



Spare parts

Tim Anderson compares components for Basic and Delphi. Plus, for those who can't see the wood for the trees there's a guide to choosing a visual programming package.

Visual programming means dropping components into your application and making them go by setting properties and calling methods. That, at least, is the plan and here is an evaluation of some recent components. Your views matter most though, so please let me know which components work well or badly for you.

Crystal Reports 5.0

Crystal Reports is hard to avoid, being widely bundled with products like Visual Basic and Visual dBase. Seagate naturally hopes that users of these bundled versions will want to upgrade. Version 5.0 is the latest release. The Standard version supports most desktop database formats like dBase, Access and Paradox, while the Professional version adds full ODBC and various native SQL formats.

There is a new interface for designing reports, with better drawing features and in-place OLE editing. Of most interest to developers is the new sub-report feature, which enables you to insert a report within a report. Normally, this would be linked to the main report for displaying child records, but it can also display unrelated data. Another new feature enables you to export HTML for adding to web sites.

Crystal is a powerful tool and has components for most development languages including 16- and 32-bit Visual Basic, Delphi and C++. But I do have reservations. One is the sheer size of the product: the main print engine is now over 3Mb; another is that Crystal has its own formula language and although reasonably capable, it is ugly and old-fashioned. Nevertheless, version 5.0 is a significant

upgrade and developers with better things to do than write their own report engine will find it invaluable.

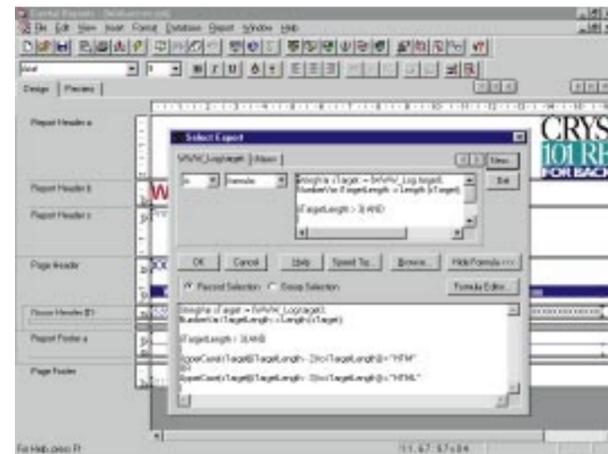
GeoPoint 1.0

Visual Components is responsible for some of the best ActiveX components around, including Formula 1 and Visual Writer. GeoPoint is a more specialist control. It

displays maps in MapInfo or Autocad format. By using it alongside the separate Legend control, you can programmatically control the text and shading of each area of the map. A technique called "binning" lets you categorise data into ranges, and then shade the map accordingly: a typical example would be a display of sales performance by region. The user can also

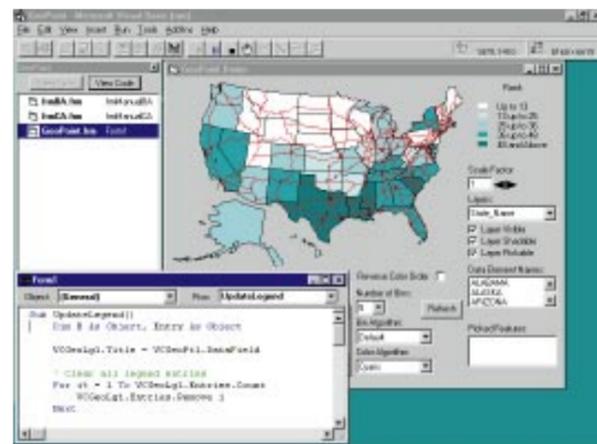
select an area of the map by clicking, so the application can show related data. GeoPoint can be bound to a data control to display your data.

This useful component is spoiled by its presentation. There is no printed manual and the online documentation is poor. The other snag is that the few supplied



Above The Crystal Select Expert lets you create custom fields using the formula editor

Right GeoPoint is a neat tool with which to analyse geographic data, but its USA map will not be of much use in the UK

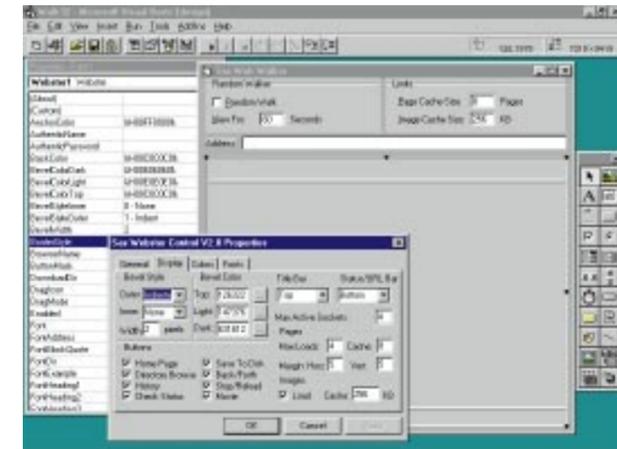


maps, heavily biased to the USA, are not likely to be what you want. That means purchasing add-on maps, or buying MapInfo or Autocad to create your own. Making full use of GeoPoint will be expensive.

