

Maximising the Benefits of Lotus Domino 8.5.x with XPages

1 What is XPages?	3
2.0 Risk-Benefit Analysis	4
2.1 Software Impacts.....	4
2.2 Delivery Options	5
2.2 Learning Requirements	6
2.3 Technical Analysis of XPages Functionality	7
3.0 Summary.....	9

1 What is XPages?

With Lotus Domino 8.5.0 IBM introduced a new development extension to the Domino environment called XPages. XPages allows developers to more easily surface Domino data to browsers, the Lotus Notes client and mobile devices with a Web 2.0 look and feel. The integration of the Dojo framework and AJAX means a richer, slicker user experience with less coding for developers.

XPages was originally developed for Lotus Workplace Designer and Lotus Component Designer. It was introduced as a development tool for Domino Developers at Lotusphere 2009 and available with Notes & Domino 8.5.0. 8.5.1 was released in September 2009 which included significant enhancements, including allowing XPages applications to be opened in the Notes Client. A managed beta program for 8.5.2 began shortly thereafter and 8.5.2 is due for release in Q3 2010*. There are already signs of significant enhancements in that version, both in terms of functionality and productivity.

XPages added several new design elements into the nsf of Domino Designer:

- XPages
- Custom Controls
- Server Side Javascript Libraries
- Themes

XPages adds an XPages tab and additional options to the Launch tab in the Application Properties in Domino Designer. Traditional resources such as Javascript Libraries, Stylesheets, Image Resources and File Resources are utilised within XPages code. Forms and views are used as data sources for XPages design elements.

XPages will be a significant technology for the future of Lotus Domino, as evidenced by the heavy investment from IBM into XPages and its role in the display of content for Project Vulcan, the vision expounded at Lotusphere 2010 as core to the direction of Lotus technologies over the next five years.

* At the time of publication Lotus Notes & Domino 8.5.2 was in managed beta code drop 4. Any features discussed for Lotus Notes & Domino 8.5.2 are based on that code drop and may or may not be included in the final shipped product.

2.0 Risk-Benefit Analysis

2.1 Software Impacts

Software impacts will always be a deciding factor when IT departments consider taking advantage of functionality in newer releases of Lotus Notes & Domino. The standard sunset period for IBM Lotus Domino software support is two years after the go-live date for the version two releases subsequent. This meant that as of April 2010 IBM ceased support and maintenance for Lotus Domino 6.x. Consequently it is important to ensure timely version-to-version upgrades. This is an important reason why some customers are choosing to migrate to Domino 8.5.x at this time, especially when combined with the benefits of DAOS and XPages on a Domino server and XPages, composite applications and the plugin capabilities on the Eclipse-based Lotus Notes R8 client.

Pros

- Supported version
- Additional software enhancements
- Do not need to replace existing servers
- Enhancements from 8.5.0 / 8.0
- Variety of environment options

Cons

- “The collection has become invalid” error
- Potential hardware impacts for clients

Understandably some customers are reluctant to upgrade to the most current version because of the risks of bugs or instabilities. The Lotus Notes 8.5.1 client and Domino 8.5.1 server are significantly more stable than previous R8.x versions and performance impacts of the Eclipse platform for the Lotus Notes and Domino Designer clients have been significantly improved. However, a bug has been identified affecting all Notes versions in the R8.5.1 timeframe (8.5.1, 8.0.2 and 7.0.4). This arose from changes to the view indexing process and manifests itself in intermittent errors with the GetDocumentByKey, GetEntryByKey, GetAllDocumentsByKey, GetAllEntriesByKey methods of the NotesView object. These intermittent errors are “Error 4000: %a’s certification log” (in R7.0.4) or “Error 4678: The collection has become invalid” (in R8.x) errors, and can only be resolved by amending LotusScript or Java code to set NotesView.AutoUpdate to False. Because of the complexity of the fix, no hotfix was available, but a fix is available in 8.5.1 FP3 (released May 2010) and 8.5.2 (due for release Q3 2010).

When considering rolling out Lotus Notes and Domino Designer clients, the Eclipse-based clients require more memory and greater processing power than the R7 clients. Without that additional processing power it is not possible to take advantage of the client functionality. This may require hardware changes.

But there is an alternative option if customers would like to take advantage of XPages for browser- or mobile-based applications. XPages code and design elements do not need to be in the same nsf or even on the same server as the data being viewed, created, updated or deleted. Providing the relevant servers are both trusted, a separate 8.5.x server can be used solely to utilise XPages functionality.

