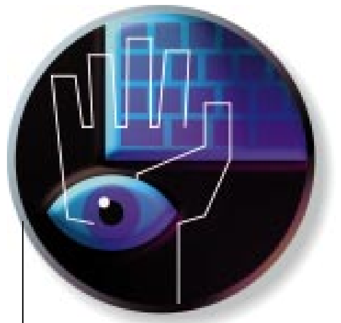


Future **threat**

What's behind Visual J++? Might it overtake C++ as a mainstream Windows application? Tim Anderson talks to Microsoft. Plus, screensavers in Delphi, and a free MSDN sample.

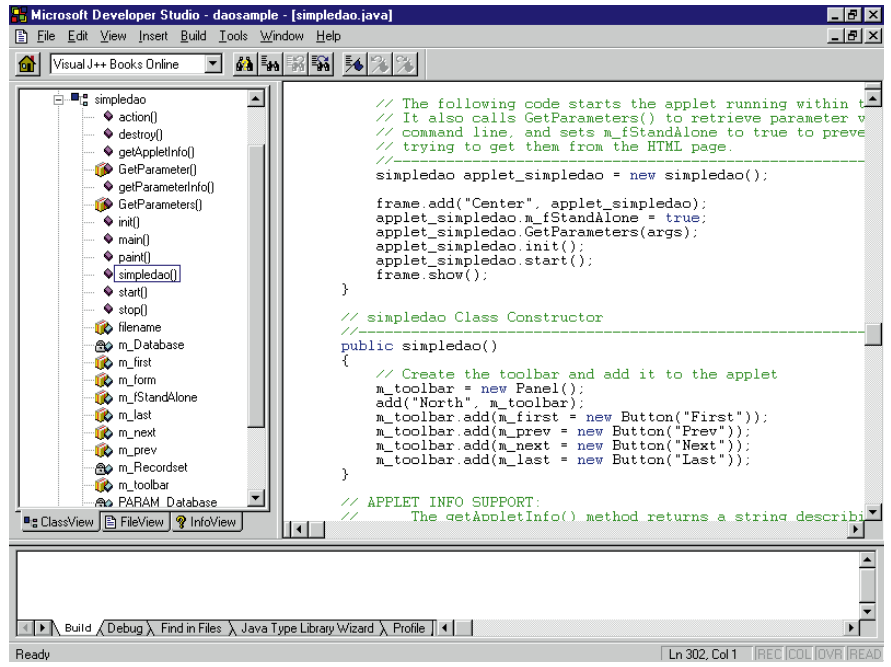
Microsoft's Visual Java tool is now out (see our review in "First Impressions"). But what is the company doing with a language that threatens Windows desktop dominance? Another consideration is that Sun's proposed Java Beans component model is at odds with Microsoft's ActiveX strategy.

I interviewed Microsoft's development manager, Greg DeMichillie, to find out more.

Visual J++

PCW: *I was disappointed by the lack of a visual environment for building an interface.*
DeMichillie: "I totally agree. Our long-term direction *is* towards graphical interface builders but the reason we haven't got this is because of work going on with class libraries. The question is whether the Abstract Window Toolkit is the long-term windowing model, or whether it will it be an alternative? I'll be back in less than a year talking about a new version of Visual J++."

DeMichillie: Aiming for large-scale development in Visual J++



```
        // The following code starts the applet running within t
        // It also calls GetParameters() to retrieve parameter v
        // command line, and sets m_fStandAlone to true to preve
        // trying to get them from the HTML page.
        //----------------------------------------------------------
        simpledao applet_simpledao = new simpledao();

        frame.add("Center", applet_simpledao);
        applet_simpledao.m_fStandAlone = true;
        applet_simpledao.GetParameters(args);
        applet_simpledao.init();
        applet_simpledao.start();
        frame.show();
    }

    // simpledao Class Constructor
    //--------------------------------------------------------------
    public simpledao()
    {
        // Create the toolbar and add it to the applet
        m_toolbar = new Panel();
        add("North", m_toolbar);
        m_toolbar.add(m_first = new Button("First"));
        m_toolbar.add(m_prev = new Button("Prev"));
        m_toolbar.add(m_next = new Button("Next"));
        m_toolbar.add(m_last = new Button("Last"));
    }

    // APPLET INFO SUPPORT:
    //      The getAppletInfo() method returns a string describi
```

Visual J++ integrates seamlessly into Windows, but will other Java players accept Microsoft's ActiveX standards?