Sax Webster or Internet Explorer?

The Sax Webster control displays HTML documents. Now at version 2.0, it comes as 16- or 32-bit OCX controls that you can drop into your application.

It is easy to use. You can, for example, display a web page by setting the PageURL property. The main change from the first Webster control is the HTML version supported, now version 3.0 but without frames. It works well and may be useful on 16-bit systems or where a small memory footprint is required. Otherwise, on 32-bit Windows a better option is Microsoft's freely available Internet Explorer 3.0. In Visual Basic 4.0, open the Custom Control dialogue and check Microsoft Internet Controls. This installs the WebBrowser component which is the HTML display part of Internet Explorer. It is just as easy to use as Webster, and far richer in terms of HTML support. You will need to get hold of the Internet Explorer object model, which is part of the ActiveX SDK available on the Microsoft web site.

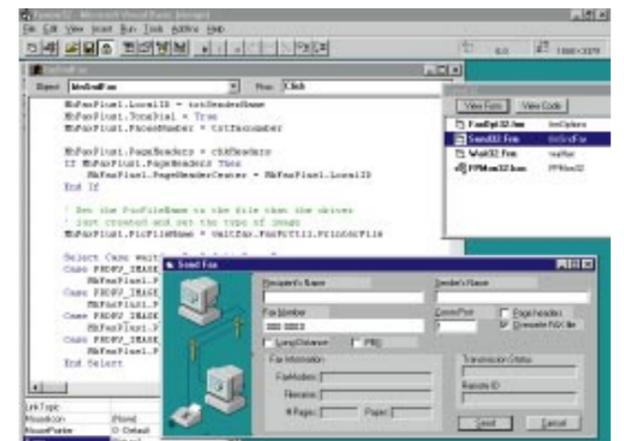


Left Sax Webster: it works, but why not use Internet Explorer instead?

Microhelp Fax Plus

Fax software may not be exciting but it is exceptionally useful, at least until the whole world gets webbed.

Fax Plus 2.0 is the new 32-bit version of Microhelp's Fax add-on. It is designed for Windows 3.1 and 95, but not NT. It consists of several controls, including a fax control



Below FaxPlus: does it have to be so complicated?

How to choose a visual language

Shaun Nicolson writes: "I am considering buying a visual programming package but cannot decide which one. I am considering Microsoft's Visual Basic and Visual C++. The language I choose would have to produce network applications. What are the pro's and con's of these?"

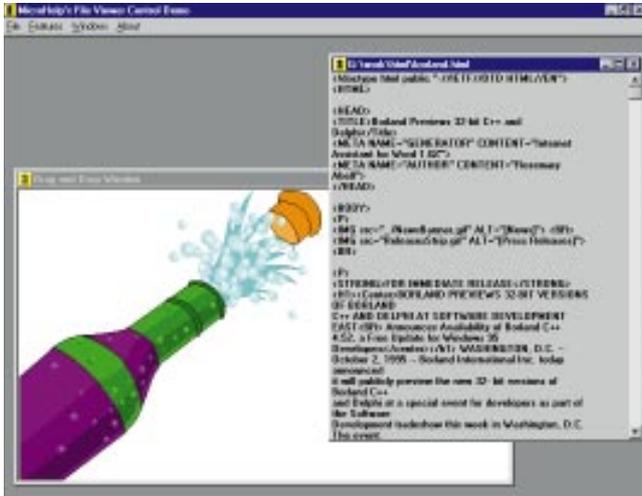
1. Ease of use. This is where Visual Basic scores highly, since you can have a simple utility up and running very fast. That does not mean VB will be the easiest for a large, complex application, since many other factors will then come into play. Delphi also scores well, while Visual C++ is hard to learn with limited visual tools.
2. Performance. This is where languages that compile to native executable code, like C++ or Delphi, generally win over interpreted languages like VB, FoxPro or Java. Database processing speed should be judged separately as all products use fully-compiled database engines. In some applications, performance is not an issue, or is determined more by factors like graphics or hard disk speed rather than the language used.
3. Power. The developer's nightmare is to spend months on a project, only to find that some intractable problem means that it cannot be completed with the current tool. Visual Basic is vulnerable since some features of Windows like callback functions or custom message handling are not available. There is usually a way around it by using a custom control or DLL but these must be written in another language. Version 5.0 should solve some of these problems. If you dread brick walls, C++ is the safest option, with Delphi a close second.
4. Database engine. Most languages have a native database engine, along with the ability to connect to other databases via SQL libraries or ODBC. If you know which database you will be working with, good connectivity is the first thing to check.
5. Availability of add-ons. This is where Visual Basic scores best. Most VBX and ActiveX controls are designed for VB, and may not work well elsewhere. There are also plenty of code libraries for C and C++, but native add-ons for other languages are more limited.
6. Reusable code. To protect your investment, you want to write code that will be reusable in future projects, perhaps even on other computer platforms. This is one of the benefits of object orientation, with Delphi scoring well, C++ fairly well, and Visual Basic less well. Best of all is Java, which forces you to write object-orientated code and runs on multiple platforms.

