intec

Maximising the Benefits of IBM Domino 9.0.1 with XPages

Contents

1 What is XPages

It is now five years since IBM introduced the world to XPages in IBM Lotus Domino 8.5.0. The JSF-based technology, already some years in development before it was integrated into Domino, has now matured into a game-changing development platform.

XPages was originally developed by IBM Business Partner Trilog. It was acquired by IBM and announced as a development tool for Domino Developers at Lotusphere 2009, becoming available with Notes & Domino 8.5.0. The integration of the Dojo framework, AJAX, the CK Editor, and Twitter Bootstrap via an OpenNTF project means a richer, slicker Web 2.0 user experience for browsers, the IBM Notes client and mobile devices.

Point releases contain not just fixes but considerable feature enhancements. Every release has provided an updated Dojo release. Notes and Domino 8.5.1 was released in September 2009 which included significant enhancements, including allowing XPages applications to be opened in the Notes Client.

Notes and Domino 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 Single Copy XPage Design. This allowed developers to use a single XPage template for multiple databases. Unlike normal Domino templates, this did not refresh the design down to dependent databases, but rather at runtime loaded the design from the template, thus improving performance significantly. Without doubt the biggest enhancement in Domino 8.5.2 was 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 IBM Lotus Domino 8.5.3, released in October 2011, focussed on the server architecture, source control enablement and development IDE, as well as performance enhancements. The Java design element was introduced. OSGi developments made it easier to deploy the Extension Library and tooling enhancements allowed 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.

IBM Domino 9.0 provided more tooling in Domino Designer, added the Jar design element, added enhancements to the source pane which is used by most XPages developers, brought the Upgrade Pack into the core product, provided further performance enhancements, XPiNC enhancements, an SSJS debugger, Social Business Toolkit for integration with Connections and Domino Access Services for accessing mail and calendaring via REST services outside Domino. CSS and JavaScript resources were also aggregated by default. IBM Domino 9.0.1 added still more enhancements with updated XPages templates, mobile development enhancements, Java calendaring APIs and enhancements to file upload capability.

The XPages Extension Library was a stellar leap forward for XPages in the delivery mechanism and the functionality. It further encouraged developers to take the leap into Java, moving Domino development from a proprietary technology full of hacks to a mainstream development platform that allows developers to leverage the wide gamut of open source libraries and frameworks with virtually no hacks in a way that is more accessible than ever to non-Domino developers. The development tool, Domino Designer, has also been enhanced to encourage more industry-standard approaches, with source control enablement. This is a more significant development than it seems, enabling developers to use the web development tools of their choice for CSS, JavaScript and Java files. With headless Designer available in 9.0.1 to build an NSF from source control files, it also provides the potential for continuous integration with a Domino database, another standard in software development.

IBM and the community has embraced the platform and standard approaches, with extensions to the core Domino Java API itself, a community-driven IBM-published book, source control management integration into OpenNTF, and StackOverflow as the community forum.

The enhancements from point release to point release mean it is more critical than ever before for developers and IT departments to leverage the latest releases. Without them, developers and end users will not get the best experience for their applications.

2 Recommendations and Discussion

2.1 Hardware and Software

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 better tooling, better out-of-the-box performance, greater functionality and more open source projects.

Anything earlier than IBM Lotus Domino 8.5.3 with the Extension Library will significantly affect the end user experience. Indeed the Dojo version in use in 8.5.2 – the most current at the time of release – is not supported by Internet Explorer 9. The only way to use an XPages application running on IBM Lotus Domino 8.5.2 is to run in compatibility mode, which impacts the user experience.

IBM Domino 9.0 or 9.0.1 is highly recommended. 9.0.1 Fix Pack 2 extends support to Internet Explorer 11. Performance of applications both on the server and the browser client are enhanced out-of-the-box. There are also enhancements to mobile browser development.

Every developer should be using Domino Designer 9.0.1 if at all possible. XPages elements of databases can be compiled against earlier Domino versions by selecting the relevant version in the Xsp Properties. This removes any risk of accidentally including unsupported code. The result will be a better performant, more stable development platform with more tooling available.

