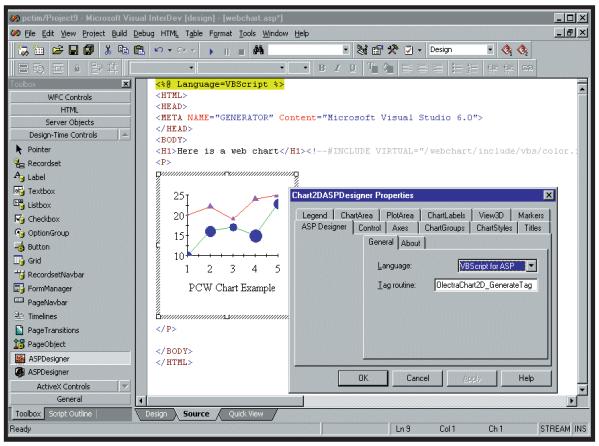
Chartist movement

Tim Anderson looks at an app that turns an ActiveX control into a cross-platform component.

Apex Software's True WebChart component is a sign of the times. It claims to be version 6.0, but is really version 1.0 of the WebChart component. The reason for the numbering is that True WebChart is Olectra Chart in another guise. The previous incarnation is a well-established ActiveX charting control. True WebChart is an ASP (Active Server Pages) server-side component for displaying charts in web pages.

The interesting point is that converting an ActiveX charting control to an ASP component is, in principle, very easy. The output from a charting control is an image. All web browsers understand images, so the ASP script only needs to drive the charting object, get it to save an image, and serve the image back to the browser.

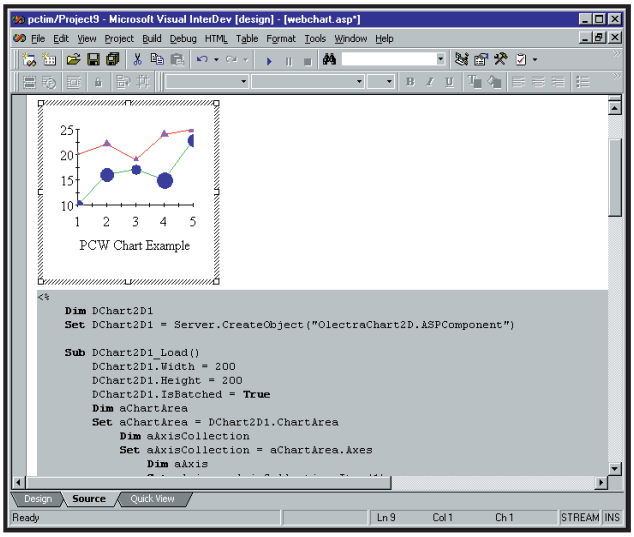
This is exactly what True WebChart does. You also get a design-time control for Visual InterDev. Design-time controls are a little mysterious at first, so here is an explanation. Visual InterDev is a RAD environment for building ASP pages. If you include a COM object (such as True WebChart in an ASP script) then, depending on what sort of object it is, you may well end up with reams of code setting its properties. True WebChart is a good example, with over 100 lines of code used simply to define a default chart. The design-time control is an ActiveX component that generates this code for you, using properties set via a familiar property sheet dialog. When you view the source, you see by default a visual representation of the object.



Behind the mystery: the default view of a design-time control in the Visual InterDev source editor

If you right-click and choose View run-time text, then the generated script is displayed below it. This is pretty much essential for debugging. You cannot edit this script, other than by going back to the property sheet. If you need to tweak the code in ways that go beyond what the design-time control can achieve, there is an option to Convert to run-time text. This deletes the design-time control but leaves the generated script, now in read-write form. There is no turning back since you cannot recover the visual control. The system works well, saving time and reducing errors.

