teamstudio®
Strengthening the platforms business is built on.

IBM WebSphere® for Lotus® Notes® and Domino® Professionals

White Paper by Teamstudio, Inc.

Executive Summary

Lotus Notes and Domino developers and project managers who want to understand and prepare for the future direction of their development practices will do well to begin considering Java- and J2EE-based technologies.

Through events such as the annual IBM Lotusphere® conference, the message is becoming increasingly clear that these development platforms will be important both for integration with current Notes/Domino applications and future enterprise application direction. As a result, IBM's WebSphere is becoming a technology focus for organizations with a large Notes/Domino investment. The reasons for this are as follows:

- WebSphere is developed and maintained by IBM, the same company that maintains Lotus Notes and Domino
- IBM has announced integration with Lotus Notes/Domino as a clear goal of WebSphere's future direction, in addition to protecting the investment so many organizations have already made in Notes/Domino
- IBM's desire to keep current Notes/Domino customers in the IBM development "family" will ensure they respect the needs of the installed Notes/Domino base, as opposed to companies who are providing competing technologies with the end goal of "killing" Lotus Notes

Many organizations are now struggling with the decision of whether or not to replace Notes/Domino, and if they decide to do it, how soon to begin the process. Notes/Domino shops are drawing a myriad of conflicting inferences about IBM's future plans with Lotus Notes. This report doesn't seek to validate any rumors about the future of Notes. Rather, this paper seeks to introduce the basics of IBM WebSphere to managers and technologists coming from a Notes/Domino background.

First, we will undertake a brief overview of the Lotus Notes architecture; then we'll explore the basics of IBM's WebSphere infrastructure platform. Finally, we'll take a look at the current environment for integrating Lotus Notes and WebSphere applications.

teamstudio.com

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

I. A Quick Refresher on Lotus Notes and Domino

Lotus Notes/Domino is a unique and very well-integrated combination of many components essential to complex application development, including:

- A rich and proprietary document-based data store
- A powerful security model that supports security at the server, application, user and document levels
- A feature-rich messaging layer, including calendaring and scheduling
- A fast and relatively easy-to-use application development layer including native support for industry standard technologies such as Java and JavaScript
- A revolutionary distributed data replication model

The Lotus Notes Architecture

Lotus Notes uses a three-tier software architecture (see Figure 1 below) consisting of:

- The Client (Lotus Notes Client, Designer, Administrator or Web browser for Web-enabled Domino applications or server (Domino server, server add-in tasks) layer
- The NOS (Notes Object Services) layer
- The Database or File layer (local Notes databases, shared Notes databases, other data source files)

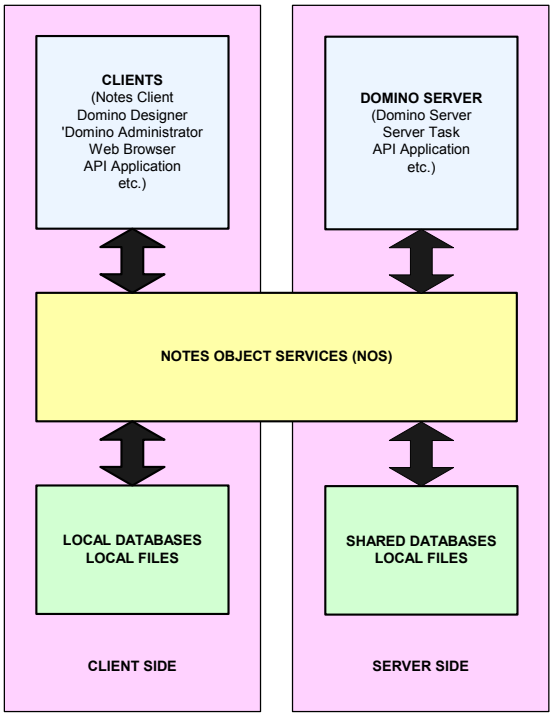

Figure 1: The Lotus Notes/Domino three-tier architecture.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

Notes Object Services (NOS)

The Notes Object Services (NOS) layer provides a set of functions and libraries that allows users (or Domino servers or API applications) to create and access information in databases and files, compile and interpret formulas and scripts, and deal with operating system services in a consistent, portable way. Besides application access, NOS provide the functionality of the Notes security model and contains the logic that allows many users or programs to access the same data simultaneously.