2.2 Developer Roadmap

One of the strengths of XPages is that it builds on a number of standard non-proprietary technologies, all implemented in standard ways: JSF, JavaScript, XML, HTML, CSS and Java mean that XPages is accessible to experienced Domino and non-Domino developers alike. It also makes a developer's skills more portable than ever before, both for Domino and any other web development. It also reinforces the skills required to integrate with IBM Connections or IBM SmartCloud.

Nonetheless, the learning curve for those new to XPages should not be under-estimated.

However, three books provide a wealth of information. "Mastering XPages", written by three of IBM's most experienced XPages developers, is now in its second edition and has added considerable content fully up-to-date for IBM Domino 9.0.1. It provides useful information for new and experienced XPages developers.

"XPages Extension Library", co-authored by IBM-ers and a number of non-IBM developers (including Intec's Paul Withers), provides detailed information on the significant number of additional controls added in 8.5.3. And "XPages Portable Command Guide" provides information for developers looking to optimise the performance of their applications as well as Domino administrators.

Community sites like XPages.info and NotesIn9.com provide considerable advice and support. StackOverflow has become the core forum for XPages questions. And there are a number of companies providing self-paced or instructor-led training, including Intec. Intec also has many years' experience providing XPages mentoring, which can help get in-depth support and explanation for problems encountered.

Furthermore XPages is structured in such a way that basic drag and drop, simple actions and Server-Side JavaScript – a language leveraging knowledge of Formula Language, the Domino Object Model from LotusScript and client-side JavaScript syntax – all make the transition easier and allow developers to build business applications and build their skills. This pseudo language is converted to run as Java code by the XPages engine, meaning developers generate Java Server Faces-based pages without needing to learn Java.

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 Designer has 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, most developers will work predominantly in the Source pane and use more advanced functionality like themes.

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.

OpenNTF has created a single project aimed at pulling together a variety of XPages-related best-of-breed projects into a single download and install, called OpenNTF Essentials. It includes projects like XPages OpenLog Logger, XPages Toolbox, XPages Debug Toolbar, OpenNTF Domino API, XPages Extension Library, Bootstrap4XPages and many more. Although released on a periodic basis, different versions of projects that comprise OpenNTF Essentials can be installed on top of it, allowing a customised setup if required.

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. But this does not preclude developers from incorporating other standard JavaScript frameworks like jQuery or the Twitter Bootstrap. A project on OpenNTF, Bootstrap4XPages – developed jointly by IBM and the community – provides easier access to Twitter Bootstrap. It has already made upgrading from Twitter Bootstrap easier

As greater experience is gained, a fuller understanding can be gained of the page lifecycle and performance optimisation. Then, for greater control, more efficient coding of back-end functionality, investigation of and extension of IBM code, developers are recommended to learn at least some Java. This allows business logic to be more fully separated from UI and allowing developers to implement an MVC model. It also allows Java code to run with a higher level of authority, which is often required to work around Java security exceptions.

See Appendix A for a recommended roadmap for XPages developers, showing learning points, next steps and resources to support learning.

2.3 Domino Administration and XPages

It is important to understand the architecture of XPages in the server. The XPages runtime runs on top of HTTP. Within the XPages runtime each database runs as its own JVM. Authentication, checking for valid databases and other resources, and checking for authority to run XPages is done by HTTP. The XPages runtime only launches if a user has access to the database, the relevant page is found in the database and the signer of all associated resources for the page has access to run XPages on the server.

There are a number of new configuration settings, log files and extensibility setup for XPages. The starting point when providing a server for XPages development is the server document. There is a new security option to define which IDs can sign XPages in order to run on the server. Any other ID used will throw an HTTP 403 error.

There are some additional XPages-related notes.ini settings. Other default settings can be modified server-wide by adding an xsp.properties file in <domino>/data/properties.

There is a sample file already in that folder with full explanations and with all settings initially commented out. This means the file can be copied and relevant settings easily set. IBM have published “XPages Portable Command Guide” which documents all configuration settings in full detail and is an invaluable resource for Domino administrators.

