

October 2003



Lotus software

WebSphere software

Domino applications and the Lotus Workplace technical strategy



Contents

1	Introduction
4	Lotus Workplace application servers
4	<i>Lotus Domino</i>
7	<i>The WebSphere platform</i>
8	<i>WebSphere Application Server</i>
10	<i>WebSphere Portal framework</i>
11	Application platform considerations
12	<i>Application deployment topology</i>
13	<i>Application purpose and use</i>
17	<i>Application life cycle</i>
18	<i>Environmental factors</i>
20	<i>Scalability requirements</i>
21	<i>Data requirements</i>
24	Leveraging your investment in Domino
24	<i>Domino integration with WebSphere</i>
26	<i>Leveraging your Domino data</i>
29	<i>Projecting Domino applications through portlets</i>
30	<i>The future of rapid application development</i>
32	Conclusions
34	<i>Related publications</i>

Introduction

In the spring of 2003, IBM announced the IBM® Lotus® Workplace platform as the new offering for collaboration and human interaction. This platform will give people access to the information and business processes that they need, in a single, secure and dynamic environment. The goal of Lotus Workplace is to:

- *Facilitate collaboration and human interaction across all areas of a business*
- *Further enhance employee productivity*
- *Enable widespread teams to work together efficiently in context of the business process*
- *Provide the people-integration component of e-business on demand™*

IBM believes that a new work paradigm will emerge in which employees collaborate with each other “in context” of business processes. Lotus Workplace supports this paradigm by using a portal framework to tie together key applications and functions based on an employee’s role, giving a peripheral view to important events or information outside the user’s current focus. Instead of using separate, stand-alone tools for collaboration, Lotus Workplace users will have features for collaboration at their fingertips wherever they work. The result is that businesses will have a platform that allows them to meet the demands of their customers and partners quickly.

To implement the Lotus Workplace vision, IBM Lotus is planning on delivering modular collaborative services for easier cross-IBM-product integration and easier integration of IBM products with third-party and custom applications. The technical implementation is a collection of services-oriented interfaces for all Lotus server products. The Lotus Workplace platform contains infrastructure elements provided by Lotus Domino™, IBM WebSphere® Application Server and IBM WebSphere Portal software.



The Lotus Domino application development and deployment environment enables you to develop collaborative applications quickly and to take them offline, bringing people, processes and data together to facilitate both productivity in e-business and quick decision-making. This means that Domino applications are an integral part of the Lotus Workplace platform. Existing custom applications built with Lotus products will support the Lotus Workplace platform, allowing further leveraging of your application investments. Lotus will continue to enhance the Domino application development model and data store (Network File System [NFS]) and in the future, will enhance it by providing IBM DB2® database management as an alternative data store.

In the IBM tradition, Lotus Notes® and Domino customers will benefit from comprehensive support for the foreseeable future. Additionally, Lotus Domino will increase its support of the Java™ 2 Platform, Enterprise Edition (J2EE™) and infrastructure standards, such as Java Server Page (JSP™) tags, Java application programming interfaces (APIs), Lightweight Access Directory Protocol (LDAP) and relational database (RDB) integration to assist developers interested in working in both the Domino and WebSphere application environments.

There are numerous advantages to using the J2EE platform. The majority of the industry's major players have begun writing applications for this open development platform. This means that the applications that you develop will have the ability to interact in a much larger environment. Using a standards-based directory and data store, J2EE helps reduce your total cost of ownership (TCO) while improving scalability and flexibility. Because J2EE is a layer on top of the operating system, it allows for application portability by enabling a developer to build an application for the J2EE platform and not for a specific operating system or device.

The WebSphere platform leverages the J2EE specification as an application development server for the Lotus Workplace platform (see Figure 1 on page 3). J2EE provides a specific architecture for building, deploying and managing



applications in multiple tiers, often broken into presentation, logic and data. This architecture is designed to provide scalability, flexibility and manageability. Although J2EE is a rich application-development platform, it has very few features to support collaboration, so it benefits from having Lotus Domino to provide rich collaborative capabilities. Applications designed to use Lotus Domino and WebSphere software blend powerful collaborative features with significant transactional scalability to deliver end-to-end e-business solutions.

Lotus Workplace platform

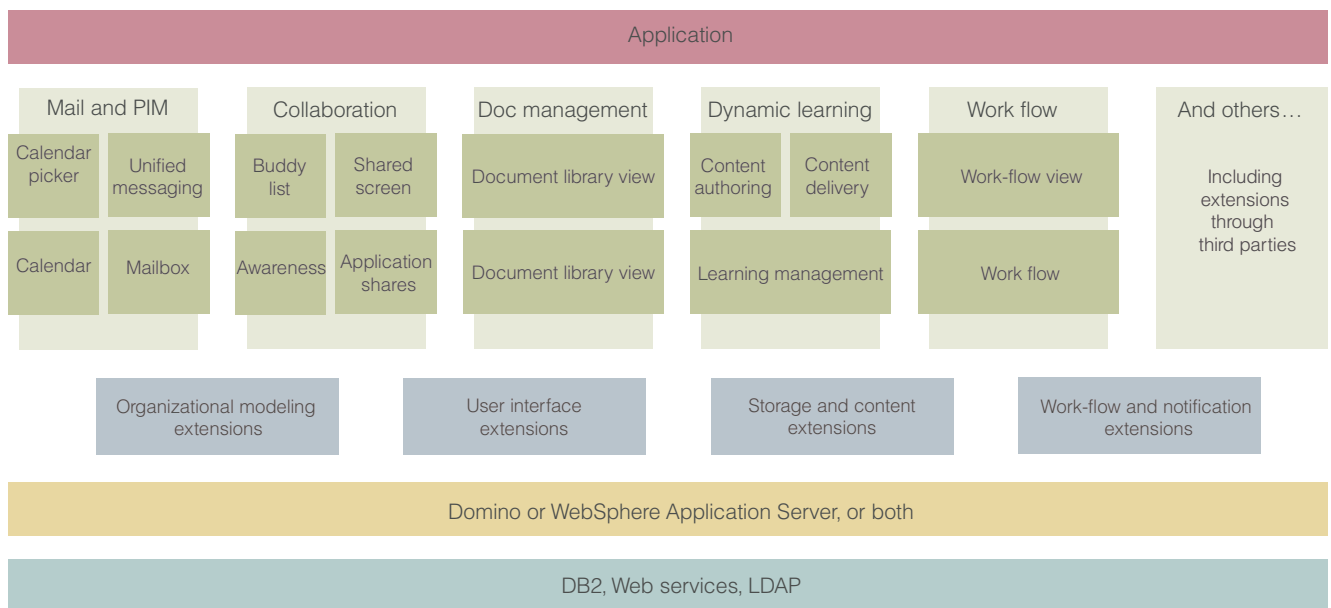


Figure 1. The Lotus Workplace platform.

