Efficiently Migrating Lotus Notes Applications to Modern Platforms

Unify Corporation
Unify.com
composer@unify.com

Introduction

How rapid migration of Lotus Notes applications improves ROI, speeds application deployment, and minimizes disruption to users—while preserving software investments.

As businesses move into the era of highly scalable architectures, Web services and service-oriented architecture (SOA), rich customer-facing applications, and mobile workforces, they are discovering that previously satisfactory workgroup and workflow systems no longer provide needed capabilities. Lotus Notes, in particular, is quickly becoming viewed as a legacy platform that is difficult to shed as part of a move to modern, scalable, and non-proprietary architectures.

Modernization of collaboration platforms within Global 1000 and mid-size companies continues to grow. Analyst firm Gartner estimates that by 2010 the installed base for Microsoft Exchange will grow to 70 percent of the collaboration market, while IBM's Lotus Domino installed base will shrink to 17 percent[1].

Although Lotus Notes applications could run faster, scale better, and integrate smoothly if moved to a Web services-based infrastructure, organizations have had few ways to modernize complex Lotus Notes applications without rewriting them from scratch for the target platform. Despite this constraint, many organizations have soldiered on with their modernization plans,[2] depending in part on in-house staff, outsourcing conversions to integrators, and relying on enterprise resource planning (ERP) and customer relationship management (CRM) vendors to replace functionality with other packages.

These approaches are workable, but they are expensive and, more importantly, they are time-consuming. Once a site has decided to move Lotus Notes applications to a new platform, the faster it can complete the adoption, the faster it can realize a return on the investment. And, of course, the faster it can begin using the benefits of the new platform to optimize and extend its software investments.

The problem is that modernizing Lotus Notes applications is fraught with difficulties, complex decisions, and considerable expense. Managers usually must combine several approaches to effectuate the modernization. They retire

[1] Tom Austin and David W. Cearley, "Dissatisfaction with IBM and Microsoft E-mail and Calendaring Is Real," Gartner, Jan. 19, 2007

[2] Says Matt Cain of Gartner, Inc.: "[Microsoft] Exchange is picking up share, and Notes/Domino is losing share. I'm seeing more defections from Domino. There is migration from Domino to Exchange."

certain applications, replace some with new packages, rewrite others, and then migrate the remaining mission-critical applications.

This white paper examines migration choices for rapidly moving from Lotus Notes to Microsoft's .NET platform. It looks at how enterprises manage the transition, and it examines a solution from Unify Corporation, called Composer for Lotus Notes, Microsoft Edition, which is a services engagement that migrates complex Lotus Notes applications to .NET and Microsoft's enterprise infrastructure.

Moving Off of Lotus Notes: The Current Options

There are many factors to consider when contemplating a move from Lotus Notes—most notably the strategic question: How best to modernize? Before choosing from the range of options, managers need to fully scope the project. Most large sites have literally thousands of unique applications that were used for ad hoc projects, but that continue to live on because no one quite knows if it is safe to remove them. So, a manager first must conduct an assessment of the Lotus Notes application environment and decide which applications to retire. At large sites, this inventory process can result in the retiring of thousands of non-critical Lotus Notes applications and reduce the migration load by as much as half.

With the remaining applications, managers have three principal options: replace Lotus Notes application functionality with packaged applications, such as ERP and CRM solutions; rewrite the applications using in-house staff or outsource the work to regional or global system integrators; or migrate the applications.

Replacing and Rewriting

Managers frequently enter into analysis of their options by projecting the time and costs associated with an application rewrite using in-house staff. Additionally, a rip-and-replace option is examined in which proprietary Notes applications are replaced by ERP, CRM, or other packaged applications. Sometimes, sites consider a combination of the two, depending on how their business needs have evolved. These options can pose serious risks to project success:

