



# Modernisation Options for IBM Domino 9.0.1 Feature Pack 8+

Produced by Intec Systems Limited

# Contents

1	Introduction	1
2	Options for Modernisation / Continuance	1
2.1	Notes Client / ICAA	1
2.2	XPages	2
2.3	IBM-Highlighted Partner Products	3
2.3.1	Darwino	3
2.3.2	We4IT's Aveedo	3
2.3.3	Sapho	4
2.4	JavaScript Application Options	4
2.5	REST Access	4
2.5.1	SmartNSF	4
2.5.2	GraphQL	4
2.5.3	Custom REST Services	4
2.6	Java Application Options	5
2.6.1	OSGi Plugin	5
2.6.2	Spring Boot	5
2.6.3	Vert.x	5
3	Analysis Tooling	6
3.1	Panagenda's ApplicationInsights	6
3.2	Intec RADAAR	6

# 1. Introduction

In May 2016 Intec published a whitepaper entitled “Migration or Modernisation” which detailed functionality added in more recent versions of IBM Domino as well as considerations for deciding whether to migrate all or part of a Domino estate. The main outcomes were:

- The importance of extending appropriate functionality to newer platforms (web, mobile etc). Since May 2016, microservices and cognitive have become additional platform disruptors.
- The importance of ensuring a user experience that does not feel dated in comparison to other business and consumer applications.
- Being mindful of the benefits of Domino, e.g. backwards compatibility, data security etc.
- Data migration is a significant undertaking and often a hidden and not inconsiderable cost. There are ways to migrate an application without requiring migration of the data.

On this last bullet point, historically, “database” and “application” have been synonymous and interchangeable terms in the context of Domino. For many modern Domino developers, there has become a semantic difference, if not a linguistic difference. Even where both are NSFs, the “database” is the NSF where the data and scheduled processes reside, the “application” is the NSF where the user interface and business logic reside. Increasingly these have been two separate NSFs and it is more important than ever to break free from the concept that they must be the same NSF.

At IBM Connect in February 2017, after a lengthy review, IBM announced their message on Domino application development modernisation and a number of reviews were published, including Intec’s own. The message was not a “silver bullet” approach, but an overview of potential approaches, some of which would be appropriate for some applications, some for others.

Open source was at the heart of the message and, although some Domino customers may have resisted open source in the past, that attitude is out-dated and unsustainable going forward. Whether modernisation or migration is the route, whether custom development platform or off-the-shelf, open source software will play some part in whatever solution is chosen.

“Choice” and the importance of “content” were two key tenets of IBM’s message. The choices available may be confusing, so the purpose of this whitepaper is to give some more context. Of course, an IBM Business Partner specialising in application development will be able to help offer more information and introduce you to partners, if you would like to investigate further third-party products.

## 2. Options for Modernisation / Continuance

### 2.1 Notes Client / ICAA

Modernisation does not necessarily mean replacing desktop client access to applications. An alternative to the full Eclipse-based Notes Client is ICAA (IBM Client Application Access). This originally followed a browser plugin model, hence the original planned name “Notes Browser Plugin. But when browsers stopped supporting a plugin architecture, an alternative lightweight approach was embraced. ICAA is a much more lightweight client, comparable in size to popular desktop clients of Slack, a persistent chat that has gained significant prominence and been a major disruptor for IT. Even when browser access is available, recent software releases have shown there is often a preference for desktop clients. This is understandable when auto-upgrading browsers regularly introduce breaking changes and fixed browser versions require all applications used by it to be upgraded at the same time, which is never achievable.

Even in an “always-on” era, some applications or environments will still require offline access. Wireless or mobile networks may be unavailable or inadequate. For these scenarios, even though it is by no means the only option as will be covered later, the Notes Client may still be the obvious or easiest solution. But this does not mean an application can’t be modernised and should be left to languish.

Domino and Notes have a strong reputation for backwards compatibility and many business-critical applications continue to work without any significant redesign of architecture or user interface. Some Notes applications look 15 years old, because no changes have been made to the user interface in that time. A little effort to modernise the user interface can give a not insignificant improvement in how modern the application looks and feels. In terms of modernising navigation, this could be small less ambitious enhancements like replacing Navigators with Outlines or moving to a more technically ambitious navigation model. Amending form layout and styling is likely to take more effort, but can ensure an application feels more fresh.

Regularly modernising the interface of any application requires effort and expenditure. But if a business is not prepared to invest in modernisation of application user interface, regardless of the platform, criticising and comparing those applications with ones that have been built more recently or modernised is unfair. Comparing and criticising the platform those applications are built upon because the applications do not look modern is even more unfair.