Going back to True WebChart, I have to admit it took some effort to get it working. I installed it on NT4 with the Option Pack. Running the demo web pages gave 'Internal server error' whenever it attempted to show a chart. Visual InterDev was no better. I could design a chart, but trying to run the page threw up the message 'Chart control cannot load this newer version of saved chart properties.' My first thought was to look on the web for help, where I discovered an Olectra chart newsgroup, three users with the same problem and no answers, a classic newsgroup scenario.



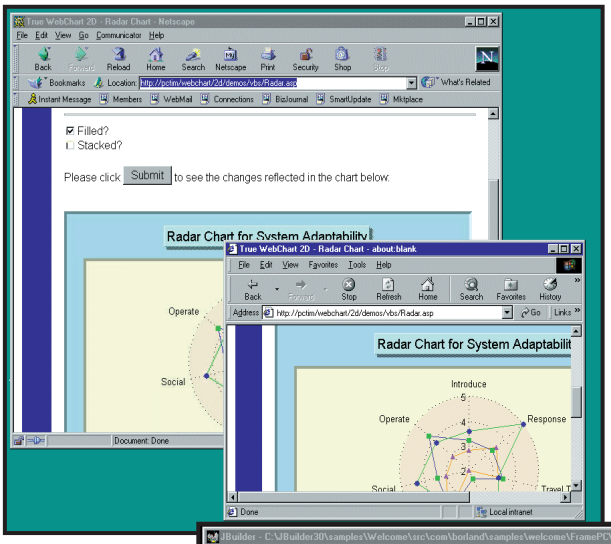
In the half-way house view, the generated code shows underneath the control

Careful debugging showed that True WebChart was not actually saving any temporary files. The idea is that if the ASP script detects Internet Explorer, then it saves a properties file for a client-side ActiveX control. The error message meant that the control was attempting to load a non-existent file. If the script detects Netscape or other browsers, it tries to save an image, and this was generating a broken link as the image file did not exist. The problem turned out to be one of permissions. True WebChart managed to set itself up with a directory for temporary files that was read-only. In the case of the demo web pages, there was no application defined in Internet Information Server so the ASP pages could not run. Both problems were easy to solve using the Internet Service Manager, and once this had been done everything started working.

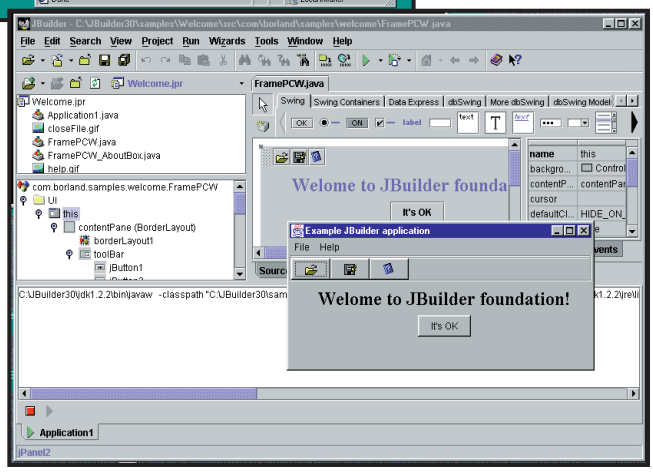
True WebChart lets you control whether Internet Explorer gets a plain image like Netscape, or uses an ActiveX control. Visually, both produce the same results. The ActiveX client has some advantages. The user can right-click the control, and get the same property sheet that appears at design-time. If you want the user to be able to change the chart

from Radar to Pie, this is a great advantage. Another factor is the download time. Admittedly, the ActiveX user has to install the control before the charts will display, which could involve over a megabyte of CAB files, but once completed a typical chart is under 2K in size, whereas the equivalent JPEG image is around 70K. On web pages for internal use, the ActiveX version wins, but for public-facing sites the simplicity of image generation has more appeal.