OSGi is used to dynamically load server-wide Java extensions to the XPages runtime. Many of these OSGi plugins or Extension Libraries come installed with the server, including the core Extension Library. But developers can also build their own OSGi plugins to provide server-wide functionality and avoid copying and pasting code from database to database. A number of OSGi plugins are also available from OpenNTF, including an RDBMS project for connecting to relational databases and IBM Social Business Toolkit for connection to Connections and other social platforms. OSGi plugins built by developers are usually added into Notes documents in an Update Site database, meaning nothing has to be installed physically in the server’s file structure. Consequently, OSGi plugins are a best practice method of extending the XPages runtime.

The Update Site database can also hold master copies of plugins to be circulated to developers or installed on Notes Clients for using XPages in Notes Client (XPiNC). A Widget Catalog and desktop policy can handle pushing them down onto relevant clients. Please note that users will be prompted to confirm installation of the plugins.

When monitoring the server console or log.nsf on the server, messages from the XPages runtime will be prefixed by HTTP JVM: CLFAD…., some of which are errors but some are just information messages. Errors will include a brief summary of the Java exception thrown but direct the developer to files in the IBM_TECHNICAL_SUPPORT folder. These files will be critical to the developer being able to identify the cause. However, two OpenNTF projects – XPages Log Reader and XPages Debug Toolbar – provide visibility of these files from a browser. Indeed XPages Log Reader may be of interest to admins because it gives access to all server-related logs as well as configuration settings.

Because XPages is running as Java within the XPages runtime, it leverages the Java security policy in the jvm/lib/security folder. This may be strange when the database ACL and the server document are so adept at managing security. Indeed the java.policy settings are so restrictive that they prevent a developer doing some things in Java that could be done by with full security by calling a LotusScript agent. But this is because the Java policy management is inherited from other IBM technologies, pre-dates XPages and has not been modified since XPages was released. If changes are needed, they can either be made directly in the java.policy or by adding a java.pol file in the same folder. However, double-check before and after upgrades because some fix packs and version changes have reverted the settings.

See Appendix B for an Administrator checklist for key tasks that may be required.