WebSphere Portal provides the framework for bringing together the modular components of the Lotus Workplace platform. It will provide both Lotus Domino and WebSphere developer communities with a modular architecture for building integrated collaborative solutions. Domino developers will be able to leverage the collaborative services provided by Lotus Workplace to augment and to enhance



their applications. WebSphere and J2EE developers can integrate collaborative capabilities from Lotus Domino and other Lotus products through Web services. In both cases, integrating collaborative solutions becomes easier and can be done with standard developer skills.

The Lotus role is to develop software that not only makes new things possible, but also makes them easier and less expensive to deliver. This is done by allowing collaborative solutions to be integrated virtually anywhere in the enterprise to boost productivity and communication in all areas of your business.

The remainder of this paper will provide developers, CTOs and other IT decision-makers with information on the tools and technologies used to support Lotus Workplace. This includes information about the application server platforms available for use with Lotus Workplace; a section on application platform considerations to assist with selecting the appropriate platform for your application; information on how to leverage your investment in Domino; and a brief overview of the future of rapid application development.

Lotus Workplace application servers

Lotus Domino

Lotus Domino is a comprehensive application platform for collaboration that handles both connected and disconnected requirements for data and applications. Most customers initially purchase Lotus Domino for the built-in enterprise e-mail, calendar and scheduling applications, making those types the most-widely deployed collaborative applications. However, the majority of customers exploit the “more than mail” capabilities that support core business processes, which enable employees to work together efficiently and securely. Lotus Domino is comprehensive because it provides the complete infrastructure needed to create, test, deploy and manage distributed, multilingual applications including directory, database, application server, administration, security, connectivity, Web server, e-mail server, calendaring engine, and so on – all in one system.



Domino developers can design applications for the Lotus Notes client, Web browsers, mobile phones and handheld devices, or most commonly, for a hybrid environment accessed by multiple types of clients. Hybrid-client Domino applications can leverage replication and offline services for secure, synchronized applications that work as well in a disconnected mode as when accessed on a server over a network. Replication enables users to save a local copy of a Domino application and its data on a file system and to periodically synchronize the data, so that users can be productive and efficient even when they are disconnected from the network.

Examples of Domino solutions include document-centric and work-flow process routing, such as project teamrooms, document repositories, discussion forums, sales-force enablement and employee self-service applications. Businesses of all sizes have benefited from Domino applications.

As a comprehensive application platform, Lotus Domino includes a tool for rapid application development (RAD), a document-based object model and broad programming language support for building custom collaborative applications. With these choices, your organization can leverage many developer skills to develop a Domino application. Domino developers can quickly build a collaborative application by applying one of the many templates that ship with Domino.

If requirements dictate application functionality beyond the one provided by one of the included templates, that application can be modified or new applications developed using Lotus Domino Designer. A RAD tool, Domino Designer provides an intuitive integrated development environment for developing and managing your Domino applications.

Domino applications can implement business logic through the use of formula or the BASCI-like LotusScript language on an event-driven or scheduled basis. For more-advanced solutions, developers can use Java, Microsoft® COM, C/C++,



or CORBA. The multiple interfaces to a single-object model that Lotus Domino provides enable developers to pick the best language for the task, reusing their skills in new applications and solutions.

Some solutions require non-Notes data and globalization support. Lotus Domino add-on tools facilitate such solutions. Using visual data-mapping techniques, a developer can easily and quickly integrate relational data with Domino data. No programming is required when using tools like Lotus Enterprise Integrator[®] (LEI) to integrate data from a wide variety of systems like IBM DB2, Oracle and Microsoft SQL Server. Domino applications that require global deployment can be translated into a variety of languages by using straightforward forms. The same application can be delivered to multiple users in their native language.

The Domino roadmap for application development builds on the fundamental premise that Lotus Domino is a flexible and open platform, which is demonstrated by Extensible Markup Language (XML) and broad programming-language support. Flexibility and openness are key to a Domino application's ability to leverage J2EE and the Lotus Workplace platform. You can extend your Domino application investment in data and application logic in a number of ways. For example, you can expose Domino data using Web services (through LotusScript or Java) or JSP tags to integrate with Lotus Workplace or WebSphere applications. Another option is to surface your Domino application directly in WebSphere Portal.

Along with the progressive innovation that Lotus will add to Domino Designer and to the Domino programming model, there will be continued integration with WebSphere Studio and WebSphere Portal development tools. This integration will not replace Domino Designer, but will help facilitate teams of developers using Domino, WebSphere Application Server and WebSphere Portal in their environments. Such teams will benefit by using the strengths of each system when building applications. For additional information about future application development and integration points between Domino Designer and WebSphere Studio, see "Leveraging your investment in Domino" on page 24.



The WebSphere platform

The WebSphere software platform for e-business delivers one of the most powerful and flexible Web application servers on the market, partly because of the broadest implementation of the widest range of leading-edge open standards. The application server is complemented by a range of products that leverage this foundation to provide functions, such as personalization and mobile computing. The WebSphere software platform also includes an award-winning line of tools, including IBM WebSphere Studio, which provides a highly integrated development and deployment environment. The platform is organized into three areas of functionality:

- ***Foundation and tools*** for building, running and deploying applications. *WebSphere Application Server, WebSphere MQ messaging and state-of-the-art development tools form a solid base for the platform. The foundation and tools provide the Internet expertise that you need, enable you to build and use Web services, and link you to a greater technical community of developers and other WebSphere users.*
- ***Business integration*** for integrating internal business processes, including processes that involve business partners. *WebSphere offerings such as WebSphere Business Integrator make it easy for your company to implement applications and business processes, including supply chain management (SCM) and the integration of existing processes with the Web.*
- ***Business portals*** for personalizing Web-based content and for making this content accessible to any device. *These WebSphere products fine-tune your users' experiences and provide broad access for your customers, employees, business partners and remote branch offices. WebSphere Portal leads the business-portals part of the WebSphere Platform. It provides an extensible framework for interacting with enterprise applications, content, people and processes. Self-service features allow end users to personalize and organize their own view of the portal, to manage their own profiles, and to publish and share documents with their colleagues. WebSphere*



Portal provides additional services such as a single signon, security, Web-content publishing, search and personalization, collaboration services, enterprise application integration, support for mobile devices, and site analysis.