NOS is the key to allowing Notes applications to function in virtually the same way across many different hardware and software environments. It supports several operating systems, and provides support for many different language compilers and interpreters (such as LotusScript, the *@Function* language, Java and JavaScript). In addition, NOS provides globalization support. Presumably, as IBM begins to add DB2 back-end support for Lotus Notes/Domino, this will come through updating Notes Object Services to provide common database structure and functionality, regardless of whether the data store is a native Notes .nsf file or a relationship DB2 database.

The Notes Storage Facility (NSF)

The native Lotus Notes data store is a proprietary document-based file format, either stored locally and accessed directly by a Notes client, or stored on a Domino server and accessed by a Notes client or standard Web browser (when Domino HTTP services are enabled). The Notes Storage Facility (NSF) file was created specifically as a rich data store that supports multiple compound document types consistent with complex, multimedia-rich applications.

Each bit of data stored within a Lotus Notes database is called a "note." A note is a simple data structure that can store different types of data in many different formats, depending on the class of the note. Significant classes of notes found in Notes databases include administration (security or access control information, or replication information), design elements (such as forms which can be used as templates to create new documents or records, and views which can store and sort lists of documents) and of course data or documents created by users of the database.

Because the NSF file was created and has evolved to directly support the groupware functionality provided by Lotus Notes, it is an extremely powerful and flexible receptacle that can store many complex bits of data that can easily be created and accessed. This power and flexibility supports the rapid application development capabilities of Lotus Notes.

Developing in Lotus Notes

Lotus Notes also includes a full-feature Integrated Development Environment (IDE). Developers used to more traditional IDEs may find the Notes IDE to be fairly idiosyncratic, but those familiar with this interface find it easy to build fairly complex application interfaces, quickly and reliably.

One advantage of the ease of use of the Notes IDE is the ability of Notes developers to rapidly prototype applications to get them into the hands of end users, who can then review requirements and request updates while "touching" an actual working application. Those from a more traditional software development environment might feel uncomfortable rolling out an application that's less

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

than complete, but most Notes developers take full advantage of the responsiveness that such a quick and facile IDE allows, and their customers quickly get used to very fast turnaround times for their change requests.

As mentioned above, Notes includes support for such widely accepted languages as Java and JavaScript. Notes developers can also take advantage of two programming languages written exclusively for Lotus Notes: the *@Functional for Formula* language, and LotusScript. These scripting languages include native support for basic Lotus Notes functionality, such as accessing data at the field, document and view level. While developers can do much of their coding in Java or JavaScript, there are areas within an application where one *must* use the native Notes programming languages, such as a *view selection* formula or a *view column* definition.

Pros and Cons

These are just a few highlights of the Lotus Notes architecture but they allow us to outline some of the pros and cons of Lotus Notes as a development platform:

Advantages

- Highly integrated groupware functionalities (e.g., security, messaging, replication, rich data support, etc.)
- Rapid application development; support of prototyping (develop as you deploy)
- Document based database: excellent for discussion, easy workflow, etc.

Disadvantages

- Proprietary format and development standards
- Traditional software development practices and procedures cannot be easily extended to Domino development
- Flat, denormalized database makes it difficult to integrate complex, rules-based content with traditional, relational data stores (Note: This shortcoming will be addressed in future releases of Lotus Notes with the provision of DB2 as a data store.)

II. WebSphere Basics

IBM refers to WebSphere as Internet infrastructure software or middleware. It's an integrated environment including several server solutions, an IDE and a host of related products that enable companies to develop, deploy and integrate complex, very large-scale, Web-enabled e-business applications, such as business-to-business e-commerce applications. WebSphere supports a full range of business applications, from simple Web page publishing and hosting to mammoth enterprise-scale transaction processing applications. However, given its power and complexity—and its cost—most organizations will use WebSphere for high-end application requirements.