The effort involved in completely redeveloping a client application for web or mobile may be significant. So, regardless of the chosen approach, it is key to re-evaluate the functionality against current processes. It is important to bear in mind the devices in use against the functionality provided, because some use cases may not be appropriate to particular devices. It is also key to remember that some admin tasks critical to the functioning of the application may rarely be undertaken, some configuration may have been forgotten and taken for granted by end users. If migrating from Domino, that needs replacing. But if Domino will still be used, even if only for the database layer, some or all admin configuration could be omitted from modernisation, relying still on Notes Client for occasional or limited access. Those managing these aspects of the application will typically also be more forgiving about the user experience.

## 2.2 XPages

Web access has been available for Domino applications for over 20 years, since Domino 4.5 was released. Design elements could be hidden from mobile devices since Domino 6. But there were historically no out-of-the-box frameworks for cross-browser or cross-platform flexibility and user interface elements were often basic. In 2009 XPages was provided to give a more modern user experience, building on standard web frameworks like Dojo, jQuery, JSF and Bootstrap. With Domino 9.0.1 Feature Pack 8 this allow developers to provide a modern, responsive user interface that better supports web and mobile access from the same application design.

The obvious strength over traditional Domino web development is consistency. Traditional Domino web development required a lot of hacks, with non-traditional editors. But with XPages, integration of external Java and JavaScript libraries and modules less regularly requires quirks. For Java libraries, adding them to an OSGi plugin will overcome many, though requires some more advanced skills. In terms of JavaScript, choosing any particular framework will require workarounds when integrating modules from another framework, as found with Dojo and jQuery.

Redevelopment of complex client applications to XPages is far from insignificant. But XPages still competes well with other frameworks in terms of rapid application development. XPages does have a definite learning curve. But the appeal is a familiar platform and development IDE. XPages may have been around for a number of years and may not have been embraced as much as hoped. But at Intec we have seen a continued demand for training. This is not to say that documentation is significantly deficient. The quantity and quality of books, samples, videos, tutorials, blog posts and community support is significant and better than many alternatives I encountered recently. As with any framework or technology, training typically gives developers a more intensified introduction to aspects and, when done well, gives recommendations for additional resources when developers inevitably encounter issues.

As experts have got more familiar with XPages and moved beyond the basics, the architecture of an XPages application has changed. User interface has become less aligned to document architecture, so a single XPage often integrates visible or editable data from multiple documents. Many developers have embraced an MVC approach, where the presentation for the user and the business logic and business objects are separate, meaning future modernisation of interface is easier. And UI has often been separated from the data NSF.

Modernising a traditional Domino web application – either redeveloping in XPages, updating for modern browsers or migrating to a new platform and framework – can be a significant task. But a hybrid approach has proved popular, integrating XPage elements into traditional Domino web applications.

If XPages is used as part or all of applications, it is more important than ever to keep underlying platform up-to-date. As previously mentioned, browsers can introduce breaking changes. Browser manufacturers have also proved quite draconian in supporting older versions of frameworks. The most recent version of Domino will ensure core components and framework versions work on the most recent browsers. Those underlying frameworks like Dojo and Bootstrap may require changes in coding, but by building on the component-based XPages framework, the framework can abstract some of those changes and minimise the effort of upgrading.

## 2.3 IBM-Highlighted Partner Products

During the Opening General Session at IBM Connect, in addition to XPages IBM highlighted three partner products for modernising applications. Of course, there are other tools available, like redpill Now.

### 2.3.1 Darwino

Darwino was one of the three products announced. Its key USP is bi-directional support with Domino, offline access to applications and microservices.

The tool is not aimed at automated modernisation of one or more applications. Nor is it aimed at migration, although the Darwino application integrates with a NewSQL database. The NewSQL database synchronises with Domino and mirrors similar NewSQL databases used for mobile offline storage. As a result, the tool is particularly relevant for exposing part of an application to mobile, whether offline and online.

User-driven business logic and user interface will need re-coding. Business logic is Java and user interface is typically via a JavaScript framework like Vue.js, Angular or React. The output can then be deployed to a JavaEE web server (e.g. Tomcat, Bluemix, Azure or Websphere Liberty), online or more importantly offline iOS or Android mobile, and offline desktop. Although it requires redevelopment of much of the application, the database layer – document architecture, security and background agents – can continue without changes. Notes Client access can also continue.

The Darwino product provides “write once, run anywhere” code conversion, although platform-specific functionality like specific iOS components can also be added. It also includes libraries for customising replication, managing authentication, supporting various NewSQL backends like SQLServer, JSON parsing, mail handling, preference handling, GraphQL and standard REST integration, microservices, filters, querying, scripting and server environment management.

### 2.3.2 We4IT's Avedo

Avedo provides an analysis and modernisation wizard for web and mobile, to quickly convert both simple and complex applications to an XPages-based interface. The ScanAlzr element (with a free initial analysis of templates) will review analysis and complexity, giving an estimate of effort to migrate.

