

Building Robust Imaging Components for the Microsoft .NET Platform

Overview

Pegasus Imaging Corporation has provided industry-leading digital imaging software development toolkits to application developers since 1991. During that time, significant efforts have been taken to ensure that products delivered are the fastest and most robust available in the market. For many Pegasus Imaging products, execution speed for each targeted delivery platform is optimized through algorithmic enhancements and coding techniques specific to that platform. In the case of products built for the Wintel platform, execution speed is optimized through extensive use of assembler code using the advanced MMX and SSE processor instruction sets. Pegasus Imaging products maintain a significant speed advantage by using these approaches, and execution speed remains a major advantage over competing products.

With the introduction of the Microsoft .NET Framework, the rules traditionally associated with component development have changed dramatically. Components, and the applications that use them, are written in managed space and require a version of the Microsoft .NET Framework's Common Language Runtime (CLR) to run on a host machine. When managed source code is compiled, the result is Microsoft Intermediate Language (MSIL) instructions, and not the machine-language instructions created during a Win32-based compile. To convert MSIL to machine instructions, MSIL is just-in-time compiled into machine code and run against the CLR. The result is a programming platform that is cross-platform compatible and processor neutral, assuming the CLR installed on the host machine can convert MSIL to machine code. However, this flexibility comes at a cost; in the imaging world, this cost is significant.

Pegasus Imaging customers (digital imaging application developers) will not be successful marketing applications with substandard performance. In particular, imaging systems require extremely fast compression, decompression, and processing operations, while maintaining a small memory footprint during execution. Unfortunately for digital imaging applications—the execution speed of managed code running against the CLR suffers when compared with the execution

speed of hand-optimized assembler code using MMX and SSE extensions on the Wintel platform. Additionally, cleanup of non-garbage-collected resources suffers greatly without implementation of deterministic destruction. In an article written by Eric Gunnerson of Microsoft Corporation, it was demonstrated that having 100% safe code (code that does not directly access memory locations through pointers) caused a color to grayscale image conversion to take 14 seconds!¹ A more acceptable speed for this simple operation was only achieved by using “unsafe” code.

Pegasus Imaging's First .NET Solution

We have worked continuously since our first .NET offering in September 2002 to provide powerful solutions while minimizing negative impacts that CLR imposes on digital imaging component performance. The September 2002 release provided .NET versions of all Pegasus Imaging components in a managed form as described in a statement from Chris Lubeck, Pegasus Imaging's Vice President of Development: “With the functionality and performance of the existing Pegasus components as the minimum requirement for a .NET version, we quickly understood that the existing code must be part of the solution. The existing code was in the form of COM objects. We developed a managed component that initializes the COM control and passes properties and methods between the managed interface and the COM layer – without requiring COM registration on the system. We then stored all code dependencies as compressed objects in the .NET component's resources, and wrote a DLL loader that could access and decompress these code resources and execute the code. This allowed the components to have a single file assembly for runtime deployment.”

The solution has been very successful, and was applied in many of Pegasus Imaging's products including its flagship imaging component toolkit—ImagXpress 7. At the time, this method provided the managed solution developers required by application developers as they evaluated the new platform, without limiting Pegasus Imaging flexibility. However, as .NET Framework application deployment continued to increase,

¹ Gunnerson, Eric. (15 November, 2001). *Unsafe Image Processing*. Retrieved November 30, 2004 from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncscoll/html/csharp11152001.asp?frame>

and the evolution of operating environments continued, it became apparent in the fall of 2003 that a more complete solution was warranted. The limitations of the September 2002 solution began to dampen the development experience for some .NET users. Pegasus Imaging responded in a resounding fashion.

Pegasus Imaging's Current Solution: .NET Imaging Without Barriers

Starting in the fall of 2003, Pegasus Imaging engineers began working to develop a component architecture that blends the best features of native execution with managed programming. The result is a way of building components for the .NET and other component platforms that ensures intuitive programming in all environments, without sacrificing features specific to each of them. Imaging technology is built utilizing native, optimized instructions, while each component layer calls into this common layer via internal mechanisms. For .NET developers, each component is now a single managed assembly, implemented the way .NET components are intended, with:

- Interface layers built entirely with managed code using C#
- No wrapping of existing COM objects as had been done previously
- Object-oriented application programming interfaces
- Typed exception handling
- .NET compliant event handling
- IDisposable interface support for deterministic destruction of unmanaged resources such as bitmaps
- API naming that follows the established Microsoft .NET Design Guidelines for Class Library Developers
- Components that verify security permissions through assembly manifest attributes
- A single file distribution maintained in most cases
- Context-sensitive help that integrates with all versions of Microsoft Visual Studio .NET

Finally, all Pegasus Imaging components using our new architecture are tested for compliance with .NET standards using the Microsoft FxCop code analysis tool. Put simply, Pegasus Imaging has developed .NET versions of our world class imaging components that closely follow established standards without sacrificing the performance that our customers require.

The first new .NET components were introduced with the release of PDFXpress™ in November 2004. During 2005, additional components were made available, starting with ISIS® Xpress,™ and culminating with the much-anticipated release of ImagXpress™ 8 in Spring 2006. Each component released will be in the new architectural form, providing the development experience that .NET developers expect.

Support for .NET Framework 2.0

Microsoft specifies that the recently released .NET Framework 2.0 will continue to run programs developed for .NET Framework versions 1.0 and 1.1. Our testing confirms that our tools continue to work fully in this new environment. And our commitment to our customers is to continue to upgrade and improve our tools to take advantage of the latest technologies provided by Microsoft as they stabilize and gain market acceptance.

About Pegasus

Founded in 1991 and headquartered in Tampa, Florida, Pegasus Imaging Corporation delivers digital imaging software development components and image compression and editing technologies. The company's products exceed speed and quality requirements for document imaging, forms processing, medical imaging, photo imaging, video applications and more. Technology is delivered as Microsoft .NET controls, COM controls, DLLs, and applications. Multiple platforms are supported, including Windows®, Linux, Sun® Solaris®, IBM® AIX® and Apple® Macintosh®. For more information about Pegasus Imaging's .NET solutions, or for other product-related inquiries, please visit www.pegasusimaging.com, email sales@jpg.com or call (813) 875-7575.