

# Business Applications: Building a User Interface to Deliver Optimal User Experience

By Ofer Spiegel,  
Director of Product Marketing

August 2009



## Business Application Richness

### User Interface vs. User Experience

With the maturity of the Internet, enterprises now have a wealth of opportunities when it comes to enhancing their IT offerings and attracting new customers.

When designing a business application, enterprises already realize the need to avoid the types of user experience drawbacks typical of browser-based designs. They also realize that end-users are still very much accustomed to the fully managed, flowing and familiar experience of desktop applications.

Keeping a desktop-like performance is important, then, in avoiding the sort of frustrations that the browser can wreck upon a user's experience. Desktop-like user experience also allows for greater application adaptation - a factor that contributes to the end-users' overall business productivity and satisfaction.

### Which is Rich?

It is important to distinguish between a **rich user interface** and **rich user experience**, especially when it comes to business applications that are more of a B2B nature.

**A rich user interface caters more to the way the application looks, while rich user experience relates more to the way the application is used.**

A rich user interface may offer a highly aesthetic design utilizing the latest graphics and may include elaborate animations, newly shaped controls, embedded media, and so on. However, a

highly designed user interface does not automatically translate into optimal user experience and in many cases, may even hinder it.

Rich user experience caters first and foremost to the ability of end-users to complete tasks in the most efficient and productive manner.

With this in mind then, rich user experience conveys a highly intuitive functionality, with a minimal effort needed to understand the user interface, optimal performance and end-user productivity and ultimately, greater end-user satisfaction.

While aesthetics plays an important role in the overall satisfaction of the end-user, **it must always be considered a means to an end rather than an end in itself.**

### Beyond a Rich User Interface

Rich user experience requires not only that the user interface be intuitive, fluent and productive, **but also that it be functionally rich.**

**A true rich internet application (RIA)** is therefore one that is **able to complete most common business rules without having to constantly resort to remote services to complete the execution.**

Functionally rich RIA clients increase the overall responsiveness of the business application which in turn contributes to higher end-user productivity.

### Local Device Integration

A significant part of an internet application's rich functionality relates to its ability to interact with the local machine on which the RIA client is executed. Many RIA technologies suggest nifty user interfaces but in

reality lack the ability to smoothly integrate with local resources and devices that reside on the client side.

Local device integration ranges from simple access to the local



file system, through local applications, to peripheral devices such as printers, scanners, cameras, bar-code and RFID readers.

### RIA Developers Productivity

With unlimited resources, almost any development team could get round the technological limitations of the tools they are using and tame them sufficiently to produce a business-optimized RIA.

**In the real world however – and particularly in today’s recession climate, RIA projects lack the privilege of endless resources and flexible timeframes. RIA projects are expected to be completed under increasingly tight schedules and with leaner IT budgets.**

### Introducing uniPaaS RIA

Magic Software's uniPaaS application platform is designed from the outset to use a more productive approach to

developing and deploying applications.

These types of applications



inherit **all the benefits of web enablement, but without compromising on the rich user experience of the desktop that’s essential for running complex business procedures.**

### Single Development Paradigm

uniPaaS provides all aspects of the client, server and inter-lying communication layers within a single solution. **uniPaaS automatically translates programming into client side and server side instructions and automatically partitions the application flow.**

### Metadata Driven Platform

The uniPaaS development environment is really a visual representation of application ‘assets’ and business rules that’s stored in XML based ‘metadata’. **This metadata business structure can be easily transferred from one deployment scenario to another** (for example; Full Client to SaaS) with minimal effort.

### .NET Based RIA Client

The application’s metadata is unfolded in deployment using a generic, automatically installed .NET based client module. The .NET RIA client provides the end-user with a genuine desktop experience. The difference is that the RIA is also transparently connected to the central remote application server (via internet). **This means it can obtain live and automatic updates and it can keep the application fully serviced with server side procedures.**

### Extendable Client User Experience

The uniPaaS RIA client is a pre-compiled .NET client module that reflects the application’s definitions described in the metadata. Although the client is a pre-compiled .NET module, the developer can easily enhance the client's user interface by extending the client module to utilize any external .NET resource.

## Extending uniPaaS .NET Client

uniPaaS offers built-in graphical user interface elements that allow developers to easily produce desktop-like Rich Internet Applications featuring a familiar and productive end-user experience.

