



visual programming

Crystal tips

Tim Anderson looks through the **new elements** in Crystal Reports 7.

Most database projects need reports, and most developers will be familiar with Crystal Reports, the leading Windows report component. A version of Crystal Reports has been bundled with Visual Basic since version 3.0 although VB 6.0 has a new native report component and Crystal has been relegated to a dark corner of the installation CD. It is still there, though, under the Tools directory and for many projects is a better choice than the newcomer. The bundled version, also available as a free web download, is for VB 5.0 but it also works with VB 6.0.

For the real VB 6.0 version you have to buy Crystal Reports 7.0. The updated VB control supports the VB Data Environment and hierarchical datasets but there is more in the package than just a new component. Particularly interesting is the Crystal Web Reports Server, which supports Java, ActiveX or plain HTML clients.

At the heart of the package is the Crystal Reports Print Engine. It is a library called CRPE32.DLL which has the dubious distinction of being the largest DLL in many windows installations and is now more than 5Mb. If you prefer to work close to the metal, you can call the Print Engine API directly but Seagate provides numerous higher-level tools. The starting point is the report designer

[Fig 1] which lets you define the database connection for a report and lay out the fields and sections. There is support for sub-reports, cross-tabs, parameters, formulas and charts.

With the product now at version 7 the feature set is extensive although report design has a habit of raising awkward problems that may require compromises. Reports are saved as files which can then be displayed programmatically or opened by the user.

There are several options for developers who need to integrate Crystal Reports into an application. In Visual Basic you can use the ActiveX Designer, a wizard that sets up a database connection, opens a report designer and adds a report viewer to the project, along with code that displays the report.

The viewer is a component that sits on a VB form complete with a navigation tree that lets you jump quickly to any report section. This makes for better integration than was possible with early versions, although you still cannot place standard VB controls on a report. The ActiveX report does support a Format event for which you can write VB code.

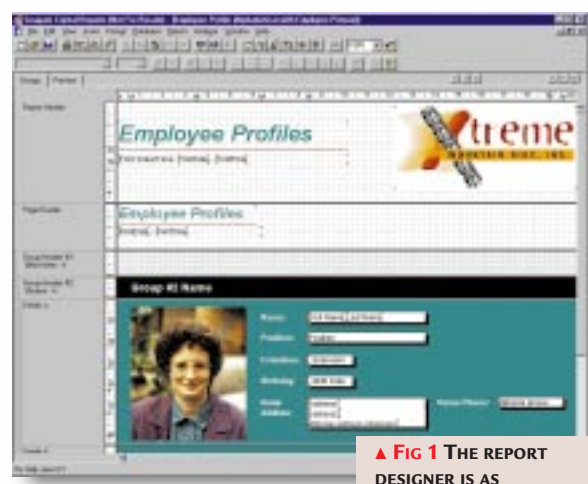
This is an easy route to adding a report to a project but less flexible than the traditional approach which involves creating report files using the full Crystal Reports designer,

and using the Crystal ActiveX control to load and manipulate the report. Other ways of programming the report engine include an automation object model, a class library for C++, and a VCL component for Delphi.

The interesting features in Crystal Reports 7 are for web applications and there are several options. One is to use Active Server Pages, the dynamic web page technology in Internet Information Server, and to create report pages using Crystal's automation interface to poke reports into an ActiveX or Java viewer. An easier way is to run the Crystal Reports server which can be programmed using Microsoft or Netscape web server API (ISAPI or NSAPI), or CGI scripts. The client can be plain HTML, or an ActiveX or Java viewer. I tried this with Internet Information Server and was impressed with the ease of setup and the results.

Seagate has made the different viewers similar in appearance and functionality. For Java, the supported clients are either Microsoft's JVM in Internet Explorer 4.0 or higher, or any browser using Sun's Java plug-in. The server needs additional licenses for each simultaneous user.