WebSphere Application Server

The need for standards

In the early stages of e-business, application servers, integration servers and customer relationship management (CRM) servers all had their own technology stacks. Over time, it became clear that the core “engine” required to build, deploy and manage any application containing new business logic could be defined as a single set of core application-server functions. And, if this common engine could be standards-based, it could more quickly and easily become the foundation for any number of specialized software packages.

As organizations invest in a J2EE server platform, choosing the right one for their environments can seem a daunting task. Yet, although there may be many choices, the core functionality of all J2EE servers remains more or less the same. These core elements of the J2EE application-server model are put forth by the J2EE specification and provide a vast array of services to be provided to and handled for the application that runs on it. The server handles the life cycle of the various application elements, providing them with current state and context information, authentication information about the current user, and transactional context for highly sensitive applications.

It's where the J2EE specification leaves off that a J2EE server vendor can provide enhancements that make one server more appealing or appropriate to a certain customer's environment. This gives server vendors the ability to develop value-added functionality without impacting the standards with which the servers must comply. Therefore, the development model for each server from various vendors is similar and allows customers to leverage the skills of administrators and developers from any one vendor's J2EE server to another vendor's server.



WebSphere Application Server provides features supporting dynamic applications, improved ease of management and support for the latest J2EE and Web services standards. It also addresses scalability and performance with automated performance-tuning and load-balancing services. WebSphere Application Server is available in several configurations across a wide range of platforms, eliminating the need to over-invest today yet maintaining growth options for the future.

In contrast to Domino, WebSphere Application Server does not contain its own data manager. Instead, it relies on a relational database being available in the environment for its applications to store information to. One way that WebSphere Application Server provides access to those data sources is by leveraging the J2EE Connector Architecture (JCA). JCA defines a set of service contracts that a connector developer can expect will be available to the adapter at application run time. The three services defined in JCA V1.0 include:

- *Connection management*
- *Transaction management*
- *Security management*

Domino and J2EE

One of the key differences between WebSphere Application Server and the Domino application server is that Lotus Domino provides a fully integrated environment; application execution, user authentication, directory services, data hosting and presentation display all in one system.

The J2EE model differs in that it has elements for providing the application with all the same information, but the J2EE server is not responsible for fulfilling all the aspects that the Domino server fulfills. The J2EE server calls out to different parts of the customer environment to fulfill the requests of data, directory information, and so on. For example, whereas the J2EE specification outlines how the server can get data from a data store into the application using the JCA or Java Database Connectivity (JDBC), J2EE does not require that the server contain the database manager itself.



WebSphere Portal framework

The portal vision

Portals serve as a simple, unified access point to Web applications. Portals also do much more: they provide valuable functions like security, search, and enable collaboration and work flow. A portal delivers integrated content and applications, plus a unified, collaborative workplace. Indeed, portals are the next-generation desktop, delivering e-business applications over the Web to all kinds of client devices. A complete portal solution should provide users with convenient access to everything that they need to get their tasks done anytime, anywhere, in a secure manner. Portals provide the tools and user interface to access information and applications, and to manage the selection and personalization of content.

WebSphere Portal architecture

WebSphere Portal is one of the industry's most-comprehensive portal solutions, and represents the de facto standard e-business architecture. WebSphere Portal integrates both IBM and Business Partner technologies to realize this architecture. IBM is also extending its portal offering to create additional products that deliver highly personalized and context-sensitive applications, accessible from any device, anytime. As e-business applications enter the on demand era, WebSphere Portal leads the way with its concepts of delegated administration, cascading page layouts, portal federation through Web services, advanced portlet-application concepts, business-process integration, knowledge management and advanced personalization. In complementary offerings, additional pervasive computing functions are enabled, such as intelligent notification, offline browsing and data synchronization.

WebSphere Portal is available in three editions, each designed to provide the infrastructure that you need to build and deploy highly scalable portals. All offerings share a common framework (the portal server) plus additional products and services. The portal server provides common services such as application connectivity, integration, administration and presentation that are required across portal environments.



Application platform considerations

The Lotus Workplace platform offers you a choice of application development models and corresponding server run time. The Workplace Platform supports both J2EE technology-based applications, and Notes and Domino platform-based applications. There are pros and cons to every server environment and application development model that will impact the success of an application. The following sections will cover items like the application deployment topology, the skills available in the organization, and the technical requirements of the application's data and scalability to help assess the best environment for an application. These will help you determine the best design strategy for new applications. In many cases, the new application will combine the strengths of J2EE and Domino for the maximum return on investment (ROI) of the application.

It is also important to note that data requirements are formulated around the different benefits between a relational datastore, like DB2 Universal Database™, and a hierarchical store, like Lotus Domino. WebSphere Application Server and WebSphere Portal are not data management products but do require a RDB to store their configurations and contents.

In some of the cases described in the following sections, WebSphere Portal has been noted separately from WebSphere Application Server. In these cases, the functionality of WebSphere Portal may impact the decision, otherwise the functionality of WebSphere Application Server is surfaced through WebSphere Portal and the same considerations apply.

To best use this section, have an application (either current or planned) in mind and answer each question as it applies to the application.



Application deployment topology

Disconnected (offline use)

Does the application need to be used from a computer that is disconnected from the network with full application fidelity?

Application fidelity means that the application functions the same, whether online or offline, including creating, editing or deleting data. This is one of the unique features of Lotus Domino. Only through Domino replication can an application be taken offline and used with full fidelity and access to data from a Web browser or Lotus Notes client. When the user reconnects, the work he or she did offline is fully synchronized with the server.