- **Replacement by adopting ERP or CRM solutions.** This approach is generally part of a larger conversion to an integrated ERP or CRM solution. The software vendor or the team of outside consultants driving the project determines how the company's business needs will be satisfied by replacing the legacy Lotus Notes applications with the packaged applications. The resulting plan tends to be expensive, and it shifts the applications from one proprietary context to another. Moreover, this rip-and-

replace strategy may reflect a bias for a specific vendor that might not always be aligned with the company's business needs. Many package vendors and consultants convert only the applications that will eventually be part of the new packaged solution and leave the other applications for the IT staff to manage. To preserve the business value in those remaining Lotus Notes applications, the applications must be rewritten in-house staff, or possibly left to execute on the existing Lotus Notes architecture, thereby forcing the site to run applications on two platforms that communicate through a patched-together integration mechanism. Essentially, the IT site has been moved from one silo to another, without getting rid of the first silo altogether. Moreover, the applications that are replaced by a packaged application generally look and behave substantially different from their Lotus Notes forebears, which imposes new training requirements on users and staff.

- **Rewrite Lotus Notes applications using in-house staff and/or system integrators**. Many businesses choose to rewrite applications in-house because of their familiarity with the business logic and the perception that Lotus Notes applications are easy to convert to standard programming languages. However, this approach can run into serious difficulties. The first obstacle is that Lotus Notes applications deeply intertwine business logic with presentation code, so the programs cannot be ported until they are restructured and redesigned. When sites recognize this need and revise the timeline to accommodate this issue, they often must reduce the scope of the modernization effort—leading to an incomplete transition or reduced functionality of the ported software. Quite apart from this concern, there is a second problem: deciding what to port and in what order.

  Sites using Lotus Notes are notorious for having many different versions of the same program—distinguished only by small variations in functionality. These one-off applications need to be extensively analyzed for their differences, checked for their usage, and then prioritized for conversion. Many applications rely on these one-off variants, and the numerous undocumented dependencies can complicate efforts to determine a valid migration plan. Finally, developers find it very difficult to reproduce the Lotus Notes user interface in the ported applications and, as a result, users must be retrained when the new applications are brought online. The upshot is that conversion through a rewrite is a slow, arduous, and expensive process, and each project has the added risk of uncertain and incomplete results that require user training.

  Most large sites hire integrators and consultants to assist in developing the new architecture and in rewriting specific Lotus Notes applications. This path has similar limitations to the approaches discussed previously: rewritten applications must be reintegrated on the target platform to run correctly. The code itself tends to be uneven in quality as different devel-

opers implement conversion characteristics in different ways and use different coding conventions to solve specific problems. Significant oversight and management by the in-house team is required to ensure that the resulting code adheres to corporate and industry standards. In addition, most integrators do not have the tools to faithfully reproduce all aspects of Lotus Notes screens and report layouts, so the delivered programs look different and on occasion require retraining of users. Because the developers build similar but not identical interfaces that jettison small, valuable features, they disrupt the end users and impede the organization's ability to rapidly adopt the new platform and recognize a return on investment (ROI) from the migration. Finally, the outsourced approach to modernization through a rewrite strategy is still expensive and time consuming, and it is an open-ended proposition that seldom guarantees exact deliverables or delivery dates, much less fixed costs.

Migration

Migrating platforms usually entails significant expense and end-user disruption. Organizations seeking to gain the advantages associated with an SOA must engage in a detailed cost justification to ensure that an appropriate ROI is attainable. Once a target platform is chosen, the initial ROI often depends on a rapid adoption of the more efficient platform and a low level of disruption to business operations.

Migration could be accelerated if an organization leveraged existing investments in its legacy applications. Creating new applications that take the business logic of the legacy application and bring it forward like-for-like to the modernized application is what Unify calls "migration."

Migrating Lotus Notes applications is the least disruptive option available to organizations interested in adopting an SOA platform, as it preserves the look and feel of the Lotus Notes application on the new platform. Organizations can rapidly adopt the new platform without incurring the end-user retraining costs associated with rewriting or replacing the functionality of critical Lotus Notes applications.

The issues related to the replacement and rewriting of mission-critical Lotus Notes applications make them costly options when minimal end-user disruption and rapid platform adoption are priorities. Moving to a modern SOA platform requires all four options discussed in this section—retire, replace, rewrite, and migrate—to ensure a successful result.