To produce an even more compelling user experience, uniPaaS allows developers to incorporate built-in .NET framework resources, or any third party .NET resources.

These enhancements can be achieved in uniPaaS with the following features:

### .NET Controls Embedding

uniPaaS developers can enhance the look, feel and functionality of any RIA by directly embedding new .NET controls on the form and integrating the controls with the overall functionality of the application.

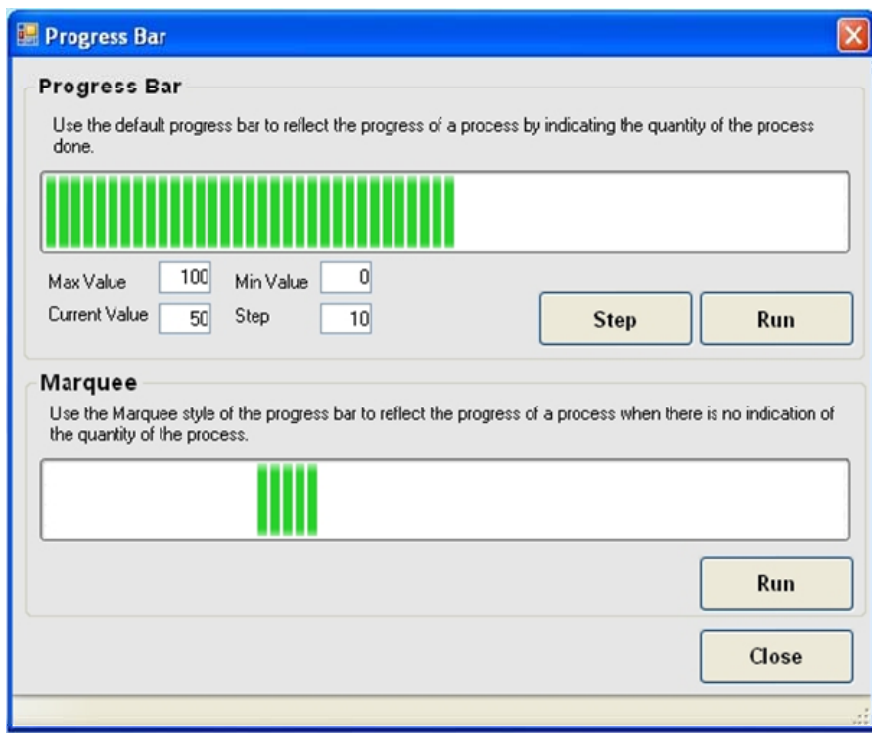


Figure 1: Embedded .NET progress bar within a uniPaaS RIA form

### Assigning Business Rules

Once the .NET control object is defined in the *data-view* layer of the task, the .NET control can be dropped onto the form. The control's property sheet can be directly viewed and manipulated from the uniPaaS *Form Editor*. From the control's property sheet the developer can set any value to every property and assign the properties with data-bound business rules. Such business rules will automatically set the property value whenever the data to which the business rule is bound changes its value.

### Live RIA Demo

You can experience uniPaaS RIA technology now at [riademo.magicsoftware.com](http://riademo.magicsoftware.com)

The ability to have properties automatically assigned with new values whenever live data is modified reduces the amount of explicit instructions needed and increases the developer's productivity.

Control properties can also be set explicitly at any logic segment and event handler.

### Handling Events

uniPaaS developers can handle .NET control events by creating uniPaaS **Event Handlers** that are invoked whenever the specified event is triggered by the .NET control. Upon selecting a control's event for handling, the required event parameters are created automatically in the uniPaaS handler.

### Invoking Control's Methods

Every control can be directly manipulated by invoking its methods. The uniPaaS *Business Rules Editor* provides a full description for every method's syntax.