The wizard migration boasts time savings of up to 70%. Workflow can be modelled via a visual interface and business logic can be coded in a variety of languages including LotusScript and @Formulas. The tool also provides multilingual setup, charting, and MS Office integration. There are also some applications available out-of-the-box, currently CRM and Project Management.

The platform is not limited to on-premises Domino. The application is accessible via Domino or Bluemix, with plans to support a variety database options including Domino, RDBMS, Cloudant or MongoDB. The product is also available as a SaaS offering in private cloud.

### 2.3.3 Sapho

Sapho's product provides micro apps on web or mobile. It can expose Domino applications to mobile, web and messenger applications. It can trigger alerts out of Domino. It can integrate non-Domino datasources into Domino applications. It can also provide proactive dashboards from both Domino and other data repositories.

The micro apps are built with a drag and drop application builder, including complex elements like charting and signature blocks. Furthermore, IBM Domino is just one of many connectors available, alongside IBM DB2, IBM Watson Workspace, Box, Microsoft Active Directory / Azure Active Directory, Microsoft Sharepoint, Salesforce, Oracle E-Business Suite, PostgreSQL, Slack, Trello, Google Docs, MailChimp, SAP ERP and others in development. This means it can be a powerful co-existence tool for all areas of the enterprise to bring modern access to part of an application with a very low technical barrier.

## 2.4 JavaScript Application Options

JavaScript frameworks like Angular and React have grown in popularity significantly over the last five years. Regardless of which framework is used, REST is the method of access to stored data.

Domino has provided basic access to view data as XML for nearly 15 years (since Domino 5.0.2) and as JSON for over 10 years (since Domino 7.0.2). However, they are not particularly appropriate for a modern application. Domino 8.5.3 introduced the Domino Data Service which gave access to view and document data as JSON. Again, they were not particularly appropriate for web applications because data being posted to the service is not validated and a lot of metadata that may not be desirable is sent from the service. But this does not mean moving to a different backend for REST data.

## 2.5 REST Access

There are a variety of REST service options for Domino, which is why Intec have developed a training course specifically covering REST and Domino.

### 2.5.1 SmartNSF

Announced at IBM Connect, SmartNSF is a flexible way of defining REST service routes within Domino Designer using a domain specific language. Currently in beta, there have already been a number of releases through 2017 on OpenNTEF.

Security, read/write restrictions, validation and pre- and post-processing events can be added to any REST endpoint created. Input / output can be values or computed formulas. An OpenAPI Specification (Swagger) file definition can also be generated for JavaScript developers to code against.

### 2.5.2 GraphQL

In spite of only being released by Facebook recently, GraphQL has quickly become a popular method for REST access. This is because it allows the consumer to choose what data a call should return, even including data across different business objects. Darwino has used the concept of schema-less GraphQL queries to provide flexible GraphQL access to its NewSQL database and also Domino. The queries can be built and processed both in JavaScript and Java.

### 2.5.3 Custom REST Services

The most flexible option for REST access to Domino data is to develop custom REST services in a REST servlet. This enables workflow to be managed on the server side leveraging the flexible Domino APIs available that give the database layer much more than basic CRUD access. This means a specific REST endpoint performs the workflow rather than expecting the workflow to be handled by the application layer, ensuring consistent behaviour regardless of where the REST endpoint is invoked from. It also breaks the tight coupling between user interface form and document storage element. In addition, it could allow logging of requests, which has proved useful in integrations with external products, to confirm whether or not data transfers have taken place.

There is more effort involved, that cannot be denied. But Intec have been heavily involved in custom REST services for Domino by open sourcing OpenNTF Domino API Starter Servlet on OpenNTF as well as writing documentation and recording a NotesIn9 episode on the project.

## 2.6 Java Application Options

XPages is just one of many Java frameworks available for developing a web or desktop application. At Intec, we have been using Vaadin for some years and have provided enablement published on Vaadin's own blog. But other frameworks like JSF are available. Which one you choose is a matter of personal preference.

The benefit of Java over JavaScript is that, with the right setup, code within the application can integrate directly with the Domino backend. Retrieving the data is just one aspect of application design. Although concepts like document locking can be enabled at API level, there is nothing in-built in the API to manage aspects like caching or paging. If Java access to Domino data is of interest, Intec have also delivered training courses to better enable developers.

The key choice is then how to deploy the application. Wrapping it within an NSF is unlikely to be an appropriate choice. There are then a number of options.

### 2.6.1 OSGi Plugin

If you wish to run the application via Domino's HTTP task, the usual approach will be to wrap the JavaEE application in an OSGi plugin. This is an approach Intec have used in the past. There will be some quirks, for example the Java servlet specification in use in Domino and integration with project dependency management tools like Maven. But there are blog posts that cover this and experts who can help.