During 2010 IBM are expanding the options available for customers to deploy XPages applications. An onsite server can be used and some business partners provide hosting capabilities. In Q2 2010 IBM provided a Lotus Domino server image for use on the Amazon EC2 Web Service. At Lotusphere 2010 LotusLive Notes was announced as a SaaS offering, either solely as a cloud or hybrid (on-premises and cloud) environment.

2.2 Delivery Options

Developers who have had to develop an application for the web will understand the challenges of cross-browser compatibility. Certain HTML and javascript code is not implemented consistently across different browsers or across browser versions. XPages functionality will work out-of-the-box on all major browsers, including Internet Explorer and Firefox, meaning developers will have less recourse to identify and search for workarounds.

In addition to browsers, XPages functionality also works on iPhone browsers without modification. Yet it is possible to identify and manage accordingly depending on whether the device being used to access the application is a normal browser, an iPhone or another mobile device. For delivery on Blackberry devices, TeamStudio have recently released a product TeamStudio Unplugged, which allows developers to deliver an application as a native Blackberry application leveraging XPages skills and without needing to learn an additional development language. The product is in the process of being extended to deliver native applications to the iPhone and iPad integration will follow in due course. Android integration may follow, depending upon demand. The licensing model means that the development kit is free to download and includes a single user license. Additional licensing is on a per user basis. This means a proof of concept can be developed with no additional software costs.

Pros

- Fewer cross-browser issues
- Build once, deploy on multiple devices
- Include as rich client components
- Not necessarily rip and replace
- Can use separate nsf for different device types

Cons

- Some XPiNC issues
- Project managers / developers need to think more about appropriate UI.

From 8.5.1, XPages is available in the Notes Client. However, some functionality (e.g. certain relative links) does not work in the same way as in current browser versions and some functionality is not available (the javascript window.open or window.close functions), because of the method of implementation for the Notes client. Equally the XULRunner version used to render XPages in the Notes Client means that there are certain visual issues, although in the beta of 8.5.2 the XULRunner has been upgraded equivalent to the Firefox 3.5.3 codestream.

At any point where the Notes Client can display a browser page, it is possible for the Notes Client to display a fully-functional XPage. This means that in the Lotus Notes 6.x or 7.x Client, an XPage can be included within the Home Page; in the Lotus Notes 8.x Client an XPage can be included as a browser element in a composite application, as a Web Page widget in the sidebar, or as a Web Page widget linked to a Live Text action for appropriate functions; and in the Lotus Notes 8.5.1 Client an XPage can be used natively as a composite application element.

Because of this flexibility of delivery, applications do not necessarily need to be converted wholesale to XPages. Rather individual elements of functionality may be added using XPages, aimed at delivery solely for a particular medium and audience.

However, the flexibility to develop for browser or mobile devices is a flexibility that may require some management of user expectations and requirements. A good quality application will not use the same user interface for the Notes Client, browser, mobile device, and/or sidebar widget. The different screen size demands a different user interface to ensure it is fit to purpose. Different devices may be aimed at different target audiences or audiences requiring different functionality sets. Sidebar widgets and composite applications perform a targeted function and as such demand a specific functionality set and user interface. Developers, project managers and users should be aware that not all media will be appropriate for all applications. To ameliorate the user experience on a wide variety of devices there will be additional cost of development for the

customer. But this does not necessitate a cluttered design environment. Because XPages allows developers more easily to separate design and data, design for different devices can be in separate nsf files, making it easier to support and develop in parallel.

2.2 Learning Requirements

For developers with no web development experience, there is a steep learning curve, with topics of HTML, CSS and Javascript needing to be tackled. However, interaction with Domino objects (NotesDatabase, NotesView, NotesDocument etc) is through Server Side Javascript (SSJS), which leverages LotusScript or Java skills. Many @Formulas and standard Client Side Javascript functions have been made available within SSJS, making it a powerful new language with a high level of entry for existing Domino developers. All are converted to run as Java code by the XPages engine, meaning developers generate JavaServer Faces-based pages without needing to learn Java. Some traditional JSF objects (e.g. facesContext) are available for implementation. However, the XPages section of the Domino Designer Help is rather limited, meaning it can be difficult to work out how to implement certain XPages functionality.