PCW: *What are the problems with the AWT library?*
De Michillie: "The first question is whether AWT will continue to only do things that can be done on 19 Unix variations plus the Mac, plus Windows 3.1, plus NT, plus 95. Second, AWT was developed by many different people and that comes through in the APIs that are exposed. There are different designs and they don't mesh well. There are problems with layout, which is entirely code-based. That means when I lay out a form, the layout is stored in the code for the form's class. There's no separation of the code from the data. Maybe these points are addressable, but the larger architectural problems are more difficult.

"I think AWT was rushed out early. A lot of the fundamental Java technology was ready to go; the byte codes, the compiler and AWT got rushed. We support AWT because there are no viable alternatives."

PCW: *How would you envisage your class library developing? Would you implement Windows-specific features, or go for compatibility?*
DeMichillie: "My personal view is that the least common denominator solutions are not ultimately compelling. We want to expose all the richness and functionality of Windows but we want to do so in a way that enables us to port to other platforms.

"For example, take Direct3D and DirectX, our multimedia systems. Those take advantage of high-performance Windows graphics cards, but the API is generic enough to implement on other systems. Or database access — there's no reason database access APIs would only be on the Windows platform."

PCW: *Is Microsoft happy to see Sun controlling the language, or would you like to see an ANSI Java, or something like that?*
DeMichillie: "We don't necessarily need an ANSI committee, Sun has control over what is considered standard Java. But there are a number of vendors working on class libraries independent of Sun and over which Sun has no influence.

"I would expect Sun to be keenly involved in things like the byte code format, but I don't think class libraries are really one of those areas. In our relationship with Sun, we're competitive in many areas and I make no apology for that. But having said that, there are going to be huge areas of commonality. We don't want to see byte code format proliferation. I think the Java language will evolve."

PCW: *What about the Java Beans proposals? Do they fall into the area that is competitive?*
DeMichillie: "It's difficult to say, simply because Beans is just so vaguely defined at this point. We want to make sure that anything Beans does works well with COM.

"Our overriding goal is to make sure that Java developers have access to the thousands and thousands of COM components that already ship. ActiveX controls and OLE controls form the most successful component software market."

PCW: *Why have you hooked into Internet Explorer (IE) and not made your product browser-independent?*
DeMichillie: "We're not directly hooked into IE. We're hooked directly into MS's implementation of the Java Virtual Machine, which currently is hosted inside IE. But because the interpreter is itself an ActiveX control, that VM could be hosted inside any executable. The first reason is that the VM offers COM support and ActiveX support, and second, the VM supports a new set of debug interfaces. I would personally love Netscape to adopt the Microsoft VM so we could cut down on the number of VMs that are out there."

PCW: *At some future point, might Visual J++ be able to compete with Visual C++ in creating mainstream Windows applications?*
DeMichillie: "Sure. My ultimate goal is for large-scale development to be possible in J++. The growth in Java will come at the expense of C++. People have now dealt with C++ for a number of years and have seen some aspects that are more complicated than they might like. Java offers a simplification that is very appealing. As the tools evolve, Java will be able to do many of the things that now would need C++.

"It's important to distinguish between component builders and component users. The majority of ActiveX controls out there are written with C++. VB 5.0 will also create controls. It will be interesting to see where the component builders go. Component users gravitate towards VB and will eventually gravitate towards VJ++ as these very GUI-based, RAD-like tools appear."