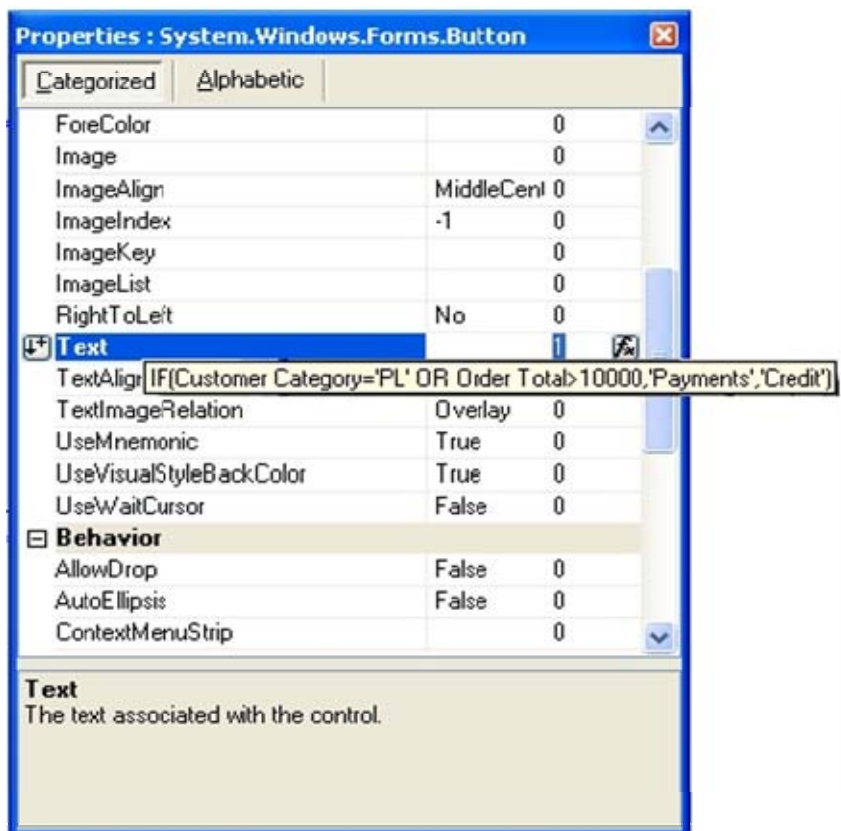


Figure 2: A business rule defines the dynamic value of a .NET control's text property

### Syntax Auto-Completion

The uniPaaS *Task Editor* provides auto-completion for any .NET related syntax. This includes the declaration of a .NET class, referencing control's properties and invoking the control's methods.

