

Announcing: Visual LANSA Framework for .NET

Framework Based Development

Frameworks provide developers with the foundation to enable a quick start to building an application. The framework takes care of the generic functionality and underlying technology, freeing the developer to concentrate on delivering the business components. Without a framework, the developer will spend much of the development effort on constructing the foundations and plumbing for the application. While this approach may suit the developer's craving to write code, it does not serve the business or its shareholders well.

In contrast, the Visual LANSA Framework provides prototyping tools that help the developer assemble the application with user participation and approval all the way through the development cycle. The developer will expand the application gradually and does not need to throw away the prototype. LANSA's holistic approach to prototyping, designing and implementing an application ensures that all of the development effort goes towards delivering a business outcome.

The Visual LANSA Framework for .NET (VLF.NET) extends framework capability so that a developer can code once and deploy the application in a Windows rich-client context, as a Web application or both. The deployed application is a compiled C# executable that runs fast (not interpreted) and is safe (.NET CLR compliant with digital signature).

What is VLF.NET?

LANSA has designed VLF.NET to facilitate the development of commercial applications. VLF.NET offers a snap-in component architecture that facilitates agile development. The framework caters for business objects and includes segments to handle filters and searches, search result sets and handlers to allow users to work with instances of business objects.

As an example, with a 'customer' object, the filters and searches provide the means to choose subsets of customers by name, location or any of the attributes associated with the customer object. The result set returns a list of individual customers that match the criteria. The handlers manage tasks like editing a customer, showing current orders for the customer and a map showing the customer's location.

The snap-in architecture allows developers to build discrete handlers as components and to snap these components into the framework when the component is ready for deployment. Multiple developers can work on components simultaneously to reduce the elapsed development time. VLF.NET comes with ready-to-use components that allow developers to associate events with business objects, as well as components to include Google maps and other mashup-style applications.

On deployment, VLF.NET runs as a compiled C# executable that is digitally

signed. The compiled executable runs faster than an equivalent HTML and JavaScript application that has to be interpreted. The application deploys on demand with near zero deployment overhead. The user interface operates in an AJAX style and interaction with the server occurs in the background. You can rearrange the layout and choose from Themes to control the color scheme.

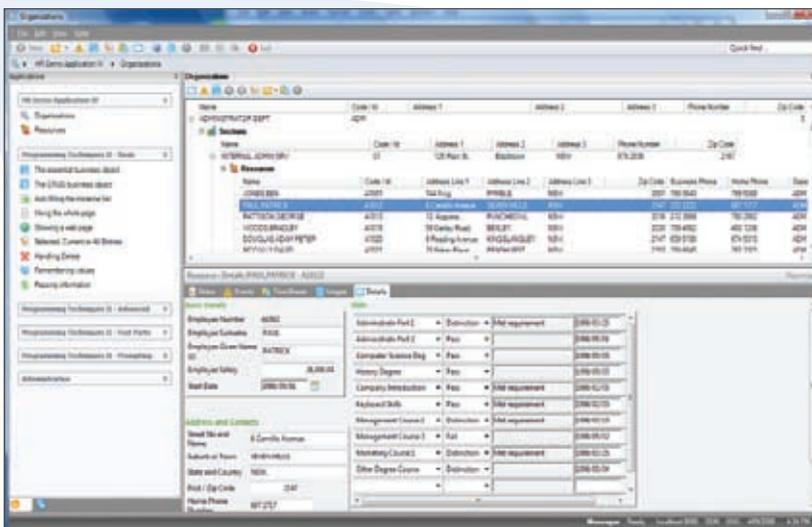
Inject New Life into IBM i Applications

In addition to being used for new development, VLF.NET can also be used in combination with RAMP, LANSA's modernization tool for 5250 applications. Developers can snap a handler for the 5250 application into the framework and RAMP will manage the user interaction with the 5250 application, presenting the green-screen as a graphical interface. Within the same framework, the 5250 application will now work side-by-side with applications built for a rich-client or Web user interface.

VLF.NET also caters for developers who need to work with the IBM i by providing snap-in components to manage spool files and link documents with business objects, (e.g. to link a digital image of an invoice with a customer and store them in the IBM i database).

Benefits of VLF.NET

- Component architecture supports snap-in capability for building composite applications (mashups)
- Supplied with readymade components for common tasks
- VLF.NET generates C# code that produces .NET CLR executable applications
- Compiled C# applications run faster than interpreted JavaScript
- Enhanced security – the digital signature ensures the safety of the application
- Near zero deployment overhead for both the Windows rich-client and Web applications
- Support for multiple application servers including IBM i
- Themes to enhance the visual aspects of the applications
- Users may rearrange the layout of the components in the user interface



Visual LANSA Framework delivers an authentic rich client experience to Web applications, built on C# and the .NET Framework.