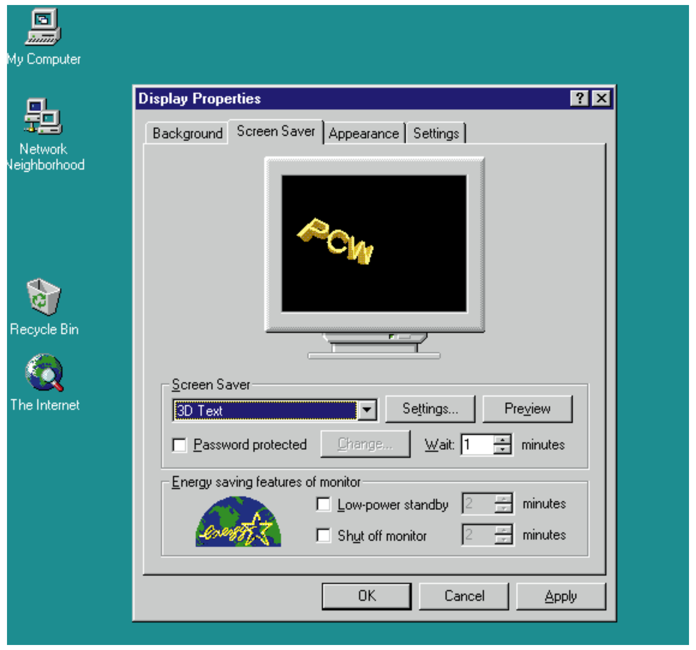
PCW: *If the lowest common denominator is not a long-term solution, does that mean cross-platform isn't either?*
DeMichillie: "No. There will be a core subset that's the same everywhere. But there will also be extra capabilities, even in Java, where one platform has an extra class and another doesn't. For example, there are a number of capability differences between Windows and the Macintosh. Do you really want to restrict the class library to only those that are common? Or do you want a class library that has the room to contain components that maybe work on three or four platforms, and other components that work on a different three or four platforms? Microsoft understands that the market is not just Windows, but includes Macintosh as well as Unix. But it does not follow that you are only going to do things that can be implemented on every platform."

Books for Visual Programming

Visual J++ by Charles Wood
The best and worst thing about this book is that it exists. It was on sale before Visual J++ was officially released in the UK and unfortunately it shows signs of haste. Nearly a third of the book is a list of Java methods — something you can get from online help. There is only brief coverage of ActiveX and COM and the author refers us to his forthcoming, more advanced book on the subject. Data Access Objects, one of the key selling points of Visual J++, are hardly mentioned. The rather important resource wizard (which lets you convert Windows menus and dialogues to Java) is skimmed over as being "beyond the scope of this book". What's left is an unexceptional general introduction to Java programming. It's not really the author's fault, since he was working with a beta product. But my message to you is, beware — the first books to come out for a new product are often not the best.

DELPHI

Anyone who writes a screensaver must have time on their hands. Screen burn is not common now, and in any case, perfectly functional savers are supplied with Windows. But they are fun! At least,



The Windows 95 screensaver control panel looks slick, but increases the work for developers

p318 ➢

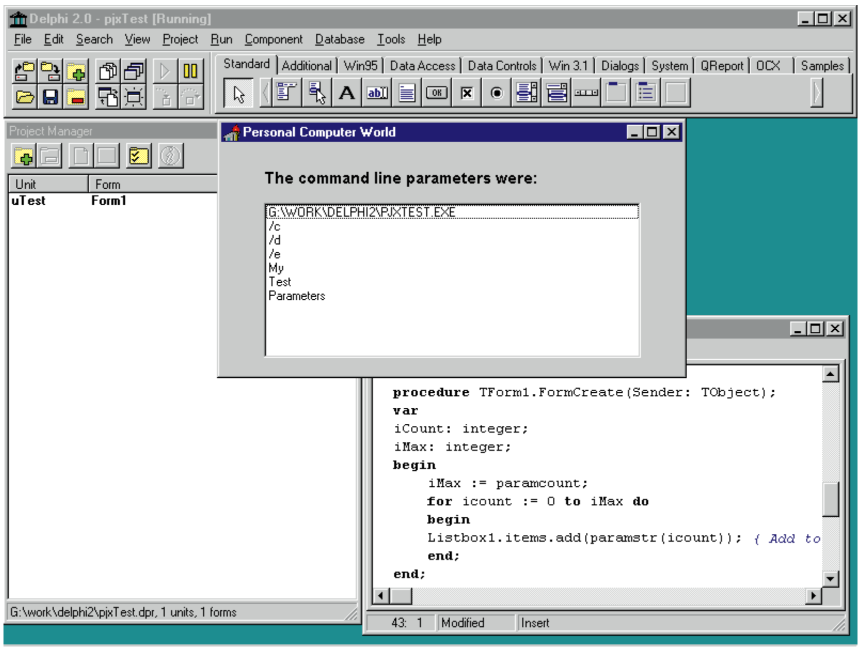
Delphi tip: Detecting command line parameters

It is often useful to supply parameters to an application at startup. For example, if an application handles documents, then passing the name of a document as a parameter should run the application and open the document. Delphi has two functions to make this possible. ParamCount( ) returns the number of command line parameters, and ParamStr(Index: Integer) returns a string representing the parameter that corresponds to Index. ParamStr(0) always returns the application name with full path.

