



A Path from Windows Desktop to HTML5

GIZMOX TRANSPOSITION: The path to modern enterprise application code

Reduce Risk, Cost, and Time to Market for Legacy App Conversions



This eBook explores the challenges of modernizing legacy desktop client-server applications and introduces Gizmox Transposition as a unique alternative to solving one of the most pressing IT challenges of the “Post-PC” era.

The majority of large enterprises have applications running on legacy desktop client-server platforms like VB6. While still deemed “mission critical,” these applications are an obstacle to realizing the advantages of a modern IT infrastructure.

With Windows XP going out of support, the security benefits of moving to Windows 7 and 8, and a general need to consolidate server infrastructure, Enterprise IT organizations are demanding that legacy applications be retired or redeployed on modern application platforms.

Furthermore, enterprise mobility initiatives and broad acceptance of Bring Your Own Device (BYOD) are defining new demands for enterprise application platforms. The days of managing mission critical business processes with applications that only run on desktop computers is over.

INSIDE:

Challenge: Security	03	Benefits: Developer	08
Challenge: Application Infrastructure Management	04	Benefits: Enterprise IT	09
Challenge: Supporting BYOD and Mobility	05	Benefits: Line of Business	10
Alternative Approaches to App Modernization	06	Learning More about Transposition	11
Gizmox Transposition	07		

“Legacy apps are IT security risks”

Legacy desktop client-server apps pose challenges to IT security initiatives and can actually represent security vulnerabilities.

Upgrading to the latest Windows operating systems is essential for reducing security risks for 2 reasons: 1) Win7 and Win8 have more and better security features and 2) Discontinued support means no more security patches. This makes older operating systems more vulnerable with time. The same can be said for upgrading specific components of the Microsoft platform (e.g. Internet Explorer, MS Office, Active X, older .NET, IIS, etc...)

Frequently, custom-developed client-server apps written in older versions of .NET or VB6 will not run on newer operating systems or may depend on a specific version of some component. While the application risks may be well understood, rewriting custom apps can take years, cost millions, and even introduce new security risks. This often results in a stalemate: Business application owners pushing for extensions to operating system upgrades while they develop more secure replacement applications; IT organizations pushing to reduce security concerns by upgrading to new operating systems.

The right approach is to redeploy enterprise applications on modern software infrastructure and be in the position to centrally manage platform upgrades and security fixes.

“Legacy apps
impede cost
efficiency and
organizational
agility”

Enterprise IT decisions are as much about economics as they are about organizational capability. Legacy application code is an impediment to both cost efficient management of infrastructure and organizational ability to nimbly respond to changing business and technology requirements.

APPLICATION MAINTENANCE: Maintaining and extending legacy applications becomes more difficult over time as it becomes harder to find high quality development resources with the requisite experience.

INFRASTRUCTURE MANAGEMENT: Legacy apps may not be compatible with new desktop operating systems or server infrastructure, making it difficult to move away from non-supported platforms (e.g. Windows XP).

VDI ROLLOUT: VDI rollouts are often held back by infrastructure requirements of old custom applications.

APPLICATION LIFECYCLE MANAGEMENT: Updating desktop-based apps is more complicated than updating applications centralized on servers.

INTERFACES WITH OTHER ENTERPRISE APPS: As data sources change and new enterprise apps come on line, it is difficult to maintain data linkages with legacy applications.

The right approach is to redeploy applications using a server-centric architecture that can be managed and upgraded efficiently.

Supporting BYOD and Mobility

“Desktop apps
were obviously not
built with BYOD or
mobility in mind”

Enterprise mobility and BYOD are top of mind in the modern enterprise. Application strategy is evolving quickly - not only for B2C applications, but for B2E (business to employee) apps that are tightly intertwined with mission critical business processes.

Desktop client-server apps were not built with either of these technology trends in mind. Many legacy applications run only on Windows desktops with limited, if any, support for access from home computers or mobile devices. While application virtualization solutions address some of these challenges, they are expensive, complicated to maintain, and have performance limitations. Furthermore, interacting with virtualized applications on mobile devices is frustrating, with the user getting the same desktop UI on a smaller mobile screen.

HTML5 is emerging as the multi-platform, open standard for deploying business applications to web and mobile. The right approach is to redeploy legacy applications for delivery on cross-platform HTML5 that runs on desktop and mobile browsers.

What are your alternatives?

