

Steve Scott's Column

Big Changes in Developer Land

The trouble with being a resident commentator (that's what I am according to the bio our esteemed editor once wrote for me) is you have to have something to comment on. Once you have a subject to comment on, you need to have a reasonable understanding of it in order to have an opinion you can defend when people challenge you (and someone always will). You also need to have an understanding of your subject's alternatives so that you can make comparisons. Modern technology is large, complex and not easy to get to the bottom of, meaning the research you have to do in order to form a reasonable opinion can be substantial and very time consuming. Fitting this research in with normal working pressures and deadlines can become quite a challenge.

Just consider the current situation. We have three major PC platforms either in existence or about to be. Firstly we have Windows. OK, I'm reasonably comfortable with Windows but would not for a single minute pronounce myself a guru. I still have plenty to learn. Then there is Linux. I worked with Unix for about two years in the late eighties. When I installed Linux I thought, "this will be easy". Oh boy, how deluded can you get? The learning curve is enormous. You don't realise how much you know about one platform until you try to use another. Then there is .NET. Yes .NET has to be considered a platform; basically it's the Windows API of the future. (I will cover what .NET is at our London meeting in March.)

Once you have established some knowledge about the platform, what about the services or facilities that are available - XML, SOAP, COM+, CORBA? Not to mention the development tools that run on that platform, such as Delphi, Kylix and Visual Studio. How do you know you are recommending the right tool for the job if you don't know what all the tools are capable of? As you can see, our current development landscape is growing rapidly.

If you have struggled over the past few years to keep up with technology, then you'd better brace yourself because 2001 is going to be a biggie. If you don't make an effort this year to at least maintain a minimal understanding of the different technologies and platforms that are emerging and expanding, then you may find the gap just too big to leap next year. We will do our best in these pages and at our meetings to keep you up to date, but unless you put some time into it yourself, you may find you're being left behind.

Kylix

By time you read this, Kylix should be shipping. I sense that Borland have found Kylix a tougher prospect than they may have first thought, and have been struggling a little to produce a clean Delphi clone. We at first thought we might see it last summer, then in the autumn, then the end of the year and then the spring. I remember the first version of JBuilder had a similar sliding release date. In the end, I felt that Borland, probably under commercial pressure, released



JBuilder 1 too early and left the product with a "runs like a dog and crashes" reputation it's only now shaking off. Borland very clearly stated that Kylix would not ship until it was ready, despite greater and greater pressure to get it out of the door.

The other issue is who is actually going to use it. When Borland first announced that the product would now actually be released as Kylix rather "Delphi for Linux" I thought that this might be a mistake, but now on reflection I have decided that this is probably a good move. Based on threads on our own newsgroups and others, I think the actual interest in Linux from Windows developers is lukewarm at best, while at the same time there is a large Linux developer community out there just waiting for a decent development tool to arrive. By not tying Kylix to its Windows sibling, it will probably fare a lot better in this community than it would have otherwise.

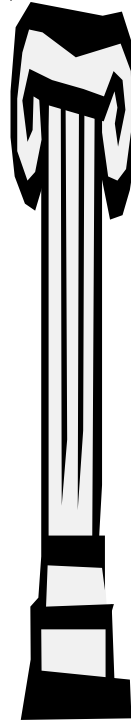
Objects of our desire

I was chatting with Brian Long recently (who is available for training and consultancy at very reasonable rates by the way) about future sessions for meetings. We agreed that finding new subjects to cover was getting to be more of a challenge each month. While I'm on this subject, by the way, we really need your input into what you want to see at meetings. We may feel we have covered a subject when you think it needs more sessions, or we may have missed areas that you think are vital. Anyway, back to my chat with Brian (did I mention Brian was available for training and consultancy at very reasonable rates?).

We discussed how there were probably loads of objects tucked away in the VCL that we use all the time yet have never really investigated fully. We mentioned a few and then Brian jokingly (I think he was joking) suggested TMouse. Well, why not? I thought. I can't quite get a 90-minute meeting session from our rodent encapsulator object but why not give it the limelight with a paragraph in these hallowed pages? In fact, I feel a regular column could be possible here. So if you have a favourite object you would like featured in my new "A Brief Moment of Glory For" column then please just email me the name of the object you would like me to cover along with your current credit card details.