Lotus Notes client access

Does the application require a rich client?

Lotus Notes provides a desktop environment for maximum user proficiency in collaborative applications, such as e-mail, calendar and scheduling, personal information management, and custom collaborative applications. Lotus Notes provides a graphical window user interface and an extensible client framework for running collaborative and document-centric applications. The following list includes Lotus Notes client features that differentiate it from a Web browser:

- *Richly formatted content editing (tables, spell check, inline graphics, full font support, find/replace, and so on)*
- *Microsoft® Office integration for attaching, storing and distributing files*
- *Offline and disconnected e-mail, calendaring, PIM, and applications with virtually full functional fidelity while offline and synchronization when reconnected*
- *Data encryption with security features, wire encryption and digital signatures not available in a browser environment*



WebSphere Portal ships with portlets allowing access to Domino applications and mail. An integrated feature of the portlets allows the user to launch the Notes client from within the portal when viewing Domino data. This allows the portal's capability of application and data aggregation to work while still proving the end user with the ability to manage Domino data from within the Notes client for its rich feature support.

Application purpose and use

Application content

Will the application include rich-format content editing, creating and rendering?

Many user-facing application require creating, editing or deleting richly formatted text or the ability to display, store and process Microsoft Office documents. Lotus Domino and Notes have built-in capabilities for handling this type of content through the built-in rich-text editor and Microsoft Office integration in the Lotus Notes client. Domino can also serve rich-text data to Web browsers that is stored in the Domino datastore.

WebSphere Portal also has rich-text editing capabilities through the use of portlets that will store the content in the portal's content management repository and through Microsoft Office integration using portlets that will render and manage Office documents. In either case – Domino or WebSphere Portal – it is required that a licensed version of Microsoft Office be installed on the client machine for Microsoft Office integration.

Planned for a future version of WebSphere Portal are portlets that will allow for the creation and management of spreadsheets, documents and presentation files from within the portal, and without the need for any productivity suite to be installed on a client machine.



Transaction processing

Is the application performing transactional tasks?

A transactional application is one in which a series of steps in the application is monitored so that the process can be completely reversed if necessary. A classic example is a banking system in which an account credit must be reversed if the preceding debit step isn't complete. Transaction processing is a key benefit of WebSphere software-based applications. Not only is WebSphere Application Server able to handle such transactions, the underlying datastore must be able to support those features as well, such as an RDB. Domino itself does not support transaction-oriented applications but could be integrated providing the collaborative services for the human intervention necessary for resolving exceptions that occur in business transactions.

User-centric forms-driven work flow

Does your application need to route electronic forms to users in a series of predefined steps to complete a work-flow process?

As an example: an employee self-serve help-desk application in which the employee fills out a form that is routed by e-mail to the appropriate help-desk professional when submitted. Upon closure of the problem, the completed form is routed back to the originator for sign-off. After sign-off, the form is logged into a help-desk tracking database.

These types of solutions are relatively easy to build and deploy on Lotus Domino due to the built-in e-mail APIs and flexible roles-based identities and access control.

A value-added solution is to integrate Lotus Domino with WebSphere Application Server and WebSphere MQ using Lotus Domino for the user-centric forms-based work flow that occurs in support of transaction-processing work flows that are most suitable to be managed these WebSphere applications. Such end-to-end work-flow solutions that marry systems transactions with human processes is a capability that few single vendors can offer.



For content-management-approval work flow, Lotus Domino is a valid option due to its flexible work-flow capabilities as well as WebSphere Portal Content Publisher. The WebSphere Portal Content Publisher has approval work-flow routing built in and can leverage Lotus Workflow or IBM Content Manager work flow.

Access control

Do users need to delegate their actions?

Some applications require a feature that grants one user the access-control rights of another user. This is most frequently used to enable one user to act on behalf of another, while tracking which user did what. A classic example is a manager delegating responsibilities to an assistant. The assistant should not receive equal access rights to *be* the manager, but should get access rights to *act on behalf of* the manager. The difference is that any create/read/update/delete actions by the assistant are noted separately from those actions by the manager for auditing purposes. This type of functionality is tied to the support of the underlying datastore using role-based identities and access to individual elements out of a database. Lotus Domino supports role-based identities and a finer-grained access-control model than a RDB, making it easier to implement these types of applications on Domino.

Does a user's access control need to be limited to a subset of records in a table or fields on a form?

For example, a manager is granted access to the “employee salary” table, but he or she can only see the records of the employees that he or she manages. Likewise, a user filling out a form might see only certain fields, whereas a manager using the same form has an expanded instance with additional fields shown.

Lotus Domino has a unique security model in which access control can be applied to very granular subsets of the application and the data. Access control can also be altered programmatically on the fly, adjusting who sees what based upon a series of dynamic conditions.



Does the application need dynamic role-based user identities?

Most applications make use of groups in the corporate user directory. However, there are many cases in which a group is application-specific. Storing application-specific groups in a corporate directory is unwise because of management implications, so a better solution is to incorporate the user roles into the application, storing the roles as part of the application data and writing logic to determine the role of the current user.

A good example of an application that requires roles is an approval process. For example, in a human resources (HR) hiring application, the people who need to approve a salary offer are the HR manager, the finance manager and the hiring manager. Each of them is a member of different groups in the directory, but salary-approval rights cannot be granted to these groups because they contain too many other people. A new group called *Salary Approval* cannot be created because the hiring manager member changes with every hiring instance. So the solution is to create an application-specific role that users move in and out of. Role membership is changed on the fly through application logic, and administration is done at the application level instead of at the directory (systems) level to simplify application administration overhead.

Although it is possible to build a role-based user-identity model on any application platform, this is something that is a simple and common practice in Domino applications. This capability also provides another integration point for Lotus Domino with WebSphere Application Server or WebSphere Portal for any J2EE application that needs dynamic role association.



Application life cycle

Strategic versus tactical

How strategic is the application? Or, is it a tactical or point solution?

This question asks you to evaluate the application purpose and life cycle. If the application is expected to have a useful life cycle of five years or more, or if the application is crucial to supporting the core business, then it is *strategic*. *Tactical* or *point solutions* support a relatively short-term project or business initiative. These solutions may also include applications that are used infrequently. For example, an HR employee performance-review application used only once a year is a tactical or point solution because it is infrequently used and is only indirectly tied to core business operations.