that handles communications, and a FaxImage control for creating and modifying fax bitmaps. There is a printer driver and control which lets you send faxes by printing from any Windows application. The Fax Plus driver creates a fax image and then fires a StartDoc event in the printer control, so that your code can handle sending the fax.

Unfortunately though, using FaxPlus is not as easy as it should be. In part this is because of fickle telephone lines and diverse hardware that turn faxing into a trouble-prone business. Other problems are down to FaxPlus itself, which is awkward to code and not entirely bug-free. For instance, at the time of writing, the VBX version is unable to correctly convert text files to fax images. Still, it beats trying to write your own fax driver.

Microhelp VB Viewer 2.0

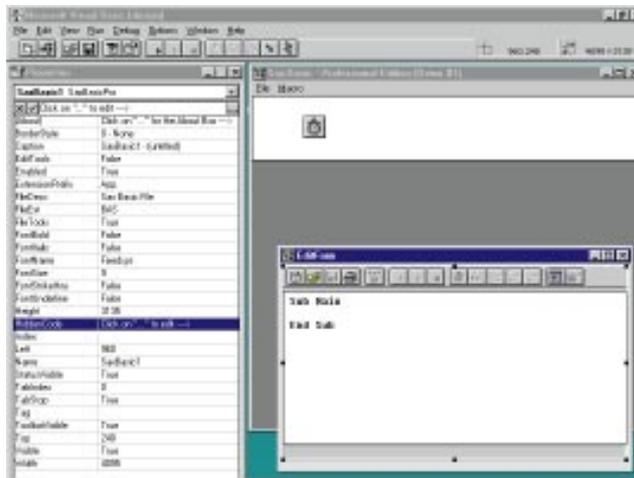
VB Viewer is a product of limited ambitions. Drop it onto a form and you can use it to display files of around 30 different types, including multimedia files. With text documents you can search for text and



Left VB Viewer can manage a picture, but struggles to display HTML

Below A language within a language, Sax Basic lets you deliver programmable applications

copy to the clipboard. But overall, VB Viewer is a disappointment. One let-down is that formatted documents in word processor formats are displayed as plain text only, unlike the much better QuickView utility that comes free with Windows 95. Some basic formats like HTML and Rich Text Format are not supported at all. You can set VB Viewer to use QuickView viewers, but files then appear in the QuickView window rather than embedded in your form.



has a more elegant solution. You write extensions in a VB class module, and then add them to Sax Basic using the control's AddExtension method. If your application needs a macro language, Sax Basic is ideal and a lot cheaper than licensing the genuine

Sax Basic Engine

One way to impress users is to supply an application with its own macro language, like Excel or Word. The Sax Basic Engine lets you do just that. The control has its own IDE, so getting started takes little more than placing it onto a form. The language itself is compatible with Visual Basic for Applications, with excellent support for OLE automation and class modules.

To make Sax Basic useful, you need to extend its language to communicate with your application. The way to do this differs, depending on whether you use the VBX or OCX version. With the VBX, you can add keywords that fire an event called AppExec. You can then write code using Select Case to interpret the command. The OCX version

Merging bitmaps using VB and the SRCAND bitwise operator

VBA from Microsoft.

Visual Basic

MT Emms writes: "Using Paint I've created four BMP files. Each is divided into four sectors, the other three being left transparent. I have written a program to merge these bitmaps into one but the last one dominates — in other words, the transparent sectors are not transparent. It was simple on the Mac and Archimedes; surely VB should be able to cope?"

