



30 Years Sustained Prosperity: JPEG and Integrated Development with Visual Studio and RAD Studio

One year ago, in October 2021, the Independent JPEG Group (IJG) celebrated the 30th anniversary of the most successful electronic image format JPEG.

In order to be prosperous, a file format intended for common image interchange must be supported by as many applications, software or hardware, as possible. Software applications with a Graphical User Interface (GUI) are readily created in an Integrated Development Environment (IDE), combining several components of the development process in a consistent framework. Today we find two major IDEs for graphical application development which can look back on a history almost as long as the history of the JPEG image format, and therefore going to celebrate their 30th anniversary in the near future: Microsoft Visual Studio/Microsoft Visual C++ (MSVC) and Embarcadero RAD Studio (Rapid Application Development)/C++Builder/Delphi. Both were first released in 1997 and will celebrate their 30th anniversary in 2027 (Delphi is actually two years earlier – first released in 1995 and 30th anniversary in 2025).

Apart from the long history of their existence, these two Integrated Development Environments have another feature which makes them stand out in the large quantity of available development tools. In fact we can find a lot of different compilers, assemblers, editors, debuggers, GUI frameworks and configuration platforms. But Visual Studio and RAD Studio are the only environments where all necessary components for development from the lowest level (machine language/assembler) to the highest level (APP/GUI framework) are provided from a single source with optimal consistency. It is clear that Microsoft, as the vendor of the dominant Operating System for Personal Computers for decades, needs an appropriate assembler for machine language programming. An Operating System provides an interface between the hardware components of a computer system and the user applications to make the user applications independent from the details and changes in hardware components. User applications can therefore use higher level programming languages and are not constrained to operate on the low machine level. However, the Operating System must use machine language (assembler) to control the hardware on one side of the interface. So Microsoft developed their own assembler with a powerful macro system, called Microsoft Macro Assembler (MASM), and first provided it as a separate product. Later it was supplied with Microsoft Windows Software and Device Development Kits and it was included with Microsoft Visual Studio/Microsoft Visual C++. User applications prefer to use higher level languages like C/C++, and Microsoft provides the Windows SDK (Software Development Kit) as an interface between the application and the Operating System for the higher level languages. A corresponding interface on the assembler level also exists but is not supplied by Microsoft. It is called MASM32 SDK and provided by Steve Hutchesson (Hutch).

In March 1971 the Swiss computer scientist Niklaus Wirth published the paper “The programming language Pascal” and thus officially launched the Pascal programming language. It was named after the famous French mathematician Blaise Pascal of the 17th century.

In 1983 the French mathematician Philippe Kahn founded the company Borland, after moving to Silicon Valley, California. Looking for a handy name for the company, he was inspired by the name of

US astronaut Frank Borman. Borland gained huge success with an Integrated Development Environment for the programming language Pascal on the CP/M and MS-DOS Operating Systems, called Turbo Pascal. Niklaus Wirth writes in an article titled "50 Years of Pascal" for Communications of the ACM in March 2021: "Philippe Kahn at Borland Inc. in Santa Cruz surrounded our compiler with a simple operating system, a text editor, and routines for error discovery and diagnostics. They sold this package for \$50 on floppy disks (Turbo Pascal). Thereby Pascal spread immediately, particularly in schools, and it became the entry point for many to programming and computer science."

In 1995 Microsoft launched their Windows 95 Operating System. Turbo Pascal, renamed in the meantime to Borland Pascal, is further developed to Borland Delphi. The programming language is further developed to Object Pascal, and components are integrated to support the new Windows 95 GUI programming paradigm, thus establishing the Rapid Application Development (RAD) concept. The name Delphi was first only the internal codename for the project during development at Borland. One of the major features should be database connectivity, and Oracle was an important database system. So Danny Thorpe at Borland came up with the Delphi suggestion, referring to a place in ancient Greek mythology: "If you want to talk to [the] Oracle, go to Delphi." The release name was then determined to be AppBuilder, but another company Novell published a product called Visual AppBuilder, and so eventually the Delphi codename was used as the product name.