RAD environments come as a result of ever-increasing integration of a platform and its components. Most RAD environments enable ease of use over complete flexibility and future extension of the application being built. Due to the RAD nature of Lotus Domino Designer, most Domino applications are short-term or tactical solutions because developers can build and deploy applications in a relatively short period of time. If your organization has such applications deployed on Lotus Domino, it is not advisable to re-architect them on J2EE because it is unlikely that you would gain sufficient return to offset the development expense.

Strategic applications are architected to deliver maximum returns over the duration of the application life cycle, which includes unforeseen future extension and integration of the application. Careful assessment of the application shelf life and consideration of the factors described in this paper should help determine the appropriate platforms for strategic solutions.



Time to value

Does your application need to be designed or enhanced, developed, tested, and deployed within six months?

For example, following an internal audit, a department has three months in which to implement a change-management system to show compliance with ISO 9000. There isn't time for a lengthy development cycle, but without the solution, the department won't reach compliance. As mentioned earlier, one of the key benefits of Lotus Domino is its rapid application development environment. It can be possible to develop, test and deploy an application in weeks as opposed to months on other application server platforms. "Mileage" may vary, of course, depending upon the complexity of the application. Many applications well-suited for Lotus Domino are relatively simple to build, especially when the developer can use one of the standard system templates that ship with Lotus Domino.

Environmental factors

IT enterprise infrastructure standards

Is the application subject to IT strategic policy or enterprise governance?

Many IT organizations are making strategic decisions regarding their application infrastructure to simplify systems management and to save money through server consolidation. Most companies base their strategy on the J2EE application architecture, LDAP for directory services, and RDBs for data storage. If your application architecture is governed by such IT enterprise strategy, then it probably belongs on the WebSphere platform, keeping in mind that Lotus Domino may be needed for features not yet available in the J2EE architecture. Another fact to remember is that you can configure the Domino Directory to be an LDAP directory, so your existing Domino Directory will fit within the IT policy, IT infrastructure and support resources.



Are full IT resources available?

There is always overhead associated with application development, deployment and management, including IT infrastructure (database, application server, directory and security) and human resources for system administration, database administration, directory services or security architecture.

Consider an application that needs to be deployed to a field site in a remote, underdeveloped country without adequate local IT support or wide area network (WAN) bandwidth. Along with such infrastructure constraints, budget constraints are another contributing factor. For example, a business unit has identified the need for a solution, but is unable to secure adequate funding for IT resources. These are just two of many cases in which there is a justifiable business need for the application in absence of available IT infrastructure and human resources.

Lotus Domino would be the most-suitable solution for these problems due to it being a fully integrated, stand-alone application server. Conversely, the WebSphere application platform relies on various other infrastructure elements being available in the environment such as an LDAP directory and a relational datastore that are installed and managed separately from the application server, requiring more IT resources to support the platform.

Application development skills

Do you have obtainable or available Lotus Notes or Java development and administration skills?

Many applications need developers to create and maintain them, and system administrators to manage them. Oftentimes, application architecture is determined by the available skills in the organization. Making an assessment of which resources are currently available or obtainable through hiring or contracting is an important factor in determining which platform you choose. Although a good starting point is the type of skills that you have available in-house, it is also important to keep skills up to date, which also includes



cross-training of skills. Domino developers would benefit from gaining an understanding of the value and role that WebSphere can play, as well as WebSphere developers understanding the collaborative power that Domino brings to an application.

Scalability requirements

Number of users

Will the number of concurrent Web-browser users exceed 5000?

There are several factors that determine the scalability requirements of a Web application, one of which is the number of concurrent, active users of the application. As a general rule, if the number of concurrent active users exceeds 5000 and they use Web browsers, then the application is probably best suited for the WebSphere Application Server.

However, if the application is deployed to Lotus Notes clients, then choose Lotus Domino. Lotus Notes uses a client/server architecture in which much of the processing is done client-side to permit greater server scalability.

Infrastructure configuration for scalability

Does your application need to scale horizontally or vertically, or both?

Deploying the appropriate hardware configuration is key to meeting application scalability and availability requirements. *Horizontal scalability* refers to distributing an application across several physical servers or several server instances in a “blade” configuration. *Vertical scalability* refers to taking advantage of multiple processor configurations through upgrading to a more-scalable hardware and operating system (HW/OS) platform. Be sure to consider both the scalability requirements for today and for the future. WebSphere Application Server, WebSphere Portal and Lotus Domino all take advantage of vertical scalability because they support a broad range of HW/OS platforms from Windows servers at the low end to the IBM @server zSeries® server at the high end, and can scale across multiple processor configurations of those systems.



The key factor for an application's ability to scale horizontally is the amount of state-based information maintained for the user in the application. It is much easier for near-stateless applications to scale horizontally than for applications with large amounts of per user/per session data. By default, Domino keeps a large amount of user and session information open during the use of an application, whereas it is up to the application developers of a WebSphere platform-based application to manage and track the information that they want. Thus, WebSphere platform-based applications written to not use much user and session information can leverage horizontal scalability better than a Domino application.

Data requirements

Data model

Does the application need a structured or unstructured data model?

There are several types of data storage; however, relational and hierarchical are the two that apply to this paper.

Relational storage maintains multidimensional links between “related” records. Hierarchical storage uses a comparatively flat parent/child relationship model. Typically, relational storage is better suited for structured data where many-to-many relationships exist between records, such as the linkage between customer records, invoices, purchase orders and vendor records.

Hierarchical storage is better suited for unstructured data, such as documents, lists, Lotus Freelance Graphics® or PowerPoint presentations, and other forms of “free form” information. Another characteristic of hierarchical data is the ability to change the data structure, such as adding or deleting fields at will without rebuilding the database. Other characteristics unique to unstructured storage are:

- *Allows a field to have multiple values*
- *Allows simultaneous editing of records by different applications or clients (no record locking)*



- *Works best for relatively simple (low-complexity) searches and searches of document contents*
- *Works best with data that lends itself to being displayed as a categorized list or in a single table*

Many applications require both relational and hierarchical data storage, so this is perhaps the strongest case for building applications that integrate the features of Lotus Domino and WebSphere Application Server. Lotus Domino has a built-in unstructured data store and WebSphere Application Server and WebSphere Portal have robust connectivity to RDB systems. It's also important to remember that hierarchical Domino data can be surfaced through WebSphere Portal through the supplied portlets, and to WebSphere applications through the IBM Lotus Domino Toolkit for WebSphere Studio and the Domino Objects for Java.