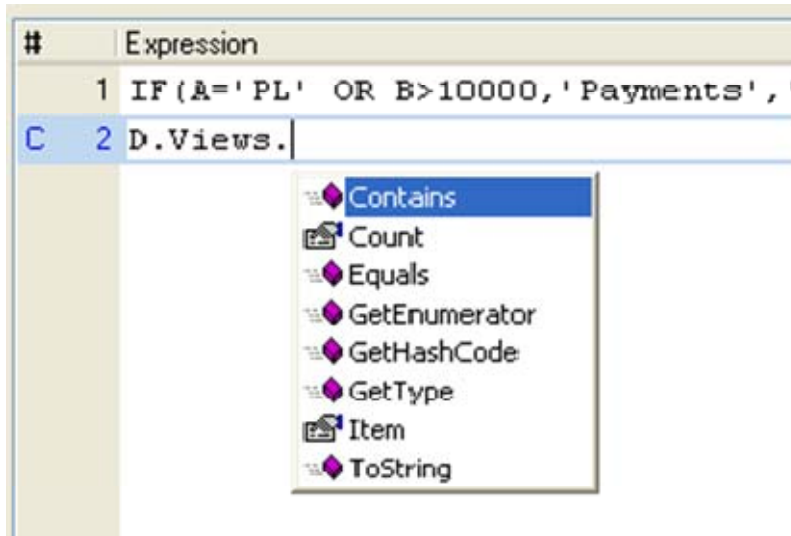


Figure 3: Auto-completing of .NET instruction syntax

### Automatic Type Conversion

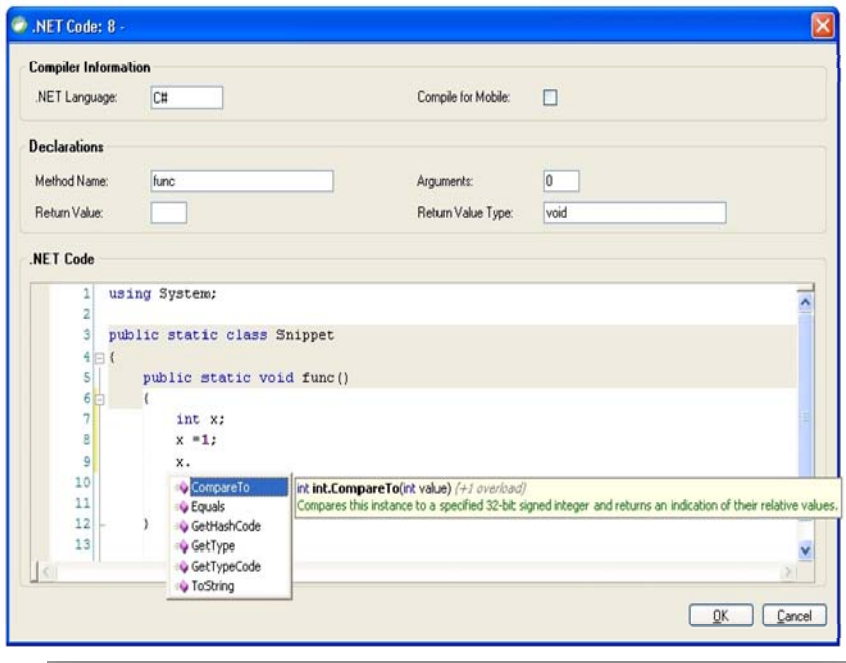
uniPaaS data types are simple to use and understand. When passing information from uniPaaS to the .NET object, uniPaaS automatically translates the data into the expected structure of the .NET object property or method parameter. This capability relieves the uniPaaS developer from unnecessary type casting.

### .NET Assembly Utilization

uniPaaS developers can add to the user experience and functionality of their RIA not just by embedding .NET controls but also by integrating .NET assemblies that can provide services of any sort and even to combine full .NET forms. This is particularly useful for organizations that already have pre-built .NET resources and who want to combine these resources into their uniPaaS RIA project.

### Direct .NET Coding

uniPaaS enables developers to access all elements of a .NET object using *Operations and Expressions*. Extensive .NET coding can also be embedded in a RIA program by explicitly placing pure C# or VB.NET code as a *Code Snippet*. The uniPaaS platform compiles the code snippet at runtime and then executes.



**Figure 4: uniPaaS built-in C# and VB.NET script editor**

The direct .NET coding enables even smoother integration with the .NET objects and eliminates any limitation in embedding .NET procedures.

### Local Device Accessibility

uniPaaS offers built-in functionality for accessing local device resources that reside on the client side. The .NET extensibility provides more functionality options for RIA developers, connecting local devices such as cameras, scanners, barcode and RFID readers, and printers.

### Extendable Mobile Client

The uniPaaS .NET RIA client is also compatible with Windows Mobile based handheld devices. The .NET based client extensibility is also available for mobile clients. uniPaaS RIA projects can therefore be designed to offer sophisticated and compelling rich internet applications for mobile devices as well as for PCs.

## Conclusion – Carefully Select your Platform Tools

When designing and developing a rich internet business application, optimal user experience is achieved mainly by giving users a familiar and intuitive interface. The better the user experience, the more productivity the business application will generate for its owners.

Top notch graphical design may contribute to overall user experience, but in today's economic environment of leaner IT budgets, too much focus on design can end up impeding productivity.

Enterprises and software vendors looking to make the greatest return on their business application investment should therefore consider platforms that cater first and foremost to user experience - while keeping the door open to extra graphical and functional enhancements that can be added when the time is right.

## About Magic Software

Magic Software Enterprises (NASDAQ: MGIC) is a global provider of application platforms and business and process integration solutions. With over 25 years of experience, our technology gives our partners and customers the power to leverage existing IT resources, enhance business agility, and focus on core business priorities.

Magic Software has thousands of successful customer installations worldwide and a global network of ISVs, system integrators, value added distributors and resellers, and consulting and OEM partners. Our technological approach, product roadmap and corporate strategy are recognized by the leading industry analysts. Magic Software has offices in 10 countries, a presence in over 50 countries, and partner alliances with global IT leaders including SAP AG, Salesforce.com, IBM and Oracle.

For more information about Magic Software and the uniPaaS application platform visit [www.magicsoftware.com](http://www.magicsoftware.com)

For more about our industry related news, business issues and trends, read the [Magic Software Blog](#).

For the latest news about uniPaaS and application platforms from Magic Software's Director of Product Marketing, read Ofer Spiegel's [Usability, uniPaaS, uNameIt Blog](#).