### 2.6.2 Spring Boot

Spring Boot is an option for creating standalone applications and can use an embedded web server like Tomcat. If this is a preferred delivery option, it may encourage a particular Java framework for the application, e.g. Spring MVC or Thymeleaf. But there are also starter templates for building a web application with other frameworks like Vaadin. This might be a good approach for a client application, for example allowing web access to offline Domino data as an alternative approach to using Notes Client or XPages-in-Notes-Client.

### 2.6.3 Vert.x

Vert.x is a toolkit for developing flexible and reactive applications. It is not limited to Java, but can be coded in Java. Vert.x-Web is a subset of tooling for developing web applications and HTTP microservices. There may need to be some development to support a particular framework, but there are open source implementations for many frameworks like Vaadin. One of the strengths of Vert.x is that it is non-blocking which makes it very performant and scalable for server applications. Whereas some database layers like MongoDB need a specific client to support non-blocking access, the core Domino API has always been non-blocking, making Domino well-suited as a database layer. Like the Spring Boot approach, interaction with Domino only requires inclusion of specific jar files from the Notes or Domino install. There have been blog posts on integrating Domino and Vert.x (at one point it was a candidate for Verse on Premises) and Intec have continued that research.

## 3 Analysis Tooling

The first step for any modernisation is analysis, to identify what to modernise.

### 3.1 Panagenda's ApplicationInsights

At IBM Connect, IBM announced that Panagenda's ApplicationInsights would be available as a free entitlement for Domino customers whose licenses have not lapsed. This was released in April with a related FAQ technote. The entitlement is available for 12 months from April 11 2017 and analyses 50 most used application databases, although additional packs are available for opening up analysis of subsequent batches of 500 applications. The tool is available as a virtual appliance and gives insight into usage statistics and design analysis. The output provides:

- Overall usage and design complexity rankings.
- Database, replication and Access Control summary.
- Volume usage splitting between access from Notes Client and web.
- Design complexity identifying number of design elements, total lines of code broken down by language and other metrics
- Design insights identifying operating system dependencies, mail dependencies and dependencies on other databases.
- Analysis of code for readiness for use from web.
- Analysis of hardcodings for e.g. filenames, IP addresses, email addresses etc.
- Coding relying on Rich Text or notes.ini variables.

Any analysis of code cannot identify redundant code because functions may be called externally (e.g. from another database), design elements or functionality may no longer be used, code may be commented out or unreachable and functionality may not be relevant for all methods of interface. Consequently, the output needs to be reviewed in the context of an understanding of the design and an understanding of the current business usage of the application. But it can provide useful information on next steps for an application.

### 3.2 Intec RADAAR

A significant amount of summary data is accessible via core and community APIs. Intec have harnessed and extended this with RADAAR (Review and Analysis of Domino Application Architecture Robustness). The tool provides a rich analysis of application information, in a single interface. Dashboards highlight areas for further investigation. By analysing design elements and profiling data, the tool is designed to provide areas for consideration whether the end goal is continuance on a more modern Domino version, modernisation with XPages or another UI framework, or migration from Domino. The overviews can also provide a window into the application for scenarios where ongoing support is being transferred to new experts.



The output is continually being enhanced, but version 2.0.0 provides:

- Overview of database sizing, version and inheritance.
- Overview of all database properties that can be set via Properties panels in Notes Client or Domino Designer, as well as some properties that are set by server management (e.g. DAOS, 16Mb summary field limit extension).
- View analysis including dashboard highlighting areas for further review like prohibit design refresh set, private views, high number of re-sortable columns, document count and view index size concerns etc.
- Agent schedule overview at database and server level.
- Agent analysis including dashboard highlighting short or long-running agents, agents without profiling enabled, agents that have not run recently or never run etc.
- Form analysis including dashboard highlighting Forms that have replication/save conflicts, orphaned documents, potential referencing integrity issues, response documents, prohibit design refresh set, encrypted / signed fields, authors / readers fields, high numbers of documents, high numbers of fields etc.
- Profiling of documents by created date and last modified date as tables or charts. The data is profiled by year and, for the last 12 months, by month.

## About The Author

Paul Withers has been working with IBM Domino since 1998 and has guest-blogged for Vaadin Ltd. He is one of the founding contributors for OpenNTF Domino API, a Java API that extends functionality for IBM Domino. He is also a contributor to CrossWorlds, an open source project for accessing Domino data from standard web applications running on IBM Websphere Liberty Profile. Paul was co-author for XPages Extension Library, published by IBM Press (2011). He has blogged extensively about using Vaadin to access Domino data and surfacing Domino data through custom REST servlets.

## 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 we can help customers anticipate change and profit from new opportunities.



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