The portal document management feature also allows for users to create unstructured data within the portal, which is accessible through the provided APIs.

Does the application need direct access to a system of record?

A system of record is a relational database feature that enforces existence of one, and only one, instance of a record. This is a requirement of many applications that run in regulated industries, such as insurance, healthcare and pharmaceutical. Although this is a feature of the datastore itself and not the application platform, the unique Domino characteristic of containing its own datastore and replication model is counter to a system of record requirements. Applications that require a system of record generally rely on a relational datastore and are more common on the WebSphere platform.



Data volume

Will the data volume exceed 2 GB?

The amount of data storage needed for an application is an important factor in determining its architecture. Other data-related factors also apply, such as the degree to which users interact with the data with create/read/update/delete (CRUD) activities and search requirements. With Domino Release 5 and later, there is no longer a hard limit on the size of a Domino database, but a relational database and WebSphere applications are better suited for managing high data volumes and high CRUD activities.

However, a common exception is the need for storing unstructured data, such as documents, and for supporting rich text or highly formatted content, all of which is a benefit of Lotus Domino. The ideal solution in this case is to build an integrated application. If it is necessary to store the application data in a single database, then the LEI can synchronize the Domino database with the RDB.

Also, Domino 7 will support storage of information natively in DB2 along with the standard network file system (NSF). The use of DB2 will provide for increased scalability and flexibility of Domino data and its integration into other systems.

Data distribution

Does the application data (exclusive of the application logic) need to be distributed to servers across a WAN or the Internet?

In many cases, an application is architected for execution across distributed servers, but the *data* remains centralized, usually in the interest of protecting data integrity. However, there are situations in which the data needs to be distributed, especially when applications are deployed to regions with low-bandwidth due to a spotty or expensive (or both) networking infrastructure. Sometimes, data distribution is achieved by distributed data caching, but this often puts data integrity at risk and can be very costly and complex to



manage. Lotus Domino and the LEI provide an ideal solution for distributing all types of data across a network because they ensure data integrity, security and synchronization of all instances of the data set. The distribution of both unstructured and relational data using LEI provides an example of integrating the strengths of both Lotus Domino and WebSphere Application Server to meet application requirements.

Leveraging your investment in Domino

One of the greatest benefits of developing applications for the Domino platform is that applications written for Notes Version 1 can be run unaltered on a Domino 6 server. That means your company is still realizing value from an investment it made some 13-plus years ago and those applications can now be accessed by Web browsers, Java APIs and Web services. As Domino moves forward and becomes one of the server platforms for Lotus Workplace, it is important that IBM continues to protect our customers' investments.

One conclusion of the previous section "Application platform considerations" on page 11 that may have been reached was that the application requires the rich client experience provided by the Lotus Notes client. IBM plans to continue to support and enhance Domino as an application server for Lotus Workplace. This section provides information on leveraging your Domino investment from applications running on WebSphere Application Server or WebSphere Portal.

Domino integration with WebSphere

Within IBM, the Domino and WebSphere Application Server teams have coordinated efforts to develop and to deliver key integration points that customers have requested. WebSphere Portal benefits from the same integration points because it is based on WebSphere Application Server. Each of these elements stem from the need for an excellent user experience, and easier administration and overall infrastructure integration. Whereas Java and J2EE standards make it possible to integrate disparate applications at a fairly broad level, Lotus and



WebSphere have developed powerful integration points that are only available between Domino and WebSphere Application Server. For example:

- *For customers looking to standardize on a single user directory, Lotus Domino can serve the Domino Directory to other applications using LDAP. This protocol is quickly becoming the de facto standard for directory serving and is also used by WebSphere Application Server for authenticating users. Because Lotus Domino can serve as an LDAP server, WebSphere can use this directory protocol, allowing customers to manage user identities for Lotus Domino and for WebSphere in one location, easing administration skills needed in customer environments.*
- *In cases where WebSphere Application Server and Lotus Domino use the same LDAP directory (not limited to Domino LDAP capabilities alone), customers can configure Single Sign-On (SSO) between Lotus Domino and WebSphere. SSO allows a browser user to enter his or her authentication information once and to have that information stored in a secure context that both WebSphere and Lotus Domino share. With SSO enabled, users navigating between WebSphere and Domino applications do not need to authenticate multiple times as they would without SSO enabled.*
- *In addition, the Domino Objects for Java (the Domino Java API) allow for the secure processing of data in Lotus Domino by allowing the use of the SSO token to create a session in Lotus Domino. Programmatically, a developer can use the SSO token to create a secure session with Lotus Domino as the current Web-application user from WebSphere Application Server. This eliminates the need for an application to know the specific user name and password when accessing Lotus Domino, and also minimizes a security risk of users' passwords being handled by any given application.*



Leveraging your Domino data

If the choice was to have WebSphere Application Server or WebSphere Portal handle the presentation and application logic, customers can leverage the data that they have in Domino applications. Depending on the J2EE applications needs, this can be done using a RAD development tool and to more granular control using XML and Java.

WebSphere Portal tools

WebSphere Portal provides a number of prepackaged Lotus Notes portlets that allow you to integrate existing Domino applications into the WebSphere Portal environment. These portlets allow those with basic WebSphere Portal administration skills and little-to-no programming skills to configure the portlet quickly and effectively. These portlets can display data from the Mail, Calendar, To Do as well as views from any Domino application. By providing these collaborative portlets, Domino customers can easily extend the most common Domino applications to the WebSphere Portal environment.

By adding Domino capabilities to the existing WebSphere Portal Application Integrator framework, the Portlet Builder for Domino enables line of business (LOB) users to build portlets quickly and easily based on IBM Lotus Domino. This wizard-like portlet eliminates the need for programmers to write low-level functions and allows developers to easily bring in key Domino applications and data into WebSphere Portal. This portlet builder also includes IBM Lotus Sametime® Presence Awareness and Click2Action capabilities for sharing information between portlets, providing immediate productivity gains for portal users.