The following routine detects command line parameters and writes them to a log file:

```
var
iCount: integer;
iMax: integer;
F: textfile;

begin
  AssignFile(F, ‘C:\TESTLOG.TXT’);
  Rewrite(F);
  iMax := paramcount;
    for icount := 0 to iMax do
    begin
    Writeln(F, paramstr(icount)); {
                write to log }
    end;
  CloseFile(F);
end;
```


Delphi 2.0 - pjxTest [Running]
File Edit Search View Project Run Component Database Tools Help
Standard Additional Win95 Data Access Data Controls Win 3.1 Dialogs System QReport OCX Samples
Project Manager
Unit Form
uTest Form1
G:\work\delphi2\pjxTest.dpr, 1 units, 1 forms
Personal Computer World
43: 1 Modified Insert

The command line parameters were:

```
G:\WORK\DELPHI2\PJXTEST.EXE
/c
/d
/e
My
Test
Parameters
```

```
procedure TForm1.FormCreate(Sender: TObject);
var
iCount: integer;
iMax: integer;
begin
    iMax := paramcount;
    for icount := 0 to iMax do
    begin
    Listbox1.items.add(paramstr(icount)); { Add to
    end;
end;
```

Andrew Jeffries must think so, since he asks: *“I am having a few problems writing 32-bit screensavers using Delphi 2.0, running Win95 and NT: How do you do a small preview in 95 without the configuration form always appearing? How do you implement security? How do you make the screensaver’s name appear in NT and 95?”*

The problem with screensavers is that they are not well documented. The trusty Software Development Kits (SDKs) for the various Windows versions assume you will use C or C++, and that you will link your application with SCRNSAVE.LIB, a Microsoft-supplied library that holds the secrets of screensaver operation. Screensavers do not have to use SCRNSAVE.LIB, but avoiding it means extra work on the part of the programmer. Another snag is that screensavers work differently in each version of Windows.

Screensavers are executed by Windows in two ways: either when an interval of inactivity causes Windows to execute the screensaver, or when it is being configured in Control Panel. In Windows 95, screensavers have four modes of execution and these are selected by command line parameters:

■ Preview mode. When you select the saver in Control Panel, Windows sends two parameters, /p HWND, to select preview mode and to pass the handle of the preview window.

■ Configuration mode. When you click Settings, Windows sends a parameter, /c, to select configuration mode. The saver responds by presenting a configuration dialogue.

■ Password mode. When you click to change the password, Windows sends two parameters, /a HWND, to select password mode and to pass the handle of the parent window for your password dialogue.

■ Start mode. When you click Preview, or when the saver is called for real, Windows sends a parameter, /s, to select start mode.

So the answer to Andrew’s first question is that the application should check the command line parameters to see whether it should draw in the preview window or present a configuration dialogue. See the tip panel (*above*) for how to detect parameters.

Screensaver security is treated in different ways by Windows 95 and Windows NT. Under Windows 95, most screensavers call the Windows Master Password Router. This is a DLL called MPR.DLL which exports password functions like PwdChangePassword. They are usually called via another DLL, PASSWORD.CPL, which works as an extension to the Control Panel. Neither of these libraries are fully documented in the Windows SDK, but some have worked out how to use them. The alternative is to implement your own password checking and throw up your own password dialogue when the saver is called in password mode.

Windows NT is different. Passwords for NT screensavers are the same as those used for logging on to Windows. The Control Panel marks a registry entry to indicate a secure screensaver:

```
HKEY_CURRENT_USER\Control
Panel\Desktop\ScreenSaverIsSecure
```

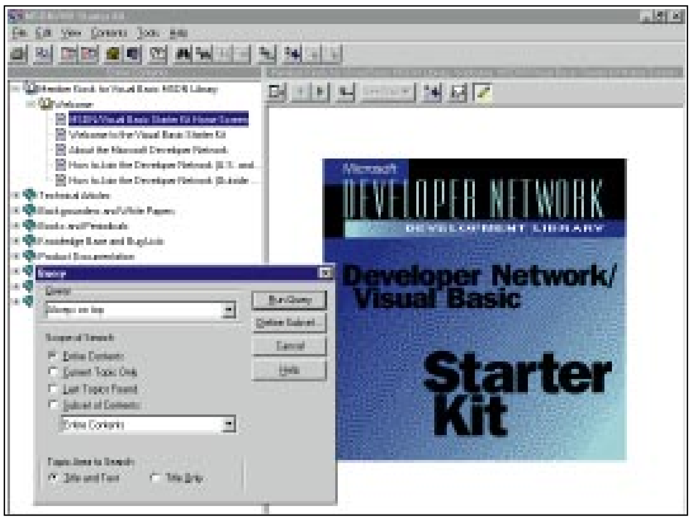
Finally, there is the matter of the description line. Confusingly, Microsoft has devised three ways of identifying this. Originally, it was the module description entry, which had to be of the form

```
“SCRNSAVE : My Description.”
```