Visual Basic can cope, but it is not as simple as it might be. The secret in this case is to use the PaintPicture method, on either a form or a picture box. The syntax for PaintPicture is:

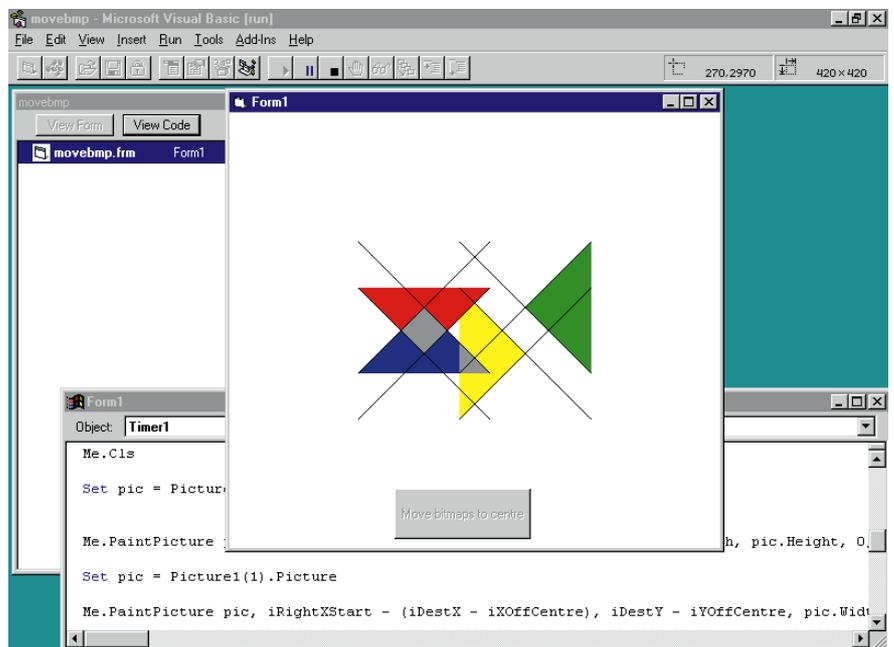
```
object.PaintPicture picture, x1, y1, width1, height1, x2, y2, width2, height2, opcode
```

The final parameter is a long value that defines a bit-wise operation which is performed on the picture as it is drawn. As the VB manual remarks, you can find a list of these operators in the BitBlt topic in the Windows SDK help.

The easy way to use them is to define them as constants in your VB application. For example:

```
Const SRCAND = &H8800C6 ' (DWORD)
    dest = source AND dest
Const SRCOPY = &HC0020 ' (DWORD)
    dest = source
```

If you then call PaintPicture with the SRCAND constant value, the bitmaps will merge in the way Mr Emms requires. Yes, it is more like programming in C than in Visual Basic, but at least it does the job. An example application is on our cover-



mounted CD, which also shows how to move the bitmaps across a form for an animated effect.

Delphi departure

In October, Anders Hejlsberg, the chief architect of Delphi, announced his departure from Borland for the safe pastures of Microsoft. Zack Urlocker, another key member of the Delphi team, has pointed out that "the architectural work that Anders covers is complete for Delphi 97. Anders' departure won't affect the shipping date or features going forward."

Even so, Anders is widely seen as the man without whom Delphi would never have happened, so his move is significant news. If anyone can knock VB into better shape as an OO language, he must be the man. Although Delphi is as good as ever, this weakens the case for migration from Microsoft tools. Personally, I hope that Borland can sustain Delphi's momentum, as it still delivers the best combination of rapid

Dear Santa...

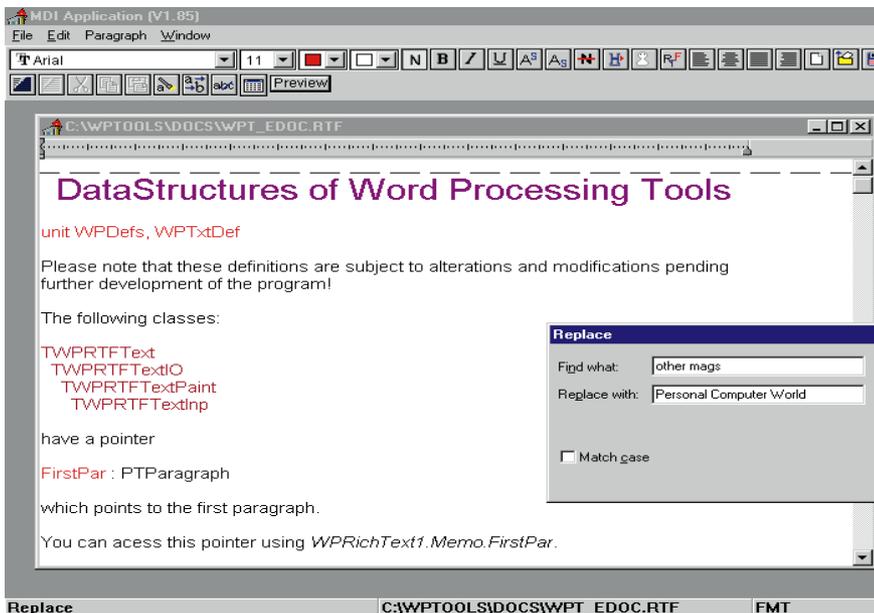
Sensible developers want an easy life. That means fast application building, reusable code, blinding performance and results that run everywhere. Well Santa, it seems you have a habit of giving with one hand and taking back with the other. Last year I asked for an end to the OS wars: a fanciful request, perhaps, but twelve months on and Java may provide an answer. Except that (dear Santa), we need easier, richer interface building, better performance, and decent support for platforms which Java finds difficult, like Windows 3.1 and the Macintosh. In the meantime, even developers who fix upon Windows have three versions with which to contend.