WebSphere Business Integration Adapter for Domino

WebSphere Business Integration applications provide the ability to support complex business models that rely on integrating data from disparate systems. The ability to connect to Domino is now provided by the WebSphere Business Integration Adapter for Domino. The Domino adapter supports bidirectional, real-time integration between Lotus Domino and one of the many other enterprise systems supported by the adapter suite including systems like Siebel and PeopleSoft.



Domino features enable applications to integrate with J2EE

Domino also ships with features that facilitate the integration of Domino data with J2EE applications. You can reap many application benefits by combining Domino and J2EE applications, most importantly, the addition of human interaction into scalable and transactional applications built on J2EE. When integrating Lotus Domino with a J2EE-based application, there are two key ways to access Lotus Domino:

- *Domino Objects for Java*
- *Domino 6 custom JSP tags*

The Domino Objects for Java are essentially the Domino Java API. These objects allow for object-based access into the data store for the key database procedures, the CRUD activities. This API allows access not only to Domino data but also to some key services, such as registering users, running agents, sending mail, and others.

The Domino 6 custom JSP tags are for use only from within a JSP. These tags are XML tags embedded in a JSP that is providing data access, data input and process control. JSP tags abstract the Domino Objects for Java and provide a quick development turnaround for building a J2EE application that uses Domino data and services.

Lotus Domino Toolkit for WebSphere Studio

Starting with Domino 6.0.2, the Lotus Domino Toolkit for WebSphere Studio provides developers with an intuitive tool to implement the Domino custom JSP tags into their WebSphere applications. The toolkit is a set of plug-ins that enhance the WebSphere Studio Application Developer user interface and it enables Domino developers to build J2EE applications without having to know the complexity of Java.



A Domino view is added to the Studio UI, which allows you to work easily with common Domino objects like forms, fields, views and agents. These objects can then be dragged and dropped onto the JSP being developed and the equivalent Domino custom JSP tag code is inserted. Another benefit of the toolkit is that it hides the complexity of connecting your J2EE application to Domino using Java.

XML

Lotus Domino also has extensive XML capabilities that can be used from a J2EE application. A J2EE application can use the XML interface to Domino to access the unstructured Domino documents without knowing specifics about the data stored in Lotus Domino. Then, the J2EE application can process the resulting XML and transform it to the specific format needed by the application. When the application is ready to update Lotus Domino, the same XML interface can be reused to update the database through XML, thus providing for a round-trip of data in and out of Lotus Domino.

Web services

Web services is a collection of emerging standards that simplifies application integration by providing a standardized access protocol known as *Simple Object Access Protocol (SOAP)*. Developers can express the application interface using XML. Lotus Domino can host Web services that expose Domino data and functionality using LotusScript. Using a combination of a J2EE server like WebSphere, appropriate SOAP classes and the Domino Java Objects, developers can expose desired portions of their current Domino applications as Web services. To do so requires some Java development skills and knowledge of the Domino object model. IBM WebSphere Application Server ships with the needed SOAP classes, and the WebSphere Studio development environment has wizards for creating, consuming, managing and deploying Web services. This helps improve developer productivity and lets developers focus on exposing the essential parts of a Domino application as Web services.



Domino on DB2

Domino 7 will support DB2 as an alternative data store to the NSF. This will bring together the strength of the collaborative application capabilities of Domino with the DB2 relational database architecture, scalability and performance. By providing access to Domino data through a relational database interface, leveraging your investment in Domino data will be easier than ever.

Projecting Domino applications through portlets

Portlets are the component model for the Lotus Workplace application assembly. Domino applications can be developed for access through a portlet. From the developer's perspective, this is similar to enabling a Notes client-based application to be accessed from a Web browser. Because the Lotus Workplace platform is built on the WebSphere Portal framework, the desire to surface custom Domino application will be great.

Domino Web applications can have various design elements that do not map well to portlets or do not adhere to the portal framework. Some of these include frame and frames tags, JavaScript namespace issues and URL redirects targeted at Domino and not the Portal server. Some of these issues can be addressed by using the iFrame or Web clipping portlets.

IBM plans to improve the ability of projecting Domino Web applications in the portal environment by providing enhancements to both Domino itself and improved portlets target for use specifically with Domino. In early 2004, a reverse proxy portlet is scheduled to be released that will allow URL redirects to pass only through the portal server. This portlet will understand common HTML constructs emitted by the Domino server and map those to constructs that adhere to the portal architecture. Additional rules for dealing with unusual HTML constructs can be added.



Because developing Domino applications for the portal environment is very similar to developing Domino Web applications, enhancements to tools used to develop and serve Web applications are also planned. These include enhancements to Domino Designer to allow Domino application development suitable for the portal framework as well as enhancements to the Domino Web engine. See Table 2 application development objectives and options.

Application development objective	Available options
Access applications through Notes client	<ul style="list-style-type: none"> • Domino Designer
Access applications through Web browser	<ul style="list-style-type: none"> • Use Domino Designer to develop applications for Domino Web server • Use Domino Designer along with one of the options mentioned in the J2EE section on page 24
Access applications through WebSphere Portal	<ul style="list-style-type: none"> • Use Lotus portlets • Develop portlet using Portlet Builder for Domino • Use upcoming portlet to surface Domino Web applications in WebSphere Portal
Access application data	<ul style="list-style-type: none"> • Domino XML support • Domino Objects for Java • WebSphere Business Integration adapter for Domino
Access applications through Web services	<ul style="list-style-type: none"> • Use Domino to host Web services written in LotusScript or Java • Use WebSphere to host the Web Service and use one of the options mentioned in the J2EE section on page 28
Move data to relational data store	<ul style="list-style-type: none"> • Use upcoming feature in Domino 7, which will allow DB2 to be an alternate data store to the NFS • Use Lotus Enterprise Integrator for Domino to transfer data to a relational data store

Table 1. Application development objectives and available options.

The future of rapid application development

With the introduction of Lotus Workplace and its inherent support for e-business on demand, the ability to rapidly develop applications will continue to be the focal point of the application development tooling strategy. With Lotus Workplace supporting WebSphere Portal, WebSphere Application Server and Lotus Domino as the application servers, each of these underlying platforms needs to provide RAD tools to support the Lotus Workplace.