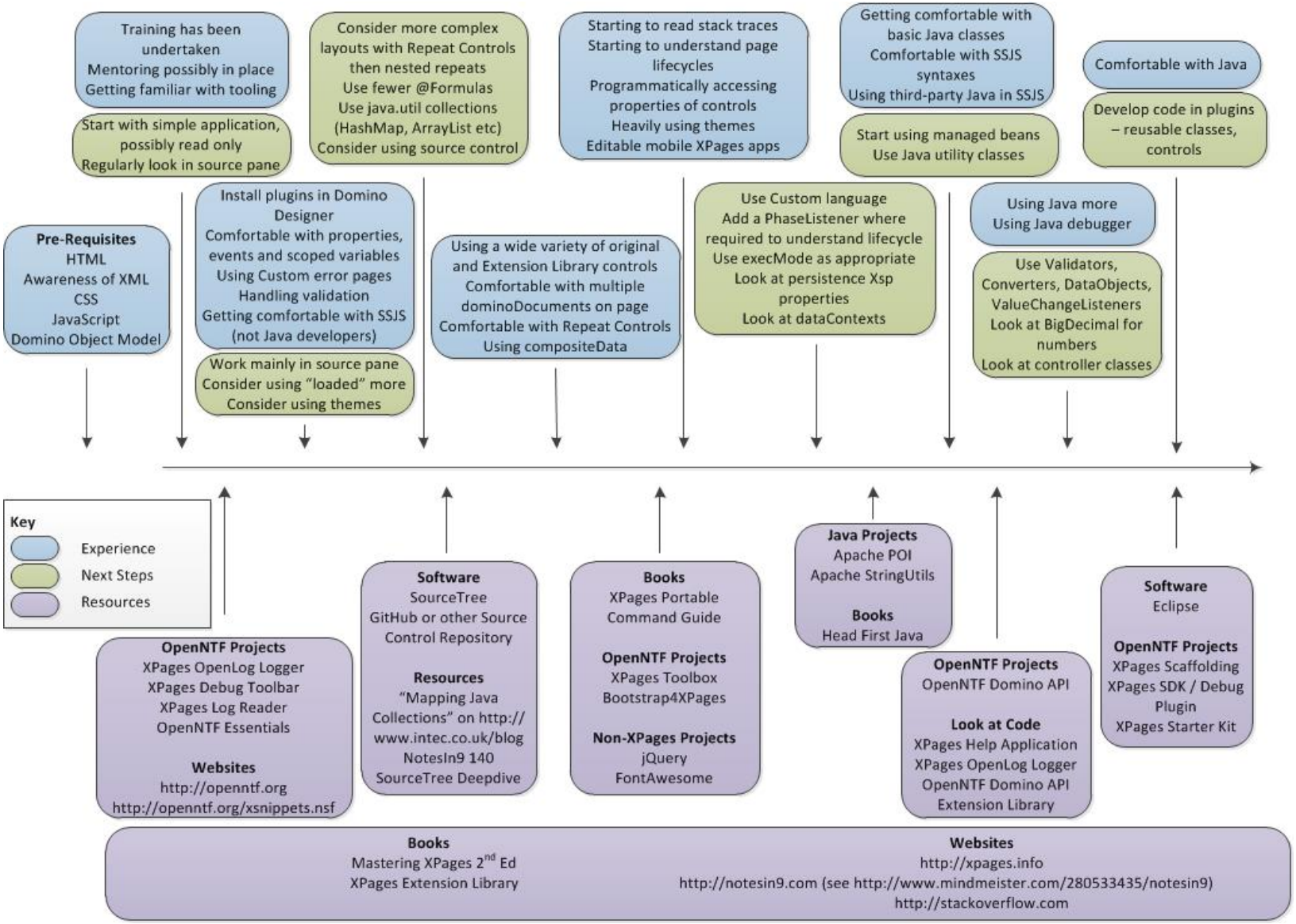
3 Summary

XPages is constantly 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 technology and a single IDE, an application can be deployed using XPages to the Notes Client, any browser or on mobile devices. In practice this should be avoided, with modifications 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.

4 Appendix A: Developer Roadmap



5 Appendix B Administrator Checklist

5.1 Server setup

- Ensure appropriate groups / user are in **Sign agents or XPages to run on behalf of the invoker** field in Server document under Security > Programmability Restrictions Who can –. Otherwise HTTP 403 errors will be thrown to a browser. XPages in Notes Client (XPiNC) uses the ECL and, in Notes 9.0 +, a new Java Code ECL setting.
- Create a database based on the **Eclipse Update Site** (9) template (updatesite.ntf) for XPages Extension Libraries, aka OSGi plugins. The usual filepath is updatesite.nsf.
- Add notes.ini variable to load plugins from the NSF: OSGI_HTTP_DYNAMIC_BUNDLES=updatesite.nsf. See "XPages Extension Library" book, Chapter 2, written by Domino Administrator and former IBM Champion Declan Lynch.
- Ensure http task is running.
- For SSJS / Java debugging add the following variables to notes.ini (only for development servers)

  JavaEnableDebug=1

  JavascriptEnableDebug=1

  JavaDebugOptions=transport=dt_socket,server=y,suspend=n,address=8000

  Port 8000 will need opening up for the Domino Designer debugger to access. If opening an alternate port, amend the notes.ini as required.