Unify's Composer for Lotus Notes service enables organizations to migrate complex Lotus Notes applications to an SOA platform in less time and at less cost than a replacement or rewrite strategy. Composer's ability to preserve the user interface as well as the behavior of the application helps to minimize

disruption to the business and enables organizations to extend and optimize the migrated Lotus Notes application.

Composer for Lotus Notes (Microsoft Edition)

Composer for Lotus Notes, Microsoft Edition is a service that migrates Lotus Notes applications in their entirety to Microsoft Windows and the .NET platform. It migrates the components of the Lotus Notes architecture to the corresponding Microsoft technologies—thereby eliminating Lotus Notes lock-in and providing superior scalability and enterprise access:

- Business logic is migrated to Microsoft's C# programming language.
- Business logic can be exposed as Web services using an SOA to provide maximum access and integration with other software.
- Lotus Notes flat data files are migrated to a centralized relational database. Microsoft SQL Server is used by default; however, organizations can also choose databases such as Oracle Database 11g, IBM DB2, and so on.
- The Lotus Notes GUI is migrated to Microsoft ASP.NET using HTML forms that have AJAX and Rich Internet Application (RIA) interfaces. As discussed shortly, these interfaces are feature-identical with the original Lotus Notes screens, so that no user retraining is required.
- Security and access control are ported to Microsoft Active Directory services.

The resulting application is a complete Microsoft-based implementation that can be integrated easily with other internal IT applications and enterprise packages (see Figure 1). Access to the migrated software is enhanced by the Web services framework included in the solution. As a result, the concept of siloed applications no longer exists and the legacy issue becomes a non-issue.

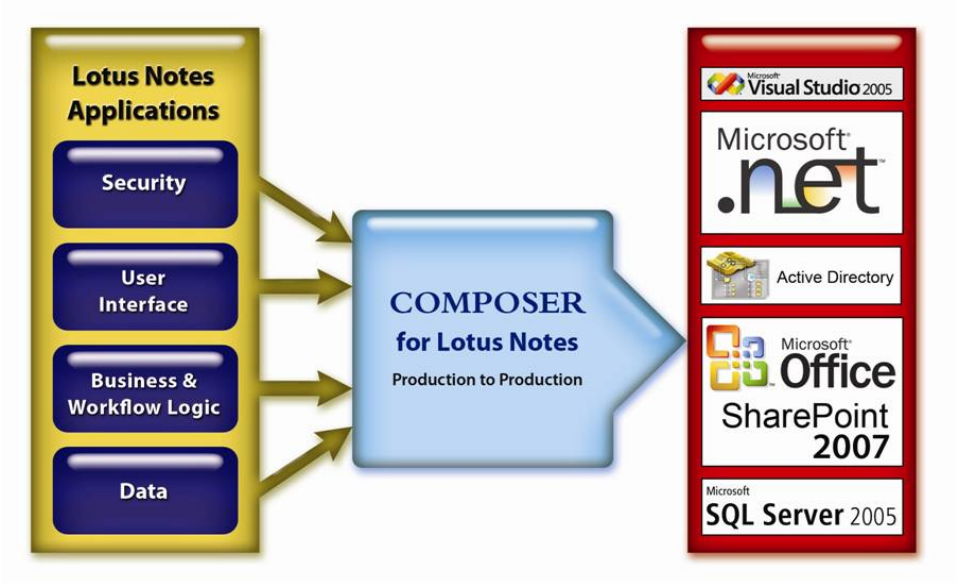

Figure 1. Unify's Composer for Lotus Notes delivers an all Microsoft, all .NET migration.

Composer for Lotus Notes provides additional features that prevent vendor lock-in. These features include:

- Complete source code with embedded comments that contain the original Lotus Notes code
- Naming of variables in C# and ASP.NET that parallels the original variable names in the Lotus Notes applications, so that developers familiar with the original programs can locate the functionality easily in the migrated code
- Complete database layouts
- Complete documentation of the Web services and SOA framework

By enabling developers and architects to access and understand the application code and the implementation architecture, Composer minimizes the retraining of the technical staff. Furthermore, Composer allows the technical staff to preserve their knowledge of the business processes and leverage it in new ways.