The core of the WebSphere solution consists of the WebSphere Application Server family, and the WebSphere Studio family.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

The WebSphere Application Server Family
The WebSphere Application Server family consists of three separate server solutions that provide specialized, optimized configurations for different needs. However, each server configuration is essentially powered by the same Java engine, thereby allowing applications to be easily ported to other WebSphere server editions as an organization's needs change.

These three servers also provide a range of flexible licensing options, such as per-processor licensing for intranet, extranet or Internet applications, or per user licensing for employee intranet applications only.

WebSphere Application Server (WAS)
WebSphere Application Server (WAS) is a full-featured, Java-based application server that provides complete J2EE 1.3 and Enterprise JavaBeans (EJB) 2.0 support. WAS also provides a proprietary, high-performance API called the Java Message Service (JMS), based on IBM's WebSphere MQ technology.

WebSphere supports the development and deployment of Web services through native support for Web services standards such as XML, SOAP, WSDL and UDDI registries. In addition, it supports many different OS platforms and allows complete server administration from any standard Web browser. Also available is an add-on package known as WebSphere Application Server Network Deployment, which includes clustering, load-balancing capabilities and high availability.

Finally, WebSphere Application Server fully integrates with the WebSphere Studio Application Developer, an integrated, open standards-based development environment (see below).

WebSphere Application Server Express (WAS Express)
WebSphere Application Server Express is an economical, simplified Web application server, which includes a development environment based on WebSphere Studio. WAS Express comes with wizards and industry-specific and cross-industry samples including out-of-the-box static and dynamic Web sites, which can easily be modified to create functional Web applications.

WAS Express does not fully support J2EE, but it does offer support for Java Server Pages (JSPs), Java Servlets and Web services.

WebSphere Application Server Enterprise (WAS Enterprise)
The focus of WebSphere Application Server Enterprise is to accelerate the development and deployment of large-scale Web applications. WAS Enterprise includes a fully integrated Web services workflow engine, and comes with support for advanced transactional connectivity to coordinate interactions with multiple back-end systems across the network.

WAS Enterprise comes with out-of-the-box compensation or recovery capabilities, such as the ability to rollback multiple transactions in reverse order, should a failure occur in a business process.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

IBM differentiates these servers as follows:

- WebSphere Application Server is for transactional applications built on core J2EE technology and Web services, hosted by virtual or physical multiservers
- WebSphere Application Server Express is for traditional departmental or standalone Web applications
- WebSphere Application Server Enterprise is for dynamic integrated applications and decentralized deployment

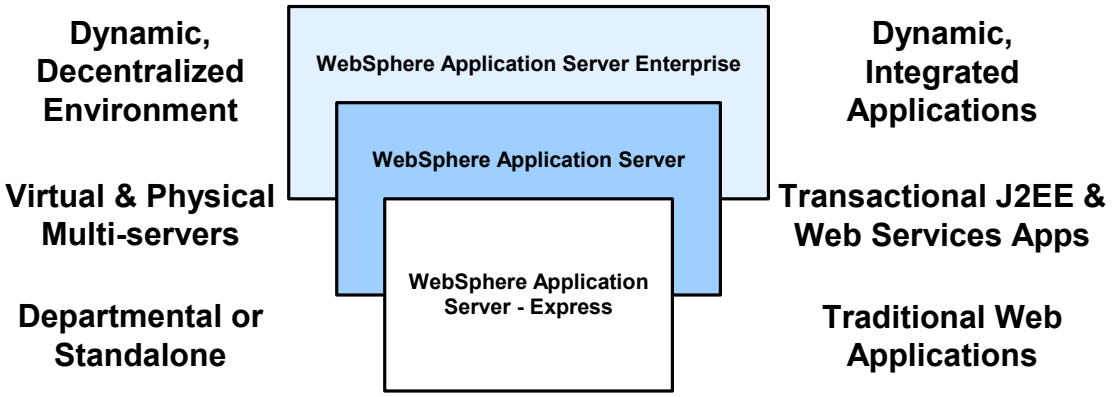

Figure 2: The IBM WebSphere Application Server Family

The WebSphere Studio Family

IBM WebSphere Studio is a Java development environment based on open-source Eclipse workbench technology, and is fully integrated with WAS. While there are many different flavors of WebSphere Studio available for different needs, the two primary tools are WebSphere Studio Application Developer (WSAD) and WebSphere Studio Enterprise Developer.