The end result is a rich data-aware charting control that is ideal for creating impressive cross-platform web applications. There are 2D and 3D chart types, with support for bubble, HiLoOpenClose to display stock prices, candle, polar and more. The 3D chart lets you control rotation, perspective and orientation, as well as all the properties you would expect, such as colour, border and legend. The package includes full OCX controls for Visual Basic, Visual C++ or Delphi, along with a DLL version for use if you prefer to avoid ActiveX. The caveat is that since the installation went wrong so easily, there is the worry of further bugs that might emerge. There is food for thought here too. The power of ASP is that some native Windows applications can convert remarkably easily to cross-platform web solutions, provided you are willing to run Internet Information Server on NT.



Netscape has loaded True WebChart bitmap, but Internet Explorer uses a client-side ActiveX. There is no visible difference



Builder's cracked: Versions of JBuilder Foundation for Windows, Linux and Solaris are free to download – this is the Win98 version

Revolution at Inprise

Inprise, or is it Borland, is embracing Linux. There are three significant events that signal this. One is that JBuilder, the RAD Java tool, has been released in a free version called JBuilder Foundation. This is a no-frills JBuilder IDE without the JBCL (JavaBeans Component Library), wizards, and some debugger features. Even so, it gives you a professional visual design tool for Java that can be run on Windows, Linux or Solaris, including drag-and-drop Swing components.

All platforms are not equal in the Java world, and in practice Solaris and NT get priority

JBuilder's new cross-platform talents indicate that it is now itself a pure Java product, so it will run on any compatible JVM. Compatible in this instance means JDK 1.2.2 or higher. All platforms are not equal in the Java world, and in practice Solaris and Windows NT get priority, while Inprise with others has made a special effort to help in providing JDK 1.2.2 for Linux. JBuilder Foundation also just about runs on Windows 95 or 98. Incidentally, if you want a full-featured JBuilder on Linux, the recently released Solaris Enterprise edition also runs, provided that you install the latest JDK as above. Be warned that JBuilder is still frustratingly slow on most systems, the main barrier to high productivity.

The next big news is that Interbase, the Inprise client-server database, is becoming an open-source product. Exactly what this will mean is not clear at the time of writing, but a likely scenario is that Interbase will be available for free deployment on an unsupported basis, the code will be available for download, and Inprise will retreat from investing in Interbase development while still hoping to profit from support contracts and related products. In a way it represents both good and bad news for Interbase. It reflects lack of success in winning market share against the likes of Oracle and Microsoft SQL Server. Interbase is a good product though, and pushing it out to the Linux and open-source community can only increase its popularity.

An Inprise product that stands to benefit from a freely-available Interbase is Kylix, the codename for Delphi running on Linux. Developers using Linux are already well served by GNU C/C++ among other tools, but there is nothing like Delphi for rapid visual development. Delphi for Linux will attract both existing Linux developers and Windows developers who like the idea of creating applications for a platform that is free to deploy. There is going to be pain in porting from Windows to Linux, even if it is Delphi to Delphi, unless Inprise plans to port the whole Windows API complete with COM and ActiveX, which I doubt. Even so, it brings closer the possibility of deploying Linux on the desktop as well as

on servers, with custom Delphi applications accessing Interbase. Another possibility is to develop web applications using Apache and Interbase, accessed by clients on any platform – much as described in this month's *Web Development* column. Windows has advantages as well, but the number of situations in which Linux is a viable alternative is increasing.