Customers who use reports typically want a large number of variations, which can result in hours of tedious editing in Crystal's designer. You can reduce this to some extent by using report parameters



▲ FIG 1 THE REPORT DESIGNER IS AS COMPREHENSIVE AS YOU WOULD EXPECT FROM A VERSION 7.0 PRODUCT



◀ FIG 2 CRYSTAL QUERY IS ANOTHER JAVA APPLLET; THIS TIME FOR DESIGN AS WELL AS A PREVIEW OF REPORTSEXCEL



or programmatically altering selection criteria but not all variations can be accommodated this way. The answer, at least with users who do not mind doing a little work, is to provide a way for end-users to create their own reports.

You can now do this over an intranet or on the web using Crystal Query [Fig 2], a Java application which lets you open, amend and save standard Crystal report files.

Seagate supplies an installer which you can download and

execute locally so that the application classes end up on the client for better performance. The installer will kick in again whenever the client needs to be upgraded.

Seagate's Crystal Reports has evolved into an impressive suite of reporting applications. Developers and end-users are well served, and the web application features are particularly good.

■ Visual Café society

An interesting question arose in a letter from reader Mark Jackson: 'I am currently trying to implement a web database in Visual Café. The problem is that I've managed to create an applet linked to an Access 97 database using the project wizard but it isn't reading in any data and all the fields are blank. I would be extremely grateful if you could run me through the steps needed to develop a database, accessible via the web.'

Visual Café's data access is via JDBC, the standard Java database API.

Microsoft has not implemented a JDBC driver for Access but there is an ODBC driver which opens up Access databases to any application that can use ODBC.

Sun provides a JDBC-ODBC bridge so you can use ODBC from Java. Sounds good? Well, it isn't and Mark has discovered one of the problems.

First, here is a look at the Visual Café databound project wizard [Fig 3].

• Step one is to choose a project type, the options being Applet, JFC Application, JFC Applet, or Application. JFC stands for

Java Foundation Classes, also known as Swing and the JFC options are much better looking. The

problem is that Swing is not fully integrated with the JDK (Java Development Kit) until version 1.2, which most browsers do not support unless you use the Java Plug-in

available from Sun. Worse, at the time of writing and despite having 'JDK 1.2' printed on the box, Visual Café 3 only really supports JDK 1.1.7 — so, for an easy life choose the Applet.

• Next, choose a data source. If you have not yet set it up, now is the time to configure an ODBC data source for your Access database. Use the 32-bit ODBC administrator in Control Panel and add a new data source, choosing the Microsoft Access driver. On Windows NT set this up as a System data source. The Access driver raises a dialogue with a Select button and here you can choose the .MDB file containing your data [Fig 4].

• Back in Café, choose Define another data source and you will find your connection listed in the JDBC-ODBC Bridge section. Select it and then select the table or query you want to use, ignoring the Access system tables with strange names like MSysAccessObjects.

• Next, Café asks you to choose the columns or fields from the table. Another step lets you assign components to each field. Visual Café

tends to default to TextArea but TextField is more suitable for most fields.

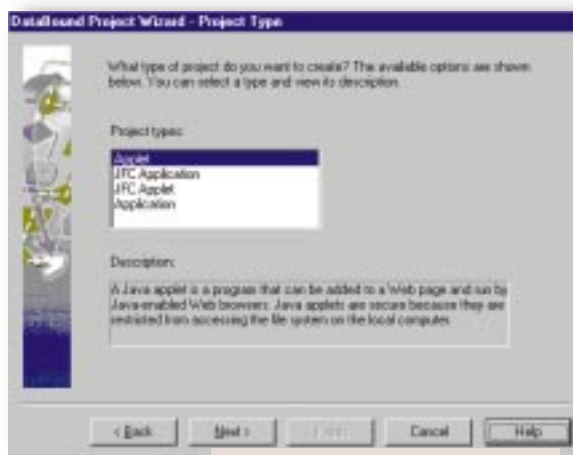
• When you finish, Visual Café builds an applet that has basic functionality including navigation, search, new and save [Fig 5].

