Domino Designer's New XPages

The New Design Element You Didn't Know You Need

Published by special permission from IBM Corporation on January 28, 2008

Email: *andrewp@thenorth.com*
Web: *http://www.thenorth.com*
Blog: *http://www.thenorth.com/apblog*

NORTHERN Collaborative TECHNOLOGIES

Domino Designer's New XPages

The New Design Element You Didn't Know You Need

New functionality in the IBM Lotus Domino Designer is always a huge draw at Lotusphere, and this year will be no exception. Much of what has been announced already -- and much of what will be unveiled in January -- is sure to bring rousing applause on first site. Some things, however, take a bit of understanding before their real power can be fully understood. One such new feature set is called "XPages" and is going to quickly become one of the most important design elements you didn't know you needed. XPages, by letting you use a single design element to work with data on multiple documents at once, solves one of the biggest user interface challenges developers face. At the same time, they break down a huge barrier of understanding between Lotus Notes developers and those more comfortable working with traditional database tables and joins.

A Work in Progress

I was fortunate enough to be exposed to XPages through design partner programs, and with special permission from IBM, I've been working with the development team responsible to create this article. We're going to take an early look at the feature so that when you see it at Lotusphere you can help explain it to everyone around you. As with all pre-release features, we need to focus on the idea and purpose. This article will outline the benefits and possible use cases for XPages as they are currently planned. Specific details of implementation, release dates, and other aspects of the final release are still very much works in progress.

One of the Oldest Challenges in Notes Development

One of the biggest strengths of IBM Lotus Notes has always been its flexible, "semi-structured" data container model. Each Database is a container full of notes and each note is a container full of items. The items stored on a note are uniquely defined to that note, and do not need to match a pre-written schema or table structure. There are design notes – like forms and views, and there are data notes – like documents.

In traditional Lotus Notes development, if you want to change data in the user interface, you do it by opening a data note (a document) in an editor through the stencil of a form. You can bring different sub-forms together or even change the form used to edit a document – but you can't use a single form to edit data on more than one document at once – not without resorting to some extensive workarounds. For example, you can show items from other notes using things like "@dbLookup" or "@getDocField" and in some cases you can work around the limitation by using "@setDocField" or custom scripts when the document is saved. ***XPages let you use a single editor to work with data on multiple documents at once.*** *This solves one of the oldest challenges in Notes Development.*

XPages aren't the first attempt to solve the limitation of having a single document note open in an editor at once. In the past, features like embedded editors, in-view editing, and extensive post-save events have been commonly used to create applications that manage data on related documents in near-real time. None of these has actually allowed the data to be open for edit in the same user interface element at the same time, however.

It's "Bound" to be More Manageable

The magic of XPages is in the binding. Like other development environments, you design your XPages based on what it will look like to the user. You put fields on it where you want users to interact with the data and generally think about the user rather than the data. Unlike traditional Notes Designer forms, however, the fields you place on XPages are not tied to any specific document. Each *element on the XPage is "Bound" to a data source. That binding could be to a Notes Document, or in the future could be to other kinds of external data. This means that on a single editable XPage you create for managing customers, you can display and edit company data that is stored on a company document, as well as customer specific data which is stored on a customer record. Instead of fancy scripting techniques executing in post-save events, the actual data is updated as part of the save process. Data isn't copied and propagated, it stays where it belongs.*

Wash, Rinse, Repeat

For new applications, using XPages instead of forms will be second nature almost immediately. They are likely to change the way applications are built and data is stored in Lotus Notes databases on a wide scale more fully and more completely than any change since the addition of Lotuscript; but what about views? To begin with, XPages come with a "View Control" designed to be modern and pretty, and which can be stylized and managed to quickly bind data from a specific view. That's the first way to handle view data but not the only one.

On XPages, you'll be able to lay out design elements that are bound not only to Lotus Notes documents, but also to the set of documents contained in a view. By marking an element bound to a view as "repeated", you will be able to display all of the documents in that view through those layout elements. Repeated elements will work as if you had created a copy of a set of elements on the XPage, and pasted a copy of that set of elements onto the form for each entry in a view. You won't have to do that, however. Repeated elements do it for you – automatically adjusting to the number of entries currently available or selected.

A "repeated" element is much more than a simple grid layout tool. You'll define groups of elements that work together and are bound to each entry in a view. These elements don't necessarily need to be in a single horizontal row. The binding is valid to one document for each instance of a repeated set of elements.

An Eventful Release

Data binding to multiple separate data sources is continued into the powerful scripting model supported by XPages. Each control has a full range of event triggers. Controls bound to data sources even support events based on changes to back end data wherever applicable. These events, as well as traditional ones like mouse-over or change of focus, can be used to trigger built-in functionality like content updates and recalculation or to trigger your own code on both the client and server side. There are even provisions for client side events to trigger server side agent code.

Like all aspects of XPages design work, as much functionality is built in and accessible with just a few option settings as possible. For example, events can be tied to both full and partial page refreshes. In the case of a partial refresh, the specific controls you want refreshed can be selected at design time or even based on run time calculation. Wherever possible, AJAX functionality is used for partial page updates. You might create a data element which is a keyword selection bound to a view lookup. By enabling 'type-ahead', your page could gain the smarts to use AJAX for interaction with the source data.

Think Globally, Act Locally

Localization is a big promise we've been hearing about with different design tools for a long time. XPages brings important advances to the challenge of using the same design elements in multiple languages and cultures. An easy click away, XPages localization will automatically generate the resource files to support localization for virtually all the design elements on the page. The resource files are pre-populated with the current language implementation being used by the designer, so that they can be given to a translator for line by line changes and easily re-integrated to the overall design.

XPages are for Notes and Domino Developers – Past, Present, and Future

New kinds of applications and much better designs for traditional ones are sure to be a result of adding XPages almost immediately. In addition to the new kinds of things we'll be able to do as developers; however, are the new developers who will find XPages much easier to understand from a design perspective than traditional forms and views. Bridging the divide between Notes database development and relational database design goes a long way to increasing the competitiveness of the platform as a whole.

XPages are being planned for shipment in a future release of the IBM Lotus Domino Designer. They are to be part of the core product functionality and won't require additional products. While integration with other data sources is clearly part of the plan, XPages are about Notes and Domino developers.

Open Questions, More News, and Details at Lotusphere 2008

With this article, I hope I've explained some of the benefits and use cases for XPages. Many questions remain to be answered as the design process continues. I hope you'll come see them at Lotusphere 2008, where you'll be able to talk with developers about the implementation details, layout tools, planned data binding sources, and maybe even targeted release dates.

About the Author

Andrew Pollack has been working with the Lotus Notes platform since 1991, and is considered a leading expert in application development, administration tasks, and security related to that platform.

Andrew has been a speaker and featured expert at more than a dozen industry conferences, including Lotusphere.

In addition to his consulting role, Andrew is a volunteer firefighter in Cumberland, Maine. He is the Lieutenant on Engine 1, and is also a member of the "Technical Rescue" team with training in Confined Space rescue, Hazmat, Extrication, Ice Water Rescue, and R.I.T. - which is responsible for the rescue of trapped, lost, or injured firefighters during the course of operations on scene.

To Contact Andrew, use the following:

Email: andrewp@thenorth.com
Web: http://www.thenorth.com
Blog: http://www.thenorth.com/apblog