Unify's commitment to minimize retraining at all touch points is a cornerstone of the solution. Composer offers far more than rip-and-replace strategies. By preserving the user experience with a similar user interface and corresponding business logic and workflow, sites can deploy the migrated Lotus Notes applications as they wish and according to their own schedule—without the delay and expense of retraining. Figure 2 demonstrates how closely the migrated screens match those of Lotus Notes.

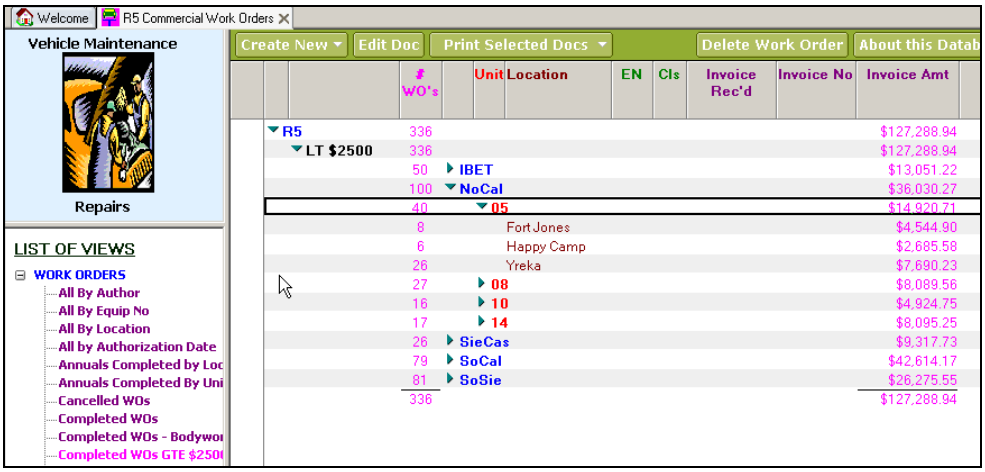

Figure 2a. A Lotus Notes screen before migration by Composer.

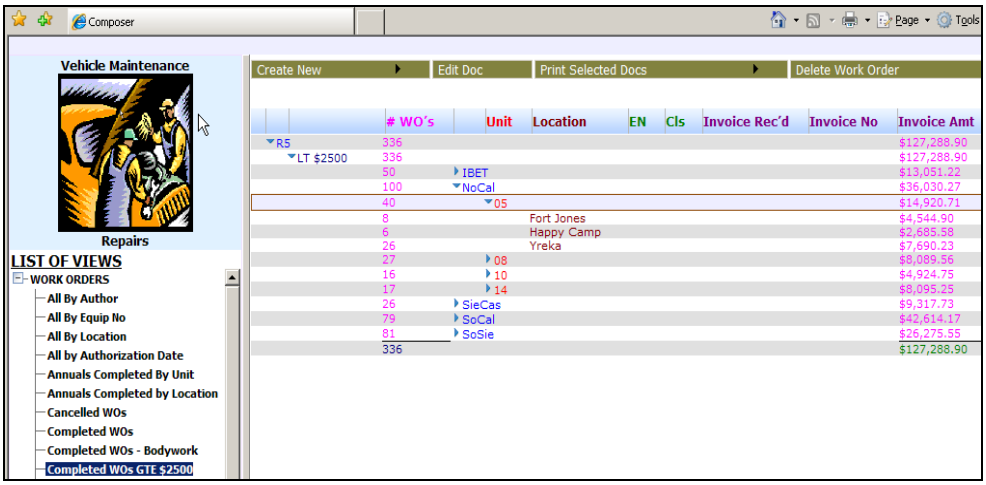

Figure 2b. The same screen migrated and running on Microsoft .NET. Notice that even the famous Lotus Notes "twisties" have been ported and work correctly.

As Figure 2 illustrates, the forms, menus, and icons are identical to Lotus Notes, so that users can be productive immediately in an environment that they recognize at first sight.

Faster Time to Market and Quicker ROI

The most immediate benefit of using Composer is time to market. Composer is a service provided by Unify, so the conversion and migration of applications can be accomplished quickly without tying up developer and technical resources for months. Rapid adoption of the new platform enables organizations to begin recognizing the ROI identified in the cost justification model at project inception, and to shut down the license costs on their Lotus Notes products.