WebSphere Studio Application Developer (WSAD)

With WebSphere Application Developer (WSAD), IBM has integrated two previously separate development tools, WebSphere Studio and VisualAge for Java Enterprise Edition. While WebSphere Studio is geared towards developing traditional Web applications using technologies such as HTML and JavaScript, VisualAge for Java Enterprise Edition is a J2EE development environment. WSAD combines the two tools into one integrated product, allowing developers to develop all the parts of an integrated Web application with full J2EE support.

WSAD supports J2EE and Web services development with support for best practices, templates, automatic code generation, Java components, EJB, servlet, JSP, HTML, XML, and Web services technologies such as UDDI, SOAP, and WSDL, all in one integrated development environment. Web services applications can be quickly developed with a graphical Web services Client Wizard. WSAD also includes a comprehensive XML development environment that offers wizards and mapping tools for creating DTDs, XML Schemas, XSL style sheets and other data transformation.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

WSAD includes tools that provide support for integration with relational database back-ends, but also help with database design and database administration. It also includes an Application Performance Tracing and Profiling tool, a Unit Test Environment and a wizard-driven EJB Testing Client. Further, it provides support for change management and version control, with full support for Concurrent Versions Systems (CVS). WSAD is packaged with Rational's ClearCase LT software.

In addition, WSAD supports the reuse of legacy assets such as CORBA, C-sharp and C++ code.

WebSphere Studio Enterprise Developer
The focus of IBM's WebSphere Studio Enterprise Developer is to support the accelerated development and integration of complex, distributed applications. WebSphere Studio Enterprise Developer extends rapid application development to diverse enterprise environments with full support for J2EE and Web services. WebSphere Studio Enterprise Developer also enables developers to integrate Web-enabled e-business systems with traditional transactional environments, such as CICS, IMS, and Batch systems, and allows visual construction of the open, struts-based Model-View-Controller (MVC) development model.

III. Integrating Lotus Notes/Domino with IBM WebSphere

Given the fact the same company—IBM—develops and maintains these two platforms, integrating Notes/Domino applications and content with WebSphere is not yet as easy to do as one may think. However, there are many options available for organizations that view integration as an immediate requirement, whether to further investigate WebSphere as a potential future technology platform, or as an immediate step on their way to fuller use of WebSphere.

It's important to note here that, for organizations with a large Domino investment, integration between Notes/Domino and WebSphere is a win-win situation. For applications that require it, a seamless mesh of Domino's collaborative functionality and WebSphere's strong transactional capabilities is a definite best-of-breed solution.

The strengths and weaknesses of each platform are evident. WebSphere is a powerful, transaction-based application environment, but it contains no native support for workflow, messaging, directory services, rich security model, etc. On the other hand, Domino is a powerful and feature-rich collaborative application environment, but does not scale nearly as well as WebSphere and is comparatively weak in transactional capabilities. Together, however, Domino and WebSphere complement each other's capabilities and help mask each other's shortcomings very well. IBM recognizes this, and has pledged to continue to support the integration of Domino and WebSphere.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

Integrating WebSphere into existing Domino applications provides many advantages, including:

- Improved application scalability. For example, using JSPs for high-volume pages such as index or home pages, while taking advantage of Domino's ability to provide rich, compound documents and workflow and messaging functionality. JSPs offer much better formatting control in presenting documents to the Web than native Domino.
- Better memory management and application performance. For example, judiciously combining Java servlets with Domino agents in Domino applications. Since servlets are instantiated just once and remain resident in memory—unlike native Domino agents, which must be instantiated each time they are invoked—overall performance is improved because server processing requirements are reduced and connections can be cached. At the same time, Domino agents have the advantage of native support of Domino's strengths, such as a strong security model and easy integration with messaging and other workflow functionality.
- New ability for Domino developers to expose Domino application functionality as Web services, which can then be accessed remotely by other servers and applications. WebSphere provides the Java and Web services engines that make this possible.