The development IDE provides WYSIWYG “Design” and XML “Source” panes for creating XPages and Custom Controls, with an Outline pane for ease of moving design elements around and wizards to set attributes. Pre-built components are available, both within the Domino Designer client and from the community. This means new developers can create rich functionality without needing to write XML, HTML or CSS. As they become more confident, they can begin to work in the Source pane but still access the wizards as required, speeding up both development and learning.

Domino 8.5.0 provided the OneUI and webstandard themes, predefined frameworks for functionality and look and feel that is consistent with other Lotus products such as Quickr and standard IBM Domino templates like the Wiki and Discussion Template. 8.5.1 extended this with the OneUIv2 frameworks, five different coloured themes supported by an open-source template made available on OpenNTF. All these themes have web-based documentation to support implementation and allow developers to provide users with a familiar and slick user experience.

The Dojo toolkit is delivered as part of the Server and Client installs. This allows developers to take advantage of a widely used framework of code and widget samples to enhance the user experience. However, developers should be aware that the Dojo version differs between versions (Dojo 1.1 in R8.5.0, Dojo 1.3.2 in R8.5.1 and Dojo 1.4 in R8.5.2). As a result some code may not work on different Domino/Notes Client versions. The Dojo toolkit has a website of support materials, but this is not focussed solely or even primarily at Domino developers, and for some topics it can be limited. Because Dojo is still evolving, some functionality is experimental or limited in its application.

The risk of developing in a more recent Domino Developer client is minimised by hover text to show which version an attribute was introduced and, in certain cases, at which version they were phased out.

Pros

- Leverages existing skills
- JSF style delivery without requirement to learn Java
- WYSIWYG and source code panes
- Wizards to help set attributes
- Dojo toolkits installed
- Pre-built components
- Risk of using unsupported functionality mitigated
- Strong community resources

Cons

- Learning curve can be steep
- Dojo toolkits vary across releases
- Documentation limitations
- Requires awareness when selecting appropriate projects

Developers seeking to learn XPages have a variety of options. Online courses are available and classroom training has provided by business partners and IBM. There is currently no XPages Lotus Domino certification, but this may change in the future. After training has been completed, some business partner organisations such as Intec offer a mentoring service to support developers with challenges during their early projects. A strong community of developers and business partners also contribute to a comprehensive corpus of technical resources, whether in blogs, wikis or podcasts.

While developers are learning the flexibility and challenges of XPages, both developers and project managers should be especially conscious of the impacts on estimates for cost and delivery timescales. Projects should be selected appropriately to support developers through the learning curve and ensure both they and customers do not become disillusioned by difficulties or limitations in delivering advanced functionality. A particularly fastidious business owner can demand functionality beyond the current abilities of the developer and this can affect the perceived success of a project. An extremely functionally rich specification (e.g advanced validation, advanced dojo functionality) will require an equally steep learning curve for a novice XPages developer. This can significantly impact cost, developer confidence and success of the project. It is advisable for developers and project managers to spend time investigating the functionality sets of the XPages templates initially delivered with Domino 8.5.0 – the IBM wiki and discussion templates available from OpenNTF. This will give a clearer idea of entry-level functionality.

2.3 Technical Analysis of XPages Functionality

XPages code and functionality is still evolving at a significant rate, partly because of heavy investment and involvement in the community by IBM, partly because of a vibrant community pushing the boundaries of what is available and thus identifying bugs or requesting new features. This means it is more important than with traditional Domino to upgrade through point releases as soon as possible. Although there are risks of being at the bleeding edge, this does allow developers to take advantage of bug fixes and improved functionality, such as the CKEditor for Rich Text with 8.5.2.

It is true that there were a number of bugs in 8.5.0 with e.g. advanced validations, unexpected behaviour when content was in custom controls. Some issues are still being identified in 8.5.1. Some attributes are not obvious (e.g. Dojo extensions, some resources) and some areas of functionality are not yet provided, e.g. the startkey parameter for navigating views, direct integration with non-Domino platforms, public access documents, running code under an authority other than the current authenticated user, and integration with existing LotusScript or Java code. Some of this functionality will be provided, to a greater or lesser extent, by Domino 8.5.2. But the absence of seamless integration with existing LotusScript or Java code and the inability to reuse existing subforms are a major deterrent for many developers when considering migrating existing Client or web applications from traditional Domino to XPages.