5.2 Server Management / Optimisation / Troubleshooting

- Server-wide xsp.properties can be set by going to <domino>\data\properties. Create a file xsp.properties based on xsp.properties.sample, uncomment and amend settings as required.
- All XPages-related messages begin **HTTP JVM: CLFAD**…. Not all are errors.
- Brief XPages-specific errors are logged to server console.
- Full errors and stack traces are logged to <domino>\data\IBM_TECHNICAL_SUPPORT folder, in files beginning xpages_exc. Developers will need access to these files. OpenNTF projects XPages Log Reader and XPages Debug Toolbar give access to these files.
- Java security for XPages applications can be managed by modifying java.policy or adding a java.pol file in <domino>\jvm\lib\security folder.
- XPages Java memory management can be done by using XPages Toolbox project on OpenNTF or a standard Java profiler such as YourKit.
- HTTPJVMMaxHeapSize notes.ini variable can be modified to increase Java memory available to XPages
- XPagesPreloadDB=filepath/xpages.xsp,server!!filepath/xpages.xsp in notes.ini can be used to preload an XPage on the current server or a remote server.
- XPagesPreload=1 in notes.ini will load up parts of the XPages runtime when the HTTP task starts. XPagesPreloadTrace=1 in notes.ini will print out more detailed preloading information to the server console.
- Issuing the command "tell http xsp show modules" to the server console will list the XPages databases and modules currently in use.
- Issuing the command "tell http osgi diag *plugin_name*" will identify any unresolved constraints preventing an Extension Library from being loaded.
- Issuing the command "tell http osgi ss *plugin_name*" will give the current status of an Extension Library.
- The "XPages Portable Command Guide" book is specifically aimed at Domino Administrators and gives full information on notes.ini and xsp.properties settings. There are also relevant chapters in "Mastering XPages 2nd Edition".

5.3 Deploying XPages Extension Libraries to Developers / End Users

- Add the Extension Library (aka OSGi plugin) to a database based on the **Eclipse Update Site (9)** template (updatesite.ntf). The features can be disabled or the plugin can be uploaded to a database not reference by the OSGI_HTTP_DYNAMIC_BUNDLES notes.ini variable, if it should not be deployed on the server.
- If a widget catalog is not already in use, create one based on the **Widget Catalog (9)** template (toolbox.ntf).
- Ensure **Show Widget Toolbar** and the My Widgets panel is checked on the Widgets tab of Notes preferences.
- Use the **Start Creating Your Own Widgets** link or **Get Started** option in the drop-down menu.
- In the **Start Configuring Widgets** dialog select **Plugins and Features.**
- Add the URL of the Update Site database and select the relevant features.
- Complete the dialog, giving the widget an appropriate name. The widget will be installed into your Notes Client. Right-click it and select **Publish to Catalog,** selecting the Widget Catalog and assigning an appropriate category.
- The plugin(s) can then be manually installed by developers / end users, by selecting the relevant widget from the Widget Catalog.
- Deployment can be automated by using a Desktop Policy – the preferred route to ensure updates are automatically installed. The widget catalog and relevant category / categories can be defined on the Widgets tab.
- See “XPages Extension Library” pp40 onwards for full instructions and screenshots.

5.4 XPages Run On Server (For Notes Client Access)

- Domino server and Notes Clients must be 9.0 or higher.
- On launch tab of Database Properties, set Notes Client to launch the relevant XPage.
- Check **Run server-based XPages apps directly on server**.
- Test by adding a reference to the database in **Run on Server** section of **XPages Performance** tab of Notes Client preferences and opening the database. Ensure a host name is set for the server – a default of www.server.com is used.
- Deploy the XPages Performance setting via a Desktop policy. On Custom Settings > **Managed Settings** tab add a new value.

  Plugin name is **com.ibm.xsp.rcp.**

  Item is **com.ibm.xsp.rcp.perf.runon**.*server.folder.filename*.**nsf.**

  Value is *folder\filename.nsf|server|***host***|protocol://serverHostName*.

  Modifying the setting does not load it correctly, so if you need to modify, remove the setting and add it again.

About Intec

At Intec we collaborate with our clients, bringing together business insight, significant experience and technology to provide a distinct advantage in today's rapidly changing business environment. Through our integrated approach to problem solving, solution design and execution we help turn our customers strategies into action. With over 25 years experience our mission is to help our customers to anticipate change and profit from new opportunities.

intec

Intec Intec House St Nicholas Close Fleet
Hampshire GU51 4JA
tel. +44 (0)1252 775400
fax. +44 (0)1252 775444
www.intec.co.uk

ISO 9001 REGISTERED FIRM

INVESTOR IN PEOPLE

Premier Business Partner IBM