

# **Maximising the Benefits of Lotus Domino 8.5.3 with XPages**

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

## 1 What is XPages?

It is now three years since IBM introduced the world to XPages in Lotus Domino 8.5.0. The JSF-based technology, already some years in development before it was integrated into Lotus Domino, has now matured into a game-changing development platform. The integration of the Dojo framework, CKEditor and AJAX means a richer, slicker Web 2.0 user experience for browsers, the Lotus Notes client and mobile devices. Because it is based on JSF it provides a development platform not only for Domino developers but also for Java developers, with data models surfaced via JDBC from relational databases, held in Java Managed Beans or traditional Domino objects like Forms and Views. Business logic can be developed in Java or Server-Side Javascript Libraries. The Extension Library, surfaced either through the download from OpenNTF or Upgrade Pack 1, significantly increases the number of pre-packaged controls available to developers. This utilises the Extensibility Framework, enabling advanced developers to provide their own native controls. This flexibility allows a wider audience than ever before to take advantage of Domino's RAD capabilities, its powerful security model, its enterprise-level integration with Domino and non-Domino datasources, while also providing for high scalability and quality development processes.

XPages was originally developed for Lotus Workplace Designer and Lotus Component Designer. It was announced as a development tool for Domino Developers at Lotusphere 2009 and became 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. 8.5.2 was released in September 2010 providing a further tranche of significant improvements, most notably the sessionAsSigner/sessionAsSignerWithFullAccess Domino Objects, public access XPages, performance enhancements and the Extensibility Framework. Shortly after release of 8.5.2 IBM released the XPages Extension Library as open source on OpenNTF. This innovative approach by IBM to software provisioning demonstrates how developers can use the Extensibility Framework to contribute extensions for their own or community use while also improving developers' user experience without waiting for a gold release. Although official support channels were not available to address bugs or enhancement requests, IBM and partners demonstrated impressive turnaround of fixes where necessary. Enhancements in Lotus Domino 8.5.3, released in October 2011, focussed on the server architecture and development IDE. OSGi developments to make it easier to deploy the Extension Library and tooling enhancements to allow developers to give a better user experience with their extensions. These laid the groundwork for IBM to introduce a new deployment model in December 2011 - Upgrade Pack 1. Upgrade Pack 1 provided a fully supported version of the Extension Library, albeit not all controls then available in the OpenNTF release. This is a new deployment model for adding additional functionality on top of the base install, leaving fix packs to predominantly contain just fixes. Functionality from upgrade packs will then be rolled back into the core product in the next point release. The introduction of upgrade packs will continue through 2012 and into the future.

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

- XPages
- Custom Controls
- Server-Side Javascript Libraries
- Themes
- Java – a specific design element to provide Java classes for use in XPages.

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 can be used as the Model in this MVC architecture, if developers wish to use Domino as the data repository. When building applications 8.5.3 introduced a fully configurable XPages perspective to the IDE in addition to the Domino Designer perspective and an additional Forms/Views perspective for non-XPages application development.

## 2.0 Risk-Benefit Analysis

### 2.1 Hardware and 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 sun-set 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 2011 IBM ceased support and maintenance of Lotus Domino 7. Consequently it is important to ensure timely version-to-version upgrades. This is an important reason why 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.

Understandably some customers are reluctant to upgrade to the most current version because of the risks of bugs or instabilities.

Lotus Notes 8.5.3 client and Domino server have been available since October 2011 and with Upgrade Pack 1 provides over 100 officially supported controls and tags on top of the original 44 controls deployed with XPages. Domino 8.5.3 adds significant performance enhancements to XPages both on the server and client as well as upgrading the Dojo version to the 1.7 code stream, which means applications will work in Internet Explorer 9 without requiring developers to resort to compatibility mode. The pace of functionality release has been considerably more aggressive than has been traditionally experienced for Lotus Notes and Domino, making it more attractive for customers to upgrade even through point releases. However, some regression bugs are encountered as a result of the aggressive release strategy, so it is important to keep up-to-date with fix packs.

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. Even with up-to-date hardware though, XPages in the Notes Client takes longer to start up than for a browser. Notes.ini settings can be applied though to preload databases, both on client and server. Properties can be applied in the xsp.properties files both at server or application level to optimise performance. The XPages Portable Command Guide book succinctly covers these.

From the server viewpoint, 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 provision XPages.

In the middle of 2012 IBM announced the IBM XWork Server which is a cheap server install unlimited on processor power or users, allowing integration with data sources both Domino and non-Domino. The main restriction is the number and architecture of XPages applications on the server. In addition XPages applications can only be replicated with another IBM XWork Server. But this does provide a good entry point to XPages as well as a good option for software vendors.