This should run fine in the AppletViewer but applets are for web pages. To create a host web page you can use Visual Page as bundled with Café, or do it yourself in Notepad. As a minimum, you will need a line like this:

```
<APPLET CODE="PCWApplet.class" WIDTH=500 HEIGHT=370>
</APPLET>
```

(Key: ✓ code string continues)

You can get the width and height from the applet properties in Visual Café. Add the web page to the project using Insert -> Files into project. Now open

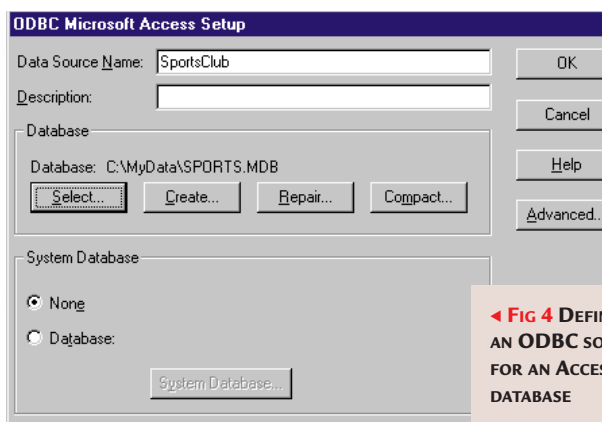


▲ FIG 3 THE DATABOUND PROJECT WIZARD IN VISUAL CAFÉ OFFERS A CHOICE OF FOUR TYPES

Project options and check the option to Execute in default web browser. Run the project and the form displays but the fields are empty, just as Mark discovered.

➤ What's wrong with the bridge?

The reason is a problem with the JDBC-ODBC bridge. This is no use for unsigned applets because it calls ODBC DLLs that run on the client, violating applet security. There are other reasons not to use it. The ODBC libraries are native code so cross-platform support goes out of the Windows. Its other major limitation is that the ODBC data source needs to be set up on each client. People choose Java to avoid exactly this kind of problem so it really is the worst of both worlds. Finally, it has a reputation for bugs and is unlikely to figure highly on Sun's fix-list.



◀ FIG 4 DEFINING AN ODBC SOURCE FOR AN ACCESS DATABASE



➤ The fix — run on the server

The solution is to call the ODBC libraries on the server rather than on the client. One way to do this is by using dbAnywhere, intermediate software which has a JDBC driver and which comes with Visual Café in the database edition. Your applet calls dbAnywhere, and dbAnywhere calls ODBC. Converting the example is simple.

1 First, run dbAnywhere. Next, run the Configure Data Source tool also found in the dbAnywhere group on the Start menu. The key fields in the dialogue are Engine, which must be set to Microsoft Access, and Server, in which you enter the name of the ODBC datasource. Make a note of the port number in the URL (usually 8889). Save and test the data source, close the configuration tool and switch back to Visual Café.

2 The next step is to edit the JDBC connection in your applet to use the new data source. Select the jdbcConnection object and edit the URL property. This opens the Insert Datasource dialogue. Choose New, select the Symantec jdbc driver, and enter the host name and port number for dbAnywhere. Note that you cannot use the default localhost for the IP name or address of dbAnywhere otherwise it will not work across a network.

Click Refresh to show the available data sources and choose the one you've tested. Run the applet again and it should display with a live connection to the MDB [Fig 6]. Note that to deploy this applet to a web server you need to ensure

that the browser can find all the necessary Java classes. Use Visual Café's deployment feature to create a jar (Java Archive) and add a corresponding ARCHIVE attribute to the APPLET tag.

3 The key step is to check the option Create Standalone Archive in the Project Option, Deployment tab, so that all the classes additional to the standard JDK are included.

The main snag

with Visual Café is its instability. Already Symantec has a 43Mb patch which takes the database edition 3.0 to 3.0a. But even with this installed, the IDE is not robust on our test system. From time to time it crashes on startup, the fix being to start Visual Café from the command line with the '-clean' parameter which forces it to delete and rebuild various configuration files.