Forget the platform, then, let's look at the tools. First, there's Visual Basic, still the most popular all-round Windows language. Last year's wish-list included a compiler and better OO. The signs are that VB 5.0 delivers some of that, although it will never have the elegance of Delphi's component library. But Office 97 and VBA 5.0 are great news for developers of Office solutions. Thanks, Santa. And thanks for Optima ++, which is real visual development for C++ at last.

While I have your attention, there are a few things I'd like in my stocking for next year. Top of the list has to be a faster, better-organised internet. The web is irresistible for developers, both for technical support and as a platform in itself. But it has to get quicker and more reliable. *Please*.

Second, an un-present: Windows 3.1, please take it away. It's as bad as DOS, but worse, because people with working Windows 3.1 installations see no reason to change. I understand their point of view, but for developers this is a disaster. Develop two versions, with all the extra costs? Develop 16-bit only and waste all the advantages in 95 and NT? Develop 32-bit only and forget half the market? Hmmm, did I hear someone mention Java?

Third, I'd like better tools for troubleshooting OLE, ActiveX, COM, call it what you will. It's funny how quickly a Visual Basic 4.0 installation produces an "object server not correctly registered" message when you install custom controls. It means a registry problem and there's no easy way to fix it. This highlights a problem that will get worse if ActiveX continues to grow.



development, power and performance.

Word processing tools

Some months ago I mentioned a shareware product called WP Tools, a native Rich Text control for Delphi. On closer inspection, I am impressed. The feature list is good, with support for merge fields, graphics, tables and hyperlinks. The range of controls has a lightweight rich text label and a data-aware text box as well as the usual word processor, toolbar and status bar components. In tests, it has proved fairly reliable, though not entirely bug-free.

The advantage of WP Tools is that as a

native VCL supplied with source, you can track down bugs and amend the code if you can work out what is going wrong. Another benefit is richer functionality. You can access the data structures for both text and formatting, giving a fine degree of control. The Finder class offers sophisticated search and replace, including formatting properties. You can print to a canvas control in order to implement page preview.

Performance is good, on a par with rivals like Visual Writer, AllText and HighEdit. If you are developing for 16-bit Windows, a custom component is all but essential for

WP Tools is a fully-featured, shareware-rich text control for 16- or 32-bit Delphi

working with formatted text, while even in Windows 95 and NT it has advantages over the built-in rich text control.

There are problems. This is shareware, and the documentation is unclear. Advanced users need to be comfortable with such things as streaming and pointers, as WP Tools uses them extensively. To succeed with this product, you must be willing to pore over the source and not be put off by the odd mixture of English and German comments in the code. The extra effort and risk is rewarded by a product that works rather well.

•PCW Contacts

Tim Anderson welcomes your Visual Programming comments and tips. He can be contacted at the usual PCW address or as freer@cix.co.uk

Components listed below are available from:
Contemporary Software 01344 873343; **Grey Matter** 01364 654100; and **QBS** 0181 956 8000.
Sax Webster £110 for the 16- or 32-bit version (£175 for both), plus VAT.
VB Viewer £110 (plus VAT)
FaxPlus £195 (plus VAT)
Seagate Crystal Reports 5.0 Standard £199, Professional £299 (both plus VAT)
Sax Basic 3.0 Pro £345 (plus VAT)

GeoPoint 1.0 costs £195 (plus VAT) from **Visual Components** 01892 834343
WP Tools is \$119 to register, available via CompuServe, or from the web at members.aol.com/JZIERSCHE/wptools