THE DOT NET EXPERIENCE

Phil Robbins
Auckland University of Technology'
Auckland, NZ
Phil.Robbins@aut.ac.nz

Proceedings of the 16th Annual NACCQ, Palmerston North New Zealand July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

ABSTRACT

Visual Basic has been taught at AIT/AUT since 1994. Version changes have occurred at regular intervals, each requiring some changes to classroom installations and to teaching material. The latest upgrade, to Visual Basic .Net (usually pronounced Dot Net) has been the most challenging yet. Not only has the language changed significantly but the environment now requires permissions and settings not required in any other software that we use!

1 INTRODUCTION

I have taught programming at AIT/AUT since 1994. Although I have used C, QBasic and Prolog, the bulk of my teaching has been with Visual Basic and Delphi. With Delphi we have used versions 3 to 6 and are currently using version 7. With Visual Basic it is a similar story, although we did not use version 5 for some reason. The latest version, 7, is part of Visual Studio.Net.

The process of migrating from one version of a language to another has followed a familiar pattern:

1. Buy a licence.
2. Install the new version on my machine.
3. Test the language mainly by completing a selection of class exercises.
4. Update the documentation for the course (lab exercises and teaching notes).
5. Install the version on the classroom machines.
6. Let the students loose!

The migration from Visual Basic 6 to Visual Basic.Net has been the most challenging yet. This paper outlines our experiences at AUT in the hope that others making the move will be benefit from them.

2 THE LANGUAGE

Microsoft has said that Visual Basic has been completely rewritten for the .Net platform. As an experienced user of the language this is quite obvious. I have felt as though I was learning a new language rather than a new version of VB.

2.1 Object Oriented Programming

Although some people have always claimed that Visual Basic was an object oriented language, at AUT we have never accepted this. The lack of inheritance has meant that "object based" has always been a more acceptable description.

A look at the code behind a newly created form is the first evidence that something has changed - as with Delphi, a Visual Basic form is now clearly a class.

```
Public Class Form1
    Inherits
System.Windows.Forms.Form
```

For my first VB.Net program I took an exercise we do on our BInfoTech degree, a shape drawing program which uses an abstract BasicShape class and derived Rectangle, Circle and Triangle classes. This was successfully implemented, something that would not have been possible in earlier versions.

2.2 Graphics

I noted in passing that the method of drawing to the screen had changed. It is no longer a case of using methods of the form, rather of the form's GraphicObject object.