■ More about Mutex

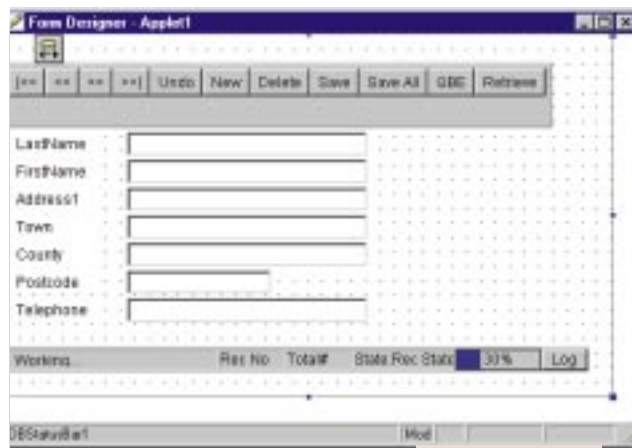
Several readers have commented on the code in the March issue column which prevents multiple instances of an application.

The code was for Delphi but adapts easily to Visual Basic as reader Malcolm Whapshott emailed to tell me. Just to

recap, the idea was to call OpenMutex to retrieve the handle of an application-defined Mutex object. If the function returns zero, the Mutex is not defined so no previous instance is running. Next, the code calls CreateMutex so that other instances attempting to start will find that the Mutex object does now exist and will exit quietly.

Reader John Bain describes 'a slight problem' with this approach: 'There's a potential race

condition between the OpenMutex calls. If two instances start at almost the same time there's a chance that both might call OpenMutex and have 0 returned before the first one gets to CreateMutex. CreateMutex will then return a valid



▲ Fig 5 THE GENERATED DATA ACCESS APPLET

handle for both CreateMutex calls and you will have two instances running.

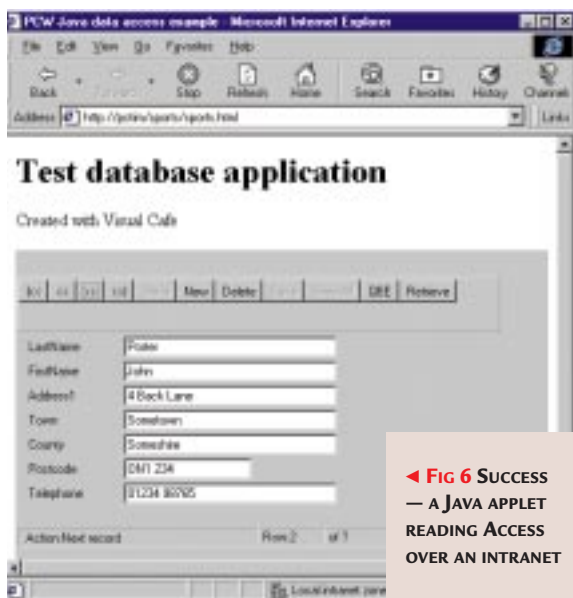
'The way around this is to exploit the fact that if you call CreateMutex with the name of a mutex which already exists, then a valid handle to the existing mutex is returned and GetLastError() returns ERROR_ALREADY_EXISTS. So, you just skip the OpenMutex calls and go straight to CreateMutex and then check GetLastError(). Only one instance can create the mutex so only one can have a GetLastError() other than ERROR_ALREADY_EXISTS. Hence no race condition.'

Another reader, Sam Edge, came up with a similar suggestion. Thanks to both of you and also for some other comments about instance detection and data sharing, for which I will try to find space in a future column.

PCW CONTACTS

Tim Anderson welcomes your feedback on Visual Programming and can be contacted via the PCW editorial office (address, p14) or email visual@pcw.co.uk

Crystal Reports 7 costs £363.07 (£309 ex VAT), or £139.83 (£119 ex VAT) for the standard edition from Contemporary Software 01344 873434. The web report server, automation interface and some other features are only available in the Professional version. More details at www.seagatesoftware.com



◀ Fig 6 SUCCESS — A JAVA APPLET READING ACCESS OVER AN INTRANET