Under Windows 95 and NT it became a resource string with an ID of 1 — and yes, Delphi can use standard Windows resources. This is the documented way; but actually, Windows 95 does not use it. It simply uses the long filename, less the .SCR extension. By the way, Windows will find any screensaver, identified by a .SCR extension, in the Windows or System folders, so at least installation is easy.

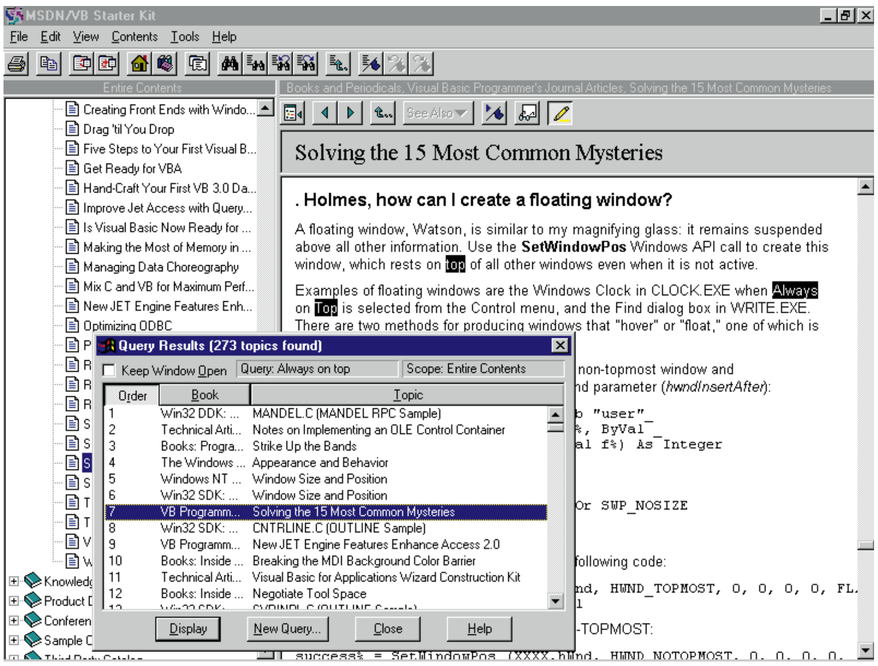
I’ve answered Andrew’s questions, but I

p320 ➢



(Left) Free on our cover-mounted CD: the Microsoft Developer Network starter edition is a mine of VB and Windows information

(Below) MSDN query results are displayed in order of likely relevance and clicking a title in the list displays the selected article


MSDN/VB Starter Kit
File Edit View Contents Tools Help
Entire Contents
Books and Periodicals, Visual Basic Programmer's Journal Articles, Solving the 15 Most Common Mysteries
See Also
Creating Front Ends with Windo...
Drag 'til You Drop
Five Steps to Your First Visual B...
Get Ready for VBA
Hand-Craft Your First VB 3.0 Da...
Improve Jet Access with Query...
Is Visual Basic Now Ready for ...
Making the Most of Memory in ...
Managing Data Choreography
Mix C and VB for Maximum Perf...
New JET Engine Features Enh...
Optimizing ODBC
Knowledg
Product D
Conferen
Sample C

Solving the 15 Most Common Mysteries

. Holmes, how can I create a floating window?

A floating window, Watson, is similar to my magnifying glass: it remains suspended above all other information. Use the **SetWindowPos** Windows API call to create this window, which rests on top of all other windows even when it is not active.