A Brief Moment of Glory For: TMouse

TMouse is a direct descendant of Tobject. It has eight properties and three methods (including Create & Destroy) and makes its nest in the controls unit. What you may not realise is that TMouse has actually grown over progressive releases of Delphi and has become a fully 21st century compliant rodent with three properties supporting the now almost standard mouse wheel. I won't give an in-depth



description of the WheelPresent property as this would take far too long, and I will skip the RegWheelMessage property, as it is not supported in Win 98 or WIN NT. That just leaves us with the Wheel ScrollLines property, which may be of some interest to our geekier readers. WheelScrollLines returns the number of lines that are scrolled for each notch that the mouse wheel is rotated. The WheelScrollLines can be set under Mouse in the Windows Control Panel. I think this piece of information may revolutionise your use of TMouse.

I Told You So

If you remember, in my last column I said that, on the basis that most government based projects are a least a year late, we should be expecting the public sector to start experiencing the millennium bug sometime this year. Just a few days after your mag hit the doorstep, I received an email detailing how all the trains on Norway's national railway had refused to start on the 31st of December 2000 because of a date based computer bug. The problem was temporarily solved by putting all the onboard computer clocks back to the 1st of December 2000.



When I posted this story on the newsgroups several people replied with other similar stories, including an incident with the 7-Eleven chain of stores in the US. Apparently all their cash registers thought it was now 1901 and would not accept credit cards issued after that date. A 7-Eleven spokesperson said the system had been "tested for Year 2000 compliance". Obviously the testing group took their brief very literally and decided "Year 2001 Compliance" wasn't their problem.

Community Makeover

Finally, I would just like to congratulate Borland on the makeover they have given their community web site at <http://community.borland.com>. The site looks great and works well. The first time you visit, though, things may look a little weird. You may find you get the new site layout with a number of the old style icons and buttons making what Borland have called "a very funky looking home page".

The problem occurs because many of the new images on the site have the same name as their old counterparts. Your browser sees the image, realises it has one of the same name from the same location in its cache and tries to save your telephone bill by using the cached version. If you enjoy funky experiences then have fun, otherwise just press Ctrl-Shift Reload to refresh the images. If this doesn't work then maybe you're just too funky for your own good.

[Please send definitions of "Funky" to the Editor.]

Steve Scott is director of Streambyte Ltd, based in Gloucestershire. He is a developer, trainer, conference speaker, UK-BUG's London Group Leader, and father of lots of small(ish) children. You can contact Steve on sscott@streambyte.com.

He also appears to have formed the official Brian Long fanclub!



How to get your current drive mapping programmatically

Create a function such as:

```
function GetNetworkDriveMappings( sl :
TStrings ) : integer;
var
i : integer;
sNetPath : string;
dwMaxNetPathLen : DWord;
begin
sl.Clear;
dwMaxNetPathLen := MAX_PATH;
SetLength( sNetPath, dwMaxNetPathLen );

for i := 0 to 25 do begin
if( NO_ERROR = Windows.WNetGetConnection(
PChar( '' + Chr( 65 + i ) + ':' ),
PChar( sNetPath ), dwMaxNetPathLen ) )
then
begin
sl.Add(Chr( 65 + i ) + ':' + NetPath);
end;
end;

Result := sl.Count;
end;
```

Then add the following to an event, in this case a button click:

```
procedure TForm1.Button1Click(Sender:
TObject);
var
sl : TStrings;
nMappingsCount, i : integer;
begin
sl := TStringList.Create;
nMappingsCount := GetNetworkDriveMappings(
sl );

for i := 0 to nMappingsCount-1 do
begin
MessageBox( 0, PChar( sl.Strings[ i ] ),
'Network drive mappings', MB_OK );
end;
sl.Free;
end;
```