Further, Lotus Notes/Domino 6 integrates with WAS 5.x to provide Single Sign-on (SSO). This gives Web users the ability to login and be authenticated by one server, and have their session information follow them while accessing content or functionality in any Domino or WebSphere server that's part of the current application.

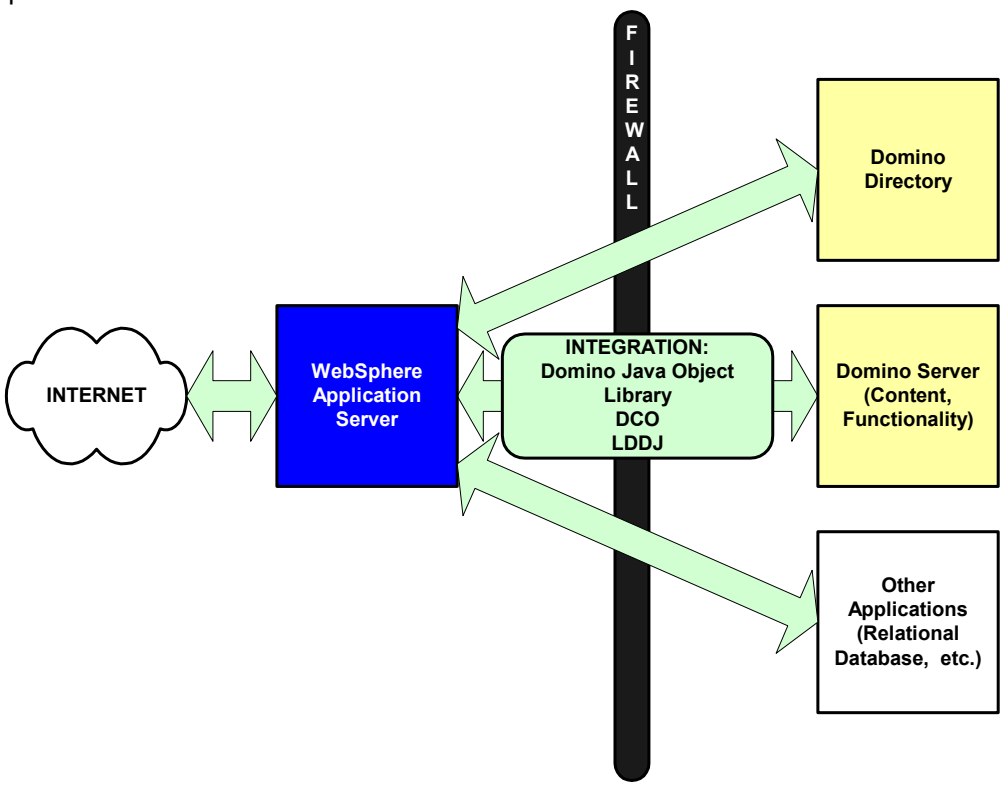

Fig. 3: Sample Architecture Integrating WebSphere and Domino with WebSphere as the Controlling Application

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

So, what are the options for integrating Domino and WebSphere?

The Domino Java Object Library

The Domino Java object library is a Java library that offers direct access to Domino content and application functionality. Using this object library, accessing Lotus Notes content such as documents,

views, Access Control List settings, etc., is as simple and direct as accessing the same objects with LotusScript. Updating or deleting existing Notes objects or creating new ones is just as straightforward.

The benefits of accessing Domino data and services from a WebSphere application include:

- Domino data can remain secure within the firewall, since all external access will be provided by WebSphere
- WebSphere can take full advantage of its high scalability and strong transactional capabilities, since it's the environment hosting the application

Local vs. Remote Access

When coding within the Domino Java object library, the first thing a developer needs to decide is whether the access will be local or remote.