#### Strengths

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

#### Possible Challenges

- Potential hardware impacts for clients
- Configuration settings for client and server need to be carefully considered

## 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 many mobile browsers without modification. The XPages Extension Library for Domino 8.5.3 and Upgrade Pack 1 provided mobile controls making it easier to develop an interface more targeted to the different screen size of mobile devices, automatically rendering in a style consistent with whichever mobile device is in use. Currently this covers iOS and Android, but Blackberry and specific controls for tablets will follow during 2012. Domino 8.5.3 also supports HTML5 which includes a wealth of functionality aimed at those developing for mobile devices. TeamStudio have released a product – TeamStudio Unplugged – which allows developers to deliver an application as a native Blackberry, iOS or Android application leveraging XPages skills and without needing to learn an additional development language.

### Strengths

- Fewer cross-browser issues
- Build once, deploy on multiple devices
- Not necessarily rip and replace
- Can use separate nsf for different device types
- Extension Library provisioning and official support improved

### Possible Challenges

- XPiNC performance issues before 8.5.3
- Can be viewed with no changes on mobile, but in most cases a specific interface is advisable
- Project managers / developers need to think more about appropriate UI

Social business has been a major focus of 2011 and with Domino 8.5.3 IBM extended the XPages Extension Library to deliver a host of social tools for integrating with public social networks like Facebook and Twitter as well as Lotus Connections. Whether customers have a fully-fledged Connections implementation or are taking advantage of the free entitlement in Lotus Domino 8.5.3 of Lotus Connections Profiles and Files, XPages can now integrate fully with those tools.

From 8.5.1, XPages was available in the Notes Client (XPiNC). However, some functionality (e.g. certain relative links) does not work in the same way as in current browser versions. For 8.5.1 some functionality was not available (the javascript window.open or window.close functions), because of the method of implementation for the Notes client. 8.5.2 improved some XPiNC performance and compatibility, but 8.5.3 is strongly recommended for better performance. From 8.5.3 XPages can also be used natively in widgets in the Notes Client. Plugin development can also integrate with XPages, allowing customers to take advantage of additional integration opportunities in the Notes Client not available in browsers.

Because of this flexibility of delivery, applications do not necessarily need to be converted wholesale to XPages, although this can be achieved either through development or third-party tools like GBS Transformer.

However, the flexibility to develop in XPages for browser or mobile devices is a flexibility that may require some management of user expectations and requirements. A good XPages application should free itself from the shackles of traditional Domino development in terms of navigation, data display and architecture. Developers, project managers and users should be aware that this could significantly alter user experiences for interacting with their applications. This may require research time to investigate and learn techniques best appropriate for XPages development. Training can provide some of the learning required to implement alternative navigation or data display techniques, but should not be expected to inform on potential options. This can only be gleaned by research into sample applications from sites such as OpenNTF, as a result of sessions delivered by subject matter experts at Lotus User Groups or as part of mentoring packages.

A well-considered 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 also 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 primarily through Server-Side Javascript (SSJS), which leverages LotusScript or Java skills. Many @Formulas and standard Client Side Javascript functions are leveraged 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. But experienced XPages developers acknowledge that Java should be embraced when the basics have been learned to get the most out of XPages. Some traditional JSF objects (e.g. facesContext) are available for implementation and some background knowledge of JSF is required to fully understand XPages.

### Strengths

- JSF style delivery without requirement to learn Java
- WYSIWYG and source code panes
- Pretty panels to help set attributes
- Dojo toolkits installed
- Pre-built components plus Extension Library
- Strong community resources and increasing number of books

### Possible Challenges

- Learning curve can be steep
- JSF and Java knowledge will be needed to optimise performance
- Dojo toolkits vary across releases
- Requires awareness when selecting appropriate projects

This makes for a lot of learning for Domino developers, but XPages allows this to be learned bit-by-bit. Documentation and learning resources are improving all the time, both within the product, online, and in books like Mastering XPages and XPages Portable Command Guide. During the rest of 2012 two more books are being published from IBM Press, including XPages Extension Library, authored by both IBMers and community leaders including Intec's Paul Withers. To support these resources some business partners also provide training and mentoring packages.