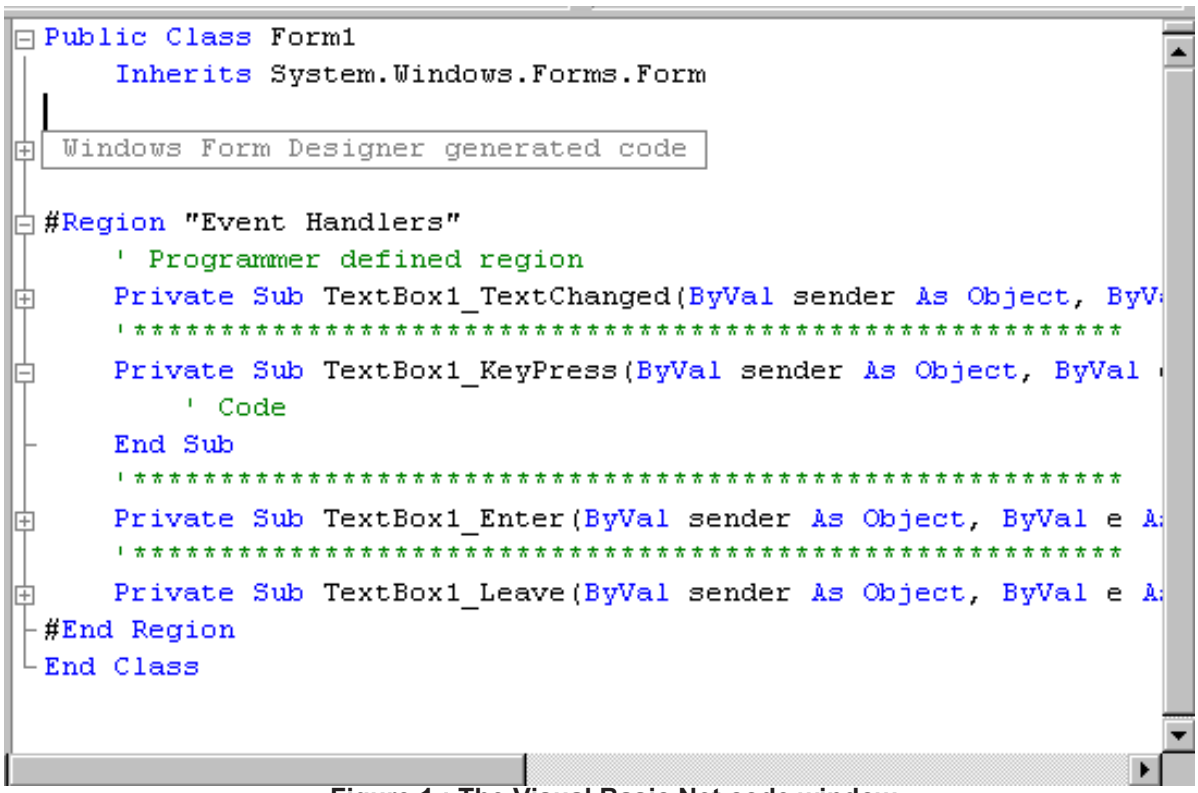

Figure 1 : The Visual Basic.Net code window

"Windows Form Designer generated code" is a region that comes with every form, hiding over 60 lines of automatically generated code.

2.3 Properties

Manipulating the properties of controls and other objects is a major part of Visual Basic programming. Changes to these have to be learned and can lead to frustration. They also make it harder to convert existing programs to the new version. Here are some of the more obvious changes.

♦ There is no longer a Caption property. In Delphi and in all previous VB versions, static text in a label, button or other control has been referenced by the Caption property in contrast to dynamic text in a text box, which the user can edit at run time. Now both are controlled by the Text property.

♦ Some properties have become read only. I had an exercise where check boxes were used to control some of the features of the font of a text box. When the user clicked on the "bold" check box, for example, the text box Font.Bold property was reset. This is no longer possible - the font is read only and has to be completely recreated.

2.3 Events

In an event driven environment like Visual Basic, all code must be written in or called from an event handler sub. Once familiar events have now changed, some being completely renamed.

♦ All events now have a "sender As Object" parameter, with "e" as a set of the appropriate event arguments.

♦ Change is now TextChanged, GotFocus is now Enter, LostFocus is now Leave.

♦ KeyPress now presents the key code as e.KeyChar which is read only. Cancelling the key stroke is done by setting e.Handled to True.

♦ Control arrays have gone, but event handlers may now be shared by adding to the list of events the sub handles.

```
Private Sub
 TextBox1_TextChanged(ByVal sender
 As Object,  _
ByVal e As System.EventArgs)
 Handles TextBox1.TextChanged
```

2.4 Database Connection

Managing a database with a Visual Basic front end has been part of all our Visual Basic courses. This

has ranged from simple display applications with entry level classes to n tiered applications with the more advanced.

It would be fair to say that the changes made in this area have been the most challenging! DAO, the data control, the Data Environment have all gone. ADO has been replaced by ADO.Net. There are a number of data access components which may be used at design time, but it is also possible to create these in code and not use the drag and drop technique.

Having done a lot of work connecting Visual Basic to databases, mainly MS Access, I find myself having to learn how to do everything again with .Net. In this respect more than any other it feels like a completely new language.