Examples of floating windows are the Windows Clock in CLOCK.EXE when Always on Top is selected from the Control menu, and the Find dialog box in WRITE.EXE. There are two methods for producing windows that "hover" or "float," one of which is

Query Results (273 topics found)

Keep Window Open | Query: Always on top | Scope: Entire Contents

Order	Book	Topic
1	Win32 DDK: ...	MANDEL.C (MANDEL RPC Sample)
2	Technical Arti...	Notes on Implementing an OLE Control Container
3	Books: Progra...	Strike Up the Bands
4	The Windows ...	Appearance and Behavior
5	Windows NT ...	Window Size and Position
6	Win32 SDK: ...	Window Size and Position
7	VB Programm...	Solving the 15 Most Common Mysteries
8	Win32 SDK: ...	CNTRLINE.C (OUTLINE Sample)
9	VB Programm...	New JET Engine Features Enhance Access 2.0
10	Books: Inside ...	Breaking the MDI Background Color Barrier
11	Technical Arti...	Visual Basic for Applications Wizard Construction Kit
12	Books: Inside ...	Negotiate Tool Space

Display | New Query... | Close | Help

non-topmost window and
nd parameter (*hwndInsertAfter*):

```
b "user"_
%, ByVal_
al f%) As Integer

Or SWP_NOSIZE
```

following code:

```
nd, HWND_TOPMOST, 0, 0, 0, 0, FL
l
```

-TOPMOST:

```
success% = SetWindowPos (XXXX.hWnd, HWND_NOTOPMOST, 0, 0, 0, 0,
```

do not mean to suggest that writing screensavers is easy. The main problem is poor documentation, especially if you are not using Visual C++. A hunt around CompuServe or the web will throw up Delphi examples and help files created by other frustrated users.

■ Psst! Want Delphi cheap? Borland is bundling Delphi 1.0 with a book, *Teach Yourself Delphi in 21 Days*, and offering it for just £34.99 (plus VAT). A similar package for Delphi 2.0 costs around £69. Borland says the packs are aimed at "students, hobbyists and programming beginners".

VISUAL BASIC

The web is buzzing with talk of VB 5.0, now likely to be released early in 1997. There's even a web site devoted to VB 5.0 news and comments from anonymous beta testers. If the rumours are even half true, it looks like both performance and features will be hugely boosted.

In the meantime, Aaron Hodgson has contacted me with a question about Visual Basic: *"I am trying to write a terminal program using SAXCOM.VBX. The terminal works fine but I would now like to add an automatic logon sequence, where the logon details are read from an initialisation file when the remote computer on the other end of the modem prompts the user to log in. If the answer to my questions involves complex things like DLLs please explain, because I don't know the first thing about using DLL files with Visual Basic."*

Reading data from an initialisation file is not difficult. You can use API functions like GetPrivateProfileString, which uses a standard Windows .INI file. Searching the free MSDN sample on our cover CD-ROM will soon pull up an example. Alternatively, if you prefer native BASIC, there are the neglected statements like Open, Print# and Input#. These are well-documented in the Visual Basic manuals, with examples.

Microsoft Developer Network

By special agreement and arm-twisting, we've included the MSDN starter edition for Visual Basic on this month's cover-mounted CD-ROM. I get regular enquiries about MSDN, and now you can try it for yourself.

Although this is only a starter edition, there is 125Mb of documentation, tips and tricks included, so no-one need feel short-changed. It even includes two complete books: Petzold's classic *Programming Windows 3.1*, and BrockSchmidt's tome explaining OLE 2. Much of the information covers VB 3.0 as well as 4.0.

The best thing about MSDN is its fast searching. For example, you might want to know how to set a window to be always on top. Click search, enter "Always on top", and then Run Query. A moment later, MSDN presents 273 topics ordered by likely relevance. Article 7, "Solving the 15 most common mysteries", has a section explaining exactly how to do it.

A subscription to the full MSDN comes at several levels and prices. Information is available on the CD, or call Microsoft for details.

■ See next month's *PCW* for a review of the new Crystal Reports 5.0, and components from Sax software and Microhelp.

Cover Disk

Files from last month's *Hands On Visual Programming* were, unfortunately, left off the CD, but they can be found on this month's cover disc.

•PCW Contacts

Tim Anderson welcomes your Visual Programming comments and tips. He can be contacted at the usual *PCW* address, or at **freer@cix.co.uk** or **www.cix.co.uk/~tim-anderson/**

Learn Borland Delphi in 21 Days £34.99 (plus VAT), or £69 (plus VAT) for the Delphi 2.0 version. Borland is on 01734 320022.
Visual J++ is by Charles Wood, ISBN 07615-0814-7. £32.99 from Computer Manuals, 0121 706 6000.
Microsoft Developer Network is on 0800 960279.