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. Domino 8.5.3 provided a specific XPages perspective, of particular benefit to developers from a non-Domino background because design elements not appropriate for XPages applications are hidden. 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, CSS or Java. As they become more confident, they can begin to work in the Source pane and use Java for business logic and data models, 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. Domino 8.5.1 extended this with the OneUIv2 frameworks, five different coloured themes supported by an open-source template made available on OpenNTF. Domino 8.5.3 extended this with more colours and provided OneUIv2.1, the theme already in use in the most recent Connections

release. 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 1.4.3 in R8.5.2, Dojo 1.5.1 (available in server files but not used by default) and Dojo 1.6.1 in R8.5.3). 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. But this does not preclude developers from incorporating other Javascript frameworks like jQuery or the OpenAjax Toolkit.

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. An XPages application can also be compiled for a specific Domino version, which identifies whenever a design element is saved whether there is any incompatible functionality used.

Developers seeking to learn XPages have a variety of options. Online courses are available and classroom training has provided by business partners and IBM. The courses available were significantly extended during 2011. Certification for Domino 8.5 was refreshed during 2011 to include XPages topics. 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 like The XCast and NotesIn9.

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, business logic or complex requirements) 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 delivered either within the product or on 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 and the Extension Library, some functionality for which (such as mobile or REST services) is only available from Domino 8.5.3.

It is true that there are a greater number of regression bugs found than traditionally encountered and that because loopholes that incorrect coding exploited also get closed with each release. This means that it is more and more important to test from version to version. But functionality enhancements with each version mean most XPages developers would heartily recommend undergoing this pain to take advantage of the new functionality.

However, the absence of seamless integration with existing LotusScript or Java code is still a major issue for many developers when considering migrating significant existing client or web applications from traditional Domino to XPages. There are third party tools available to migrate client applications but not all functionality may be appropriate and additional development may be required to improve out-of-date interfaces.

Nonetheless, this should not detract from the considerable investment by IBM in XPages even before Domino 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 considerably 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, Java or themes 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. 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 and even provide some functionality traditional web development could not.

### Strengths

- Performance
- Server-Side Javascript
- Repeat Controls
- Dojo toolkits installed
- Pre-built components
- Significant number of additional controls in Extension Library / Upgrade Pack 1
- 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

### Possible Challenges

- No seamless integration of existing LotusScript / Java / web design elements
- No inbuilt debugger for SSJS
- Best scalability, performance and error handling requires JSF / Java knowledge



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 functionality to be implemented. This can be as granular as styling for specific controls, e.g. specific attributes for all rich text controls or standard layouts or terminology for pagers. 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. However, the editor does not deliver the rich user-experience of the LotusScript or Java editors available in Domino Designer and Eclipse, which may encourage some developers to take the leap into programming primarily in Java rather than Server-Side Javascript.

Java packages or managed beans can be used in Expression Language or Server-Side Javascript to add advanced functionality and performance optimisation of an application. Domino 8.5.3 improved issues with refreshing Java code from templates to databases, but this is still a developing area. Domino 8.5.3 introduced a new Java design element but some developers have reported problems classes disappearing. Running Java code by multiple developers that uses sessionAsSigner can cause server issues. The enhancement in Domino 8.5.3 to provide better integration with source control will in all likelihood prove an invaluable resource for development teams and may encourage more developers to work on local servers and pushing code back to a central repository. With an appropriately-sized team it may be feasible for advanced developers to provide complex functionality via Java code or custom-built controls using the Extensibility Framework, while less advanced developers concentrate on developing the individual XPages.

With the latest releases of the XPages Extension Library it is easier to integrate with REST services, RDBMS datasources and social datasources using OAuth. Managed beans also provide easier access to user settings. Two new projects on OpenNTF, contributed by Philippe Riand from IBM, extend the reach of XPages still further. **Workflow for XPages** takes advantage of OSGi to allow developers to more easily build workflow into XPages applications. The workflow context is abstracted and accessed via standard APIs, allowing a simple workflow engine, Lotus Workflow, Lombardi Business Process Management or other workflow management tools. Approval chains are similarly abstracted, whether using XPages, Lotus Workflow, IBM Connections. **JDBC Access for IBM Lotus Domino** allows Domino developers to provide access to their data via SQL queries via an Domino tasklet using DOTS. This makes it easier to integrate non-Domino reporting tools. This further extends the reach of Domino into IT departments.

## 3.0 Summary

XPages is still developing and evolving as developers explore its capabilities and push its boundaries. But IBM and 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 IBM XWork server and the free Domino Designer Client, XPages can be developed to display data from a variety of datasources, whether the customer is already accustomed to Domino or new to the platform.

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 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. But the integration of the Extension Library, Java and the Extensibility Framework make it extremely powerful.

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 21<sup>st</sup> century and will be a major aspect of IBM's vision for Lotus Notes & Domino over the coming years.