Microsoft offered their Visual Basic product for the purpose of Rapid Application Development, but it was technically inferior compared to Borland Delphi in several respects. In order to improve their position in the field of software development tools, Microsoft hired the main developers from Borland for their own software development tools division. The most prominent case is Anders Hejlsberg, who was the original author of Turbo Pascal and chief architect of Delphi. At Microsoft he led the development of C# and TypeScript. After losing the main heads, the Borland management lost interest in their software development tools division and formed a spin-off called CodeGear, which was then sold to Embarcadero Technologies.

In the 1970s another programming language, "C", was introduced, which eventually became more popular than Pascal and which dominates software development to date, together with its extension C++. Both Borland and Microsoft recognized the growing popularity of C/C++ and offered corresponding development tools during the 1980s and 1990s: Turbo C, Turbo C++, Borland C++, Microsoft C/C++ and Visual C++. The development culminated in 1997 with the release of Borland C++Builder, using the same Rapid Application Development/Integrated Development Environment as Delphi, and Microsoft Visual Studio/Microsoft Visual C++, integrating the software development with other Microsoft development tools (particularly Visual Basic and later .NET) in a single IDE.

In order to cover the whole spectrum of software development requirements from the lowest to the highest level, Borland also published an assembler tool, called Turbo Assembler (TASM). It is still included in current versions of Embarcadero RAD Studio with Delphi and C++Builder, as TASM32 Version 5.4 of 2010. Development seems to be stalled, since Borland/Embarcadero don't have the requirement for Operating System development as opposed to Microsoft, and inline assembly options exist for limited use in applications. Turbo Assembler has similar powerful macro capabilities as Microsoft Macro Assembler (MASM). A unique feature of Turbo Assembler, which is usually only available in higher level languages, is the support of the object-oriented programming (OOP) paradigm, via the use of method procedures and virtual method tables. This feature also helps interfacing assembler modules with Delphi (Object Pascal) and C++ in mixed language projects.

On the high level side of development tools, Borland/Embarcadero RAD Studio provides the RTL (Run-Time Library) and VCL (Visual Component Library). They are written in Object Pascal and can be used by both Delphi and C++ applications.

Microsoft Visual Studio/Microsoft Visual C++ offers the ATL (Active Template Library) and MFC

(Microsoft Foundation Classes). MFC was initially called "Application Framework Extensions" and abbreviated "Afx". That's why one can still find many places in MFC where names contain the afx shortcut.

The Microsoft Windows Operating System contains several components providing JPEG image format support, basically Load/Save operations. The functions can be used via the Windows SDK provided by Microsoft, or via certain components/classes of VCL/MFC.

Windows has JPEG support in the OLE (Object Linking and Embedding) system, in the GDI+ (Graphics Device Interface Plus), and in the WIC (Windows Imaging Component).

The CImage class of MFC provides simple Load, Save, and Display functions of JPEG images for applications, demonstrated by the SimpleImage MFC sample application provided by Microsoft. The CImage class uses the GDI+ functions of Windows.

VCL has the TJPEGImage class to provide JPEG support for RAD Studio (Delphi and C++Builder) applications. It can be easily demonstrated by using the ImageProc sample application provided by Embarcadero. The TJPEGImage class is implemented in Object Pascal and using the old Independent JPEG Group libjpeg 6b implementation of 1998. The Pascal and IJG sources are also provided.

The disadvantage of both the Microsoft and the Embarcadero offers is that they only provide obsolete JPEG support based on the development status of 1998, thus decades ago.

If you are writing applications for Jane Doe and Joe Average, you might get away with this offer.

But if you want to write programs for Linda Brighthouse and Ben Wiseman, you have the option to use the current IJG source code for up-to-date JPEG support in your applications. This is already done by reference applications like cPicture, PhotoLine, Directory Opus and StudioLine Photo. Independent JPEG Group provides up-to-date build systems for current versions of Microsoft Visual Studio and Embarcadero RAD Studio for use with C/C++ applications. The Delphi (Object Pascal) integration is also supported with special configuration options.

Author:

Guido Vollbeding, Institute for Applied Informatics (InfAI, infai.org),
Independent JPEG Group (ijg.org, jpegclub.org/reference/)

Leipzig, October 2022