2.5 IDE

Tabbed windows are now the norm, the free floating SDI interface of

VB6 having disappeared. The code editor now has regions with + and - buttons for showing or hiding their code (see figure 1). Some are built in, such as those that are attached to a sub or function, but you can create your own..

The code window still displays the names of all controls in a project, but they are no longer sorted alphabetically! The “hidden” code contains a list of these controls which may be edited to put them in any required order. Interestingly, controls added after coding has begun are listed at the end of the file, not with the original controls list!

3 THE ENVIRONMENT

3.1 Debugging Rights

Our biggest challenge has been to provide Visual Basic.Net in the student labs. I made the mistake of testing the classroom installation with my tutor login and assuming that, because I could use it, all would be well. My first class showed me how unwise that was - students could create forms but could not run their applications because they had no debugging rights. My only excuse was that we had not had to do this with any previous versions.

Debugging rights are available to those logging in with administrator rights (Windows 2000+), or to those in the VS Developers and Debugger Users groups. As our students did not have administrator rights and had not been put into the groups, they could not debug (ie run) their programs. My tutor login gave me administrator rights.

At AUT we use Novell Netware 6, with ZenWorks 3.2 to manage the workstations. Students get a local Windows NT/2000 account created at the time they login by Novell Workstation manager, which has the same login and password as their NDS (Network) account. When the student logs out, their local user name and password are automatically deleted to prevent the PCs in classrooms from filling up with users and profiles.

We were able to place a student in appropriate groups on a machine as they login and remove them when they logout. Unfortunately, Windows 2000 demanded that, after a student had been placed in a group they have to login again for this to take effect thus, on our system, taking them out of the group again! The solution was for the student to login to the workstation only as the generic “student”1, be put in the debugging groups, then login to the network to access their files. Most students coped with this straight away, only a few needing a couple of reminders!

3.2 ASP.Net

One of the new features of Visual Basic.Net is the ability to write ASP.Net web applications. The text I was using as a guide showed which buttons to click in order to create such an application. It made no mention of the various error messages I would encounter on the way.

It took me a while to discover that I had to move from Windows NT to Windows 2000, that I had to have IIS installed (Microsoft’s Internet Information Services), that this required W2K service pack 2 and that it all had to be done before the .Net framework was installed! This was a useful experience. As part of the MS Academic Alliance, we are allowed to loan students Visual Studio.Net disks for them to install the software on a machine at home. Knowing what was involved in the installation saved many of them a lot of frustration.

In our lab we had to make additional changes to allow students to run ASP.Net. Applications have to attach to ASPNet_wp, something not permitted under a student login. This again had to be set up under the generic “student” login.

This is achieved by actually running the ASPNet_WP.EXE from the generic student login, so it owns the process, then that student login can attach to the process to debug. The details in the Microsoft Knowledgebase suggested that the only way to debug ASP.NET applications was for the user to have administrator rights. Fortunately our technical staff were forwarded a word document from Microsoft ‘University Relations’ which detailed how to do this.

3.3 Network Folders

Although most software is located on the hard disks of our lab machines, students store their files in the network folders and typically debug them from there. The default setting of Visual Basic makes it regard all network drives with suspicion, resulting in security violation errors when many student programs were run. This was eventually traced to a single line in the machine.config file which, when changed, solved the problem.

ACKNOWLEDGEMENTS

Ben Yeldon, the Software/Desktop Standards Officer of AUT's Technology Services Group has done the hard work in setting up the classrooms for VB.Net. He has contributed technical details to this paper.

Paul Andrew and Pradeepa de Silva of Microsoft have provided some support during the process of setting up Visual Basic.Net at AUT.

REFERENCES

Bradley J C and Millspaugh A C, "Programming in Visual Basic .Net", McGraw-Hill.

Note: 1 The generic "student" login is a local workstation account that doesn't get deleted at logout. This was created when the PCs were setup, and isn't managed by Novell Workstation Manager.