There are alternative approaches for managing the challenges of legacy desktop client-server apps. All have the goal of delivering centralized management and security, enabling better management of infrastructure and delivery via cross-platform web and mobile. However, all have limitations and frequently bring new challenges in managing cost, minimizing time-to-market, and reducing risk.



replace with out of the box

New out-of-the-box enterprise applications are an option if available – but often do not address needs of custom-written business apps.



application virtualization

App virtualization solutions are costly and complicated to implement. Performance overhead and lack of mobile-optimized UI negatively impacts user experience.



automated migration

Traditional automated migration technologies are ineffective at refactoring and re-architecting for web/cloud/mobile deployments – resulting in significant rewrite projects.



rewrite

Rewriting applications from scratch is time-intensive and costly. Can introduce new risks and does not make use of proven business logic of legacy app.

What is Gizmox Transposition?

Gizmox Transposition combines sophisticated source-to-source compilation technology with a highly efficient development environment that simplifies and accelerates the process of re-architecting and refactoring legacy code to modern .NET standards

01. Translation

Translator platform converts source code (e.g. VB6) to intermediate language without code freeze of source app

02. Compilation

Patent-pending virtual compiler processes translated code using out-of-box or custom-written mapping rules – highlighting areas for developer attention

05. Evolution

Leverage the full Gizmox platform to evolve, optimize, and extend application

04. Generation

Generate target code of choice (e.g. C#) with rapid compilation. Iteratively compile, test, amend code

03. Configuration

Developer audits automated output, makes edits to mapping rules as needed, and customizes code within Visual Studio if desired

KEY CHALLENGE:

Departing from
the client-server
paradigm

GizmoX Transposition was designed to automate the process of converting, re-factoring and re-architecting legacy code to the latest .NET and HTML5 standards. The transposed application is ready for deployment on modern web and cloud infrastructure. Developers can continue to manage and maintain using familiar languages like C# and VB.NET, and development tools like Visual Studio. Applications can be rapidly optimized for mobile form factors using GizmoX's Visual WebGui toolset.

RAPID DEVELOPMENT: Leverage automation and sophisticated IDE to rapidly convert legacy code to high quality .NET code

REARCHITECT, REFACTOR: More than just automated translation, Transposition supports developers in completely re-factoring and re-architecting code

LEVERAGES EXISTING SKILLS: Enables developers to re-architect existing legacy desktop apps to web and mobile using existing .NET and C# development skills

SECURE BY DESIGN: Thin HTML5 client and optimized client-server protocol insulates sensitive data and program logic deployed on centrally managed server

KEY CHALLENGE:

Managing security,
mobility, and BYOD
initiatives

Transposition helps IT organizations manage legacy app modernization under rational time-lines and budgets. The time frame, expense, and high risk of re-writing applications from scratch are often non-starters. Legacy applications are roadblocks to enterprise mobility initiatives and pose unique security challenges to IT organizations. Transposition can help companies accelerate the process of adopting modern application infrastructure.

MANAGE OBSOLESCENCE: Accelerate migration away from server and desktop platforms that are no longer supported (e.g. Windows XP) by removing legacy app dependencies

RATIONALIZE PLATFORMS: Consolidate infrastructure by standardizing on modern hardware and software platforms by removing legacy app dependencies

CENTRALIZE MANAGEMENT & SECURITY: Simplify application management by centralizing deployment on servers (with zero-install HTML5 web clients)

SUPPORT BYOD: Support enterprise mobility and BYOD initiatives by deploying applications with cross platform HTML5 web apps that run on any device, OS, and browser

KEY CHALLENGE:

Maintaining
operations while
complying with
IT standards
and leveraging
capabilities of
modern platforms

Transposition supports the migration of legacy applications to modern platforms while preserving the proven business logic and basic user interface model of the application. The business gets the benefits of operating on modern platforms (i.e. performance, scalability, security, mobility) while limiting the risk associated with migrating a mission critical application.

REDUCE TIME TO MARKET:

Leveraging existing business logic in legacy applications minimizes/eliminates need to write specifications for new development; Measure project time in months vs. the years it would take to re-write these legacy apps from scratch

REDUCE COSTS:

Project costs are dramatically lower than alternatives; Ongoing maintenance far more efficient; Avoid retraining staff; Redirect resources to revenue-generating initiatives

REDUCE RISK: Transposed applications leverage proven business logic and data dependencies of the original legacy application

IMPROVE CUSTOMER EXPERIENCE:

Extend reach of application to mobile and rapidly adapt UI to take full advantage of mobility; Deliver native-quality user experience