

# Technology Spectrum



Why .NET chosen for developing  
Exensys Enterprise Product

.NET, the changes at its core represent the biggest shift in software technology since the dawn of Microsoft Windows. .NET is portending a whole new computing model emphasizing not just superficial trading of web pages, but cooperating and collaborating systems. For ERP vendors and their customers, .NET spells the future of enterprise software applications..NET technology points to a future that demands ERP software companies rethink and rewrite their base architectures.

### **The Right Place**

First, web browser-based user interfaces would not provide the rich features and raw speed of native code. Native code (code running locally on the PC) leverages the local CPU to perform much of the computing work, giving the user interface a snappy performance, and a rich set of reactions to a user's mouse clicks and keystrokes. In the lite-client model, most of the work is performed on the server. A web browser interface is analogous to a slide show, where the projector (the server) does the work and the screen (the browser) just hangs there. Clearly, the lite-client processing model mirrored that of the mainframe with its dumb terminals rather than leveraging the nimble and ever more powerful PC.

Also, the "lite-client" architectural model put more strain on the limited IT resources and budgets available in smaller manufacturing plants. It meant ERP customers must use larger server hardware and web server software and maintain IT budgets to keep it all running. They would not be giving up their PCs; web servers would be an additional element to manage.

Finally, and possibly the biggest concern was that moving to a lite-client architecture did nothing to facilitate a rich level of connectivity and distributed processing, the need for which was clearly visible on the horizon. Hardware, networking, and communications companies continue to bring forth smarter and more powerful PCs, hand-held wireless devices, cell phones, and Dick Tracy wrist watches. Lite-client architectures do little to capitalize on these advancements.

.NET is simply a natural and logical fit.

### **.NET Software is Better Software**

.NET is the next-generation computing platform from Microsoft. At the core of the .NET hype is a new layer of software that sits above the Windows operating system. This new .NET platform is a better platform for building and running applications software than the Windows OS that we know today. .NET insulates software developers and computer users from the deficiencies and incompatibilities of operating systems. And .NET provides a new set of tools and prefabricated components of unprecedented power that can be used to write a new type of software called "managed code." Over time, managed code will become recognized as clearly superior because:

Managed code is more robust. The .NET platform not only runs this new type of code but oversees its execution allowing software errors to be caught and halted before serious problems can occur. "Memory leaks," "memory corruption," and "blue screen of death" problems that freeze our old Windows systems in their tracks are prevented by .NET or curtailed before damage is done.

.NET is all about better software. Managed code is faster, smaller and lighter, it's more hacker-resistant, and easier to deploy, install, maintain, upgrade, and if necessary, remove from your computer. Managed code side-steps the root causes of many of the bugs and security problems we suffer from today. Managed code will be cheaper to own and therefore of strategic value in the years ahead.

### **Side-by-side versions:**

.NET software minds its own business. It stays together in one place rather than smearing itself all around our systems. And in doing so, it sidesteps the old Component Object Model (COM) - based mechanisms that allowed software packages to collide with and harm one another. In fact, .NET allows two versions of a managed code application to run side by side on the same computer without any interaction or conflict. .NET thus eliminates the need to remove an old version of a software product just to install and try out a new version.

### **Better security**

Under the old Windows approach to security, software is granted permission to do things on your computer based on the permissions granted to the user who runs the software. Hackers continue to find devious ways to sneak destructive software onto our systems and trick our computers into running that software under a privileged user account. In contrast, managed code follows a security model that is inherently different; the software is granted permissions on its own merits – not those of the user. Attributes of the software itself such as who wrote the software, where it came from, and where it is located are used to dictate and police what the software is allowed to do. The .NET platform enforces the new regime. Managed code can only run on the .NET platform so must submit to this new system of strict supervision and restriction. Certainly hackers will continue their agenda, but the new .NET security model is a promising new weapon in halting rogue software and security breaches. You've heard the old adage that you're either part of the solution or part of the problem. Managed code is part of the solution. Every purchase of old COM-based software is a vote for perpetuating the security problems that haunt the computer industry today.

### **Better connectivity**

The new .NET software development tools and the standards-based nature of managed code make it easier to develop systems that employ state-of-the-art connectivity techniques such as XML Web Services. Our future will clearly be filled with a kaleidoscope of constantly collaborating systems, large and small, networked together, exchanging information and performing services for each other and for mankind. Building this high level of connectivity and requisite robustness demands new components, tools, and a standards-oriented approach. Getting there with old components and tools and without standards is an impossible challenge. .NET comes with a toolset for the next generation of software for the new "connected world."

## **Faster Software, Faster Development, Easier Deployment**

.NET is a completely new software technology, created from scratch, both to leverage groundbreaking technologies such as XML and to eliminate the old inefficient internal layers that have built up over the last 20 years in Windows software. Managed code is lean, speedy, and lightweight. These traits enable new innovation in application development and deployment. And, .NET includes a huge box of prefabricated industrial-strength components ready to use by software developers, allowing developers to save time while creating more robust and powerful applications. Aspects of managed code coupled with industry standards invite new deployment models including “no-touch” deployment and software that updates itself.

.Net are programming frameworks from Microsoft that allows us to build rapid, robust, well organized application code and allow for modular & scalable development of powerful applications. By using ASP.NET you can accomplish common tasks by EDP (Event Driven Programming). Basically the .NET development framework is a development and execution environment that give the flexibility to most programming languages & libraries to work as a group seamlessly for creating Windows platform, web or mobile applications that are easier to build, manage, deploy with stand alone applications or as networked systems.

## **Lower cost of ownership**

IT departments struggle daily to fix problems and work around restrictions rooted in the old Windows COM-based software technology. Companies expend vast amounts of corporate energy in their efforts to prevent new debilitating problems as they endeavor to apply system upgrades. Innovation and progress in leveraging new technology is slowed by severe caution and pessimism in rolling out new changes. The problems inherent in old Windows software translate to higher IT costs and lost opportunity costs. Each of the managed code advantages listed above promise to contribute to lower IT costs due to easier software development, easier and more trouble-free deployment, installation, maintenance, and security.

The advantages of managed code are clear and recognition of this fact is gradually growing in the public consciousness and corporate boardrooms. Just as there came a day when the MS-DOS-based software market was pronounced dead, so will come the day when software purchasers will only settle for managed code.

.NET is a new and better kind of software and a new set of tools to build it. And yes, .NET makes it easier to build and deploy the XML Web Services that everyone is talking about. Not to diminish their importance, but the fact that Web Services are dominating discussion right now is a symptom of a problem many ERP companies face.