Pros

- Performance
- Server Side Javascript
- Repeat Controls
- Dojo toolkits installed
- Pre-built components
- Wizards, drag 'n' drop of core controls, custom controls, SSJS speed up web development
- AJAX functionality with less coding
- Strong community resources
- Technology is constantly evolving

Cons

- Bugs may be encountered
- Some functionality not yet available
- No seamless integration of existing LotusScript / Java / web design elements
- No inbuilt debugger for SSJS
- Performance monitoring tools expected Q3 2010

Nonetheless, this should not detract from the considerable progress made even before the release of 8.5.0. No inconsiderable effort was involved in integrating appropriate @Formulas, Client Side Javascript functions, JavaServer Faces objects and Domino objects into the single cohesive, flexible Server Side Javascript language. On the whole it is easy to produce very powerful, scalable code from a high entry level point. The breadth of attributes that can be computed using Server Side Javascript enhance the power of XPages. And research has proven that Server Side Javascript runs approximately four times quicker than traditional LotusScript or Java code in Domino. This means complex code can perform very well in a production environment without additional optimisation work by the developer.

Utilising Server Side Javascript to control attributes effectively can maximise the reusability of custom controls, minimising the coding required. Core controls enable design to be built quickly. Wizards enable attributes to be set at the click of a button, refreshes of part of the web page to be triggered by merely selecting the area to be refreshed. The ease of reusing custom controls, enhanced by the Import/Export Plug-In for Domino Designer on OpenNTF, means it is possible to contribute to and take advantage of a wealth of shared code in the community to also speed up development. Repeat controls allow a single template to be developed that will automatically be reproduced to display a scalable collection of objects, e.g. a entries in a view, a NotesDocumentCollection, the outcome from an @DbLookup or @DbColumn, or a user defined array of content. Consequently, when developers have overcome the learning curve, XPages applications can provide complex functionality with quicker development times than traditional web development.

With repeat controls, content can easily be combined from multiple sources at run-time, with excellent performance, making it possible to quickly and easily join data from different views into a single display, even allowing a Domino database to simulate a relational database. The flexibility of XPages means developers have the opportunity to present data in ways that have previously never been possible in Domino. This will challenge project managers and developers to think outside the box, to imagine different concepts for presentation and layout, and possibly for data entities and relationships. But with vision, utilising the tools available can produce powerful functionality to allow users to work smarter.

The implementation of themes enables application-wide styling or resources to be implemented. This can be as granular as styling for specific controls, e.g. specific width for all rich text controls. XPages also supports localisation as well.

Although Server Side Javascript does not have any inbuilt debugging functionality, it is possible to integrate error logging tools such as OpenLogXPages to log to a Domino database based on the OpenLog open source template or Medusa, a plugin for Firebug to allow logging to the Firefox browser console.

Because XPages is still quite new and there is so much to learn, optimisation has not tended to be a focus. The performance of XPages means that is not a significant issue. But with 8.5.2 Medusa will also show performance of individual areas of an XPage. IBM has also released the XPages Toolbox to profile and monitor performance on XPages applications, which will work with 8.5.2.

3.0 Summary

XPages is not a fully mature development technology. The technology is still developing and evolving as developers explore its capabilities and push its boundaries. There are challenges, particularly when trying to convert existing applications and functionality on a one-to-one relationship from traditional Notes Client or Web to XPages. But the developer community has embraced the opportunity to contribute supporting materials – as templates, blog and wiki posts, podcasts, and code samples – to help one another maximise the power of XPages.

With as little as a single server running Domino 8.5.x and the free Domino Designer Client, XPages can be developed to display and update data in existing databases within a trusted network of existing Domino servers.

The variety of delivery mechanisms mean that with a single language and in theory a single interface, an application can be deployed using XPages to the Notes Client, any browser or on mobile devices. In practice this should be avoided, with an appropriate interface for the relevant device and audience. But the common language available means that it is easier than ever to build those interfaces to deploy to the Notes Client, different browsers, and a variety of mobile devices.

Moreover, XPages has been developed with considerable thought to leverage existing Domino language skills and ease the transition to developing XML design structures. Nonetheless, there is undoubtedly a definite learning curve, not only for developers but also for project managers, in terms of awareness of what is possible and the timescales required to achieve it.

XPages will not replace traditional Domino applications, certainly for a number of years. But it brings out-of-the-box Domino web development into the 21st century and will be a major aspect of IBM's vision for Lotus Notes & Domino over the coming years.