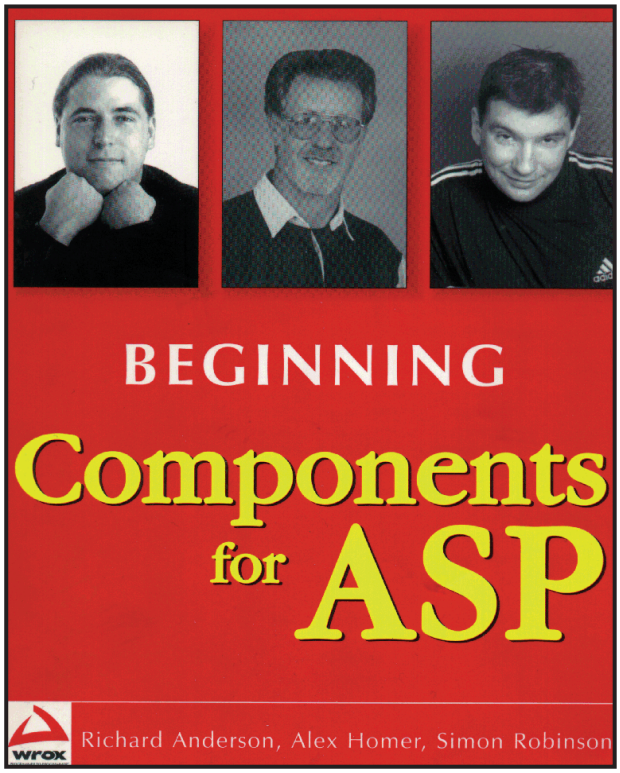
ASP components

For years Microsoft has been urging those involved in Windows development to build applications with COM components. While this has caught on to some extent, developers have resisted, for many good reasons. Every COM component has to be installed and registered in the Windows registry, making deployment more difficult and error-prone. COM itself is complex, and while tools like Visual Basic simplify some issues, others have a labyrinthine complexity. Threading models, object lifetime, managing state and preserving compatibility are some of the trickiest areas. Distributed COM (DCOM) is powerful but brings new difficulties. The safe path has been to use Delphi or C++ to write traditional Windows applications, keeping as much code as possible within the executable and DLLs (Dynamic Link Libraries) that you supply.

While this reasoning still has some force, the tide has turned. The implementation of COM in Windows has improved over the years. One reason is that Microsoft's own applications rely on it heavily, so it has to deliver reasonable performance and reliability to keep Microsoft Office running. The other factor is Internet technology. Browser-based applications make increasing sense in the web era. Partitioning an application between presentation, middle-tier logic and back-end data is essential if you need to support a variety of platforms, from Windows clients and browsers, to handheld devices. The next Visual Studio will let you create web services, COM objects that can be accessed by sending and receiving XML, a realistic proposition for almost any device. Windows, of course, is not the only game in town, and in particular Java objects such as Enterprise JavaBeans are travelling the same path.

Getting started with component development is not particularly easy. It means a new application architecture, as well as coming to terms with new technology. There are some very good books covering how to do this. One good one is from Wrox press, which has published a number of titles on how to build COM applications. The one in my hands is *Beginning Components for ASP*, which, despite its title, is not just for novice programmers. As well as the focus on ASP, it is also useful for any Windows developer wanting to understand how to create COM applications. The first two chapters cover the basics of COM components and building them in Visual Basic. Further chapters tackle data access through ADO (Advanced Data Objects) and multi-tier application architecture. MTS (Microsoft Transaction Server) is covered in detail. This is the application that manages COM objects, sharing them sensibly when there are a large number of clients and supporting transactions for robust data access. One of the problems posed by MTS is that components should be stateless, which means they do not store data such as the current record, or the identity of the client. In a stateless component, such information is used by the components methods, but not stored in global variables or properties. This means that state has to be stored in other ways. In Windows 2000, MTS has been renamed Component Services, showing how important it has become.

The more advanced sections of the book tackle: Microsoft Message Queue, which is a robust mechanism for communication between distributed components; working with the Active Directory; and how to create lightweight components using ATL (the Active Template Library) in C++. Finally, there are two detailed case studies, one covers a document management project and the other is a cinema booking application. If you are planning a Windows application that is aimed at more than single users, this is exactly the kind of information you need to absorb.



Beginning Components for ASP: an excellent guide for novices and pros

MTS has been renamed Component Services, showing how important it has become

CONTACTS

Tim Anderson welcomes your Visual Programming comments and queries. Contact him at **visual@pcw.co.uk**, or via the *PCW* editorial office (address on page 10).
Download JBuilder Foundation from **www.borland.com**
True WebChart 6.0 costs £499 (ex VAT) from: Contemporary 01344 873434. More information at **www.contemporary.co.uk**