The proven RAD capabilities of Domino Designer make it well-placed to support Domino in Lotus Workplace. Domino Designer is the application development tool for building Lotus Notes and Domino applications. As stated previously in this paper, the future of Domino Designer is one of continued enhancement, especially focusing on features that enable the development of integrated applications for use in Lotus Workplace. These plans include enhancements for Web services, features to take advantage of DB2 as a data store and the ability to customize applications for access through the portal environment. There are no plans to port Domino Designer to WebSphere Studio or Eclipse1 – it will remain a dedicated RAD tool for Lotus Notes and Domino.

With the IBM focus on the J2EE architectures, the need to provide application tools that appeal to the application development communities beyond Java technology-savvy developers is clear. This requires the creation of a RAD tool that empowers Web application developers to build and connect components, data, and application business logic, while insulating them from the complexities of the J2EE architecture.

In 2002, IBM announced a project to develop such a tool. This project benefits by understanding the needs of RAD tool users, along with the expertise gained in developing Domino Designer. This future RAD tool will provide an easy, visual way to develop rich Web applications that leverage the J2EE environment. Lastly, this future RAD tool will be a comprehensive development environment, covering all phases of application development: coding, testing and deployment. This future tool from IBM is based on the Eclipse open-source framework and leverages the tools available in the IBM WebSphere Studio family for many key functions, such as page design, script editing, debugging, bean creation, and so on.

Among Domino Designer, the J2EE RAD tool and the various tools for portlet development, developers will have a suite of RAD tools that will enable them to build the modular components that make up Lotus Workplace. But how does one “glue” all these modular components together? For the Lotus Workplace platform, Lotus is planning to provide intuitive tooling that will enable developers to bring together the collaborative components for their applications.



Conclusions

Lotus Workplace offerings are starting to become available in the market and you can begin investigating how to leverage them in your organization. As with any IT infrastructure, it is important to make the best decisions to maximize the investment made and leverage that investment wherever possible. To that end, there are elements that are applicable to any platform that you choose.

Technologies such as LDAP are becoming the standard for corporate directories. If you're using Domino today, Domino can serve as your corporate LDAP directory and also as the directory used by WebSphere Application Server and WebSphere Portal. If you haven't standardized on a LDAP directory it is recommended that you consider doing so.

Web services and XML are also an ever-important part to a forward-looking IT infrastructure. WebSphere Application Server can be your Web service application-hosting platform and Domino can integrate to provide data and content to those Java technology-based Web services. WebSphere Portal also has Web services capabilities through Web services for remote portlets (WSRP), which allows a portlet to act as a Web service. Therefore, any application served through the portal can act as a Web service to another application. XML is the language of choice for Web service data markup and Lotus Domino, WebSphere Application Server and WebSphere Portal all have robust XML functionality.

Specifically there are different advantages to using either Lotus Domino, WebSphere Application Server or WebSphere Portal for your applications. From the "Application platform considerations" section on page 11, keep in mind that:

- *The strengths of Domino lie in collaborative applications providing a hierarchical datastore, rich content handling and a robust security model*
- *WebSphere Application Server is ideal for transactional applications, highly leveraging a relational datastore and providing enterprise-level J2EE application hosting*



- *WebSphere Portal excels at application and data aggregation, providing a rich framework for Web-based applications*

When looking to leverage your existing Domino investment, there are many options. Simple configurations that enable Domino to interoperate with WebSphere Application Server and WebSphere Portal such as a common LDAP directory and SSO are a great start. From there, application integration of using Domino data within a WebSphere platform-based application can help an organization standardize on an application platform. When performing application integration, it is possible to leverage Domino using one of the many options: Lotus Domino Toolkit for WebSphere Studio (a portlet builder to generate customized Domino portlets to your application), the WebSphere Business Integrator connector, Domino Objects for Java, or even the out-of-the-box Domino view and mail portlets, just to name a few. To begin the integration journey, Domino 6 provides numerous features to make integration easier. If you haven't upgraded to Domino 6, look into it.

However, there are also times that the best option isn't integration, it's possibly rebuilding an application onto a more-appropriate platform. This could be a J2EE application that was forced onto a platform that would have been better suited to be a Domino application. The converse is also true, that a Domino application could better serve its users as a J2EE or portal application because of a high need of relational data integration. Using the "Application platform considerations" section on page 11 will help determine if integration is a possible route or if rebuilding the application would provide for better utilization of infrastructure resources.

Finally, no matter what route you take, it is recommended that you reorganize development teams so that Domino and WebSphere developers work together to build applications. Domino developers bring unique skills and experience in building user-centric collaborative applications. WebSphere developers have experience in systems programming that can be valuable to extending Domino



applications to larger-scale deployments beyond the enterprise. Through organizational alignment, these teams can tap each other's strengths to integrate collaboration to more areas of the business and beyond the corporate boundaries. This also applies to any customer using a J2EE container other than WebSphere.

Related publications

IBM continues to release additional documentation, like white papers and Redbooks, to keep you updated on new technological enhancements. See the following Redbooks for further reading related to application development and integration between Lotus Domino, WebSphere Application Server and WebSphere Portal.

- *SG24-5955-01: Domino and WebSphere Together Second Edition*
- *SG24-6854-00: Domino Designer 6: A Developer's Handbook*
- *SG24-7004-00: Portalizing Domino Applications for WebSphere Portal*
- *SG24-6998-00: Integrating Domino 6 and WebSphere V5 on the IBM eServer iSeries Server*
- *SG24-6835-00: Lotus Domino 6 for Linux*
- *SG24-7021-00: IBM Lotus Domino 6.5 for Linux on zSeries Implementation*

Redbooks are available from **ibm.com/redbooks**



© Copyright IBM Corporation 2003

IBM Systems Group
5600 Cottle Road
San Jose, CA 95136
U.S.A.

10/03
All rights reserved

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

IBM, the IBM logo, the e-business logo, the eServer logo, DB2, DB2 Universal Database, Domino, e-business on demand, Enterprise Integrator, Freelance Graphics, Lotus, Lotus Notes, Sametime, WebSphere and zSeries are trademarks of International Business Machines Corporation in the United States or other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft

Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

G224-7192-00