

Clipper is not dead! The rise of managed Clipper or Vulcan .NET

December 1, 2009

A few months ago I got a request from the owner of a small business to help out in porting a legacy application to a .Net WinForms. I thought that it probably would be some old VB6 code or ASP page that needed a new look and the conversion would be pretty straightforward.

To my amazement I was confronted with an old **Clipper application** running against a **dBASE III database** on **MS DOS 6.1**. I have seen a lot of legacy so I'm not easily startled, but this was going back to the eighties and my first encounters with a computer.

The owner underlined that it was a business critical application and that he thought it was about time to invest in some newer technology. It appeared to be an application responsible for resource planning i.e. assigning technicians to appointments with clients.

My first response was to have a look at cheap CRM or basic planning-and-calendar software. But I couldn't convince him. He was so used to working with his old software: "It provided everything that he needed!".

So my quest started:

- Is there a dBASE version that runs on Windows Server 2003?
- Are there any drivers capable of reading dBASE III *.dbf files?
- Are there any conversion tools for Clipper source code?

Apparently the Clipper community is still alive and kicking. Clipper, created in 1985, was embraced by CA and in 2002 they released **CA-Clipper**. Digging further I also found an open source project called [Harbour](#) supporting Clipper but in the end [Vulcan.NET](#) saved my day.

Vulcan .NET integrates with Visual Studio as any other [.Net language](#) and is [fully CLS compliant](#). (To get started have a look at [Vulcan.NET At Warp Speed Guide](#))

About eighty percent of the legacy Clipper PRG files, I imported directly without any problems. So a lot of the 'business logic' was saved and compiled into a separate managed DLL.

The old database tables were accessible through the VulcanDB database driver. Without any major glitches the content was recovered.

Writing the front-end was reduced to choosing any modern .NET technology. In my case, just adding some ASP. Net pages that call the converted clipper methods.

With minimal effort a lot of the existing code and content was recovered in a fresh .NET environment. Although the project was terminated here, I would personally have gone the extra mile by converting the database to SQL

Server and by rewriting the Clipper code into some more wide-spread language (C# for example).

Since the owner was happy, I'm happy but I think an of-the-shelve package would have saved him a lot of money.

www.vanderbist.com