On this point, Stephen Konopelski, vice president of IT at Mutual Boiler Re, one of the largest equipment reinsurance firms, says: "We were able to achieve our time-to-market goals and realize considerable cost savings because of Composer's automated migration of the database, user interface, business logic, and all security profiles. The loosely coupled SOA framework delivered by Composer offered us great flexibility for reuse and expanding into other user interface options such as a portal framework, handheld device, or a traditional client desktop."

The ported applications are architecturally superior to the Lotus Notes applications. Beyond the fact that they are no longer siloed, the ported applications are now scalable. For example, the flat-file Lotus Notes database, which represents a substantial performance bottleneck, is replaced with a true enterprise database that delivers high performance, even under heavy loads. The SOA architecture means that the migrated applications can now interact with other enterprise applications—thereby making it easier to integrate them into the corporate workflow. Many sites find that because of the SOA design, new functionality is easy to add, and old programs that bridged the Lotus Notes applications and other packages can be eliminated.

Time-to-ROI is further diminished by the benefits of running the software on the new platform. Applications can be maintained more easily, optimized, and, if necessary, expanded to meet new business needs. Several factors contribute to this time-to-ROI benefit:

- The ease of finding developers who know Microsoft .NET technologies compared with the increasing difficulty of finding Lotus Notes developers. Likewise, it is easier to find managers and administrators familiar with Microsoft enterprise technologies.
- Because the new applications run on multiple platforms (.NET, Web server, RDBMS), individual components can be scaled as needed.
- Because the migrated software runs on standard Microsoft platforms, organizations can use many off-the-shelf products to extend and develop the applications. In addition, should the need arise, portions of the new solution can be migrated to other platforms—other databases, or execution frameworks—with little difficulty.

Rapid migration, open architecture, superior management—these are attributes rarely associated with Lotus Notes applications. But for sites that have existing investments in Lotus applications, all of these attributes can be achieved with Unify's Composer, thereby preserving the value of investments in Lotus Notes on the new platform.

Moving Forward

Moving from Lotus Notes to a modern enterprise platform can be a difficult process. Inevitably, it requires retiring or replacing a large number of applications and converting the rest. Many sites make the mistake of using the conversion as an opportunity to rewrite the functionality of the existing applications. This approach, while necessary in some circumstances, is almost always too expensive and too slow to be practical in large modernization programs. Best practice suggests migrating the suitable applications first and optimizing or expanding them after they are running successfully on the new platform. This approach minimizes the retraining of users, who are often leery about moves to new platforms. And it resolves many of the problems and limitations of the Lotus Notes platform. For example, if a driver for migrating Lotus Notes applications is performance, the migration to the .NET platform will address the performance limitations; likewise for scalability. The key is to migrate the suitable applications as quickly as possible, so an organization can begin enjoying the benefits of the new platform and recognizing the ROI it provides.

Composer for Lotus Notes, Microsoft Edition is a complete solution to the problem of migrating complex Lotus Notes applications to an SOA platform. (Composer for Lotus Notes, Java Edition is available in addition to the Microsoft .NET version.) Composer provides a non-disruptive approach that results in complete “production-to-production, like-for-like” applications that have identical user interfaces, so that neither users nor managers need to be retrained to operate them. And because Composer is delivered as a service by Unify, it provides a fast solution, using proven techniques and technologies, that does not tie up developers and in-house staff for long periods of time. Composer has been deployed repeatedly at small and mid-sized businesses as well as large enterprises, so managers can rely on Composer to fully complete the migration from Lotus Notes on budget and on schedule.

For more information on how Composer for Lotus Notes can change the game of migrating Lotus Notes applications, contact Unify at (800) 248-6439 or go to Unify.com.

Unify Corporation
2101 Arena Blvd., Suite 100
Sacramento, CA 95834 USA
Phone: 1.916.928.6400
Toll Free: 1.800.468.6439

United Kingdom: + 44.175.3245510
France: +33.1.345.82830