at the weekend.

HighEdit

Integrated word processing

HighEdit is a fully featured programmable word processor, a VBX and DLL with true WYSIWYG

- Margins and tabs
- Search and replace
- Word-wrap
- Headers and footers
- Support for all printers and TrueType fonts.

VBX Properties

High Edit has over 30 properties that can be configured and altered at design and run-time.

Data-aware

Both text and formatting can be bounded to the Access engine in Visual Basic.

Dialog Boxes

Open and save, search, print, change font etc, all embedded into your application

"Bits Per Second has an excellent reputation for supporting its users, communicating with developers and keeping its promises"

PC Direct

BITS PER SECOND

To find out more about these products

CALL
01273 727119

FAX
01273 731925

NOW SHOWING AT A SCREEN NEAR YOU.

software DEVELOPMENT 94
November 22-24
NEC Hall 7
Birmingham

ACHIEVING SOLUTIONS

COME TO STAND 202.

Developers talk to their screens. We listen. The Microsoft Visual C++™ development system, version 2.0, is the answer to your toughest demands and critiques.

You get the new 32-bit development environment, the award-winning 16-bit Visual C++ 1.5 and the new OLE control development kit. A built-in path ports your 16-bit MFC-based