The Domino Java object library supports both local and remote access to native Domino objects. The difference is in which Domino .JAR file you include in your CLASSPATH. The Local JAR file supports accessing Domino objects on the same local machine, such as when your Java server and Domino server are located on the same computer. The Remote JAR file supports remote access to Domino objects; for example, when components of a distributed Web application are supported on different computer systems, such as a Java server, a Domino server, a RDBMS server, etc. Most applications are architected using a combination of local and remote support, taking advantage of system load balancing and minimizing network bandwidth requirements.

Accessing Domino objects is essentially the same, whether using local or remote access. However, using the local access model requires the developer to initialize and terminate each execution thread in the application. This is because when using the local Domino object library, native Domino classes are being accessed directly through the Java Virtual Machine, which does not support the extension of existing threads. This is not necessarily a problem, though, since the local Domino object library provides a class called *lotus.domino.NotesThread* to help you automatically take care of thread initialization and termination. There might still be a performance impact to your application, though, stemming from the requirement of handling your own thread initialization and termination. This is because it's not currently possible to cache any Domino objects across threads as they are initialized and terminated. As a result, every time the local Domino object is called, you must explicitly reacquire a session and all other related object instances.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

Using the remote object library for Domino doesn't directly access any Domino code via the JVM, which means there's no need to worry about initializing or terminating threads oneself. It's simple to instantiate a Domino object and maintain access to it for reuse later. This is definitely a performance advantage, but remember: it comes at the expense of remote calls over the network to the Domino server, whether the Domino server is actually on a different system or on the same system.

Another distinction between local and remote access is that sessions created using remote Java objects use the Common Object Request Broker Architecture (CORBA). The transport protocol for CORBA requests to and from Domino is the Internet Inter-Orb Protocol (IIOP). Thus, the Domino IIOP task must be running on the Domino server in order to use Remote access.

The default port that IIOP uses on the Domino server is 63148.

Domino Collaboration Objects (DCO)
Another method of accessing Domino objects from within WebSphere applications is using Domino Collaboration Objects (DCO). Rather than focusing on Domino data types and Domino functionality, DCOs focus on specific functionalities provided by Comino, such as authenticating Web users, sending e-mail or creating calendar entries. It's possible to develop applications that are highly integrated with Domino data and functionality without having to understand Domino architecture.

DCOs are included with the Lotus Domino Toolkit for Java/CORBA, Release 5.0.8.

Domino Tag Libraries (JSP)
The Domino JSP tag libraries provide yet another method of accessing Lotus Domino databases through Java, without the need to learn or understand much about the underlying Domino architecture. Domino tag libraries provide high-level access to Domino forms, views, and other Domino design elements.

In addition, Domino tag libraries support Single Sign-on (SSO) and can directly access such native Domino functionality as the ability to run full text searches or trigger Domino agents.

Lotus Domino Driver for JDBC (LDDJ)
It is now possible to access data stored in a Domino database through standard Java JDBC calls, using a Level 2 JDBC driver (Lotus Domino Driver for JDBC, or LDDJ) provided by Lotus. This driver presents a Lotus Domino database as a true relational database to any application making JDBC calls. The LDDJ provides the capability to perform simple data access (read/write) on a back-end Lotus Notes database. It does not provide access to Notes functionality, such as messaging or calendaring and scheduling (except the provision of e-mail and calendar documents as records to be read or updated), or the ability to run Notes agents.

This is another powerful interface that does not require the developer to have a deep understanding of Lotus Notes architecture.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

Integrated Application Categories

Following are some examples of integrated Domino/WebSphere applications and how each of the two environments contributes to the end functionality provided by the application.

- Secure Transactional Application
  An online banking application is a good example of a complex, high-volume transactional application that Domino and WebSphere can be integrated to create, since the application requires the definition of very specific rights and permissions for each end user.
  - Domino
    The Domino Directory contains all the necessary data to determine the rights and permissions of each end user accessing the application; and give the messaging layer of the application the ability to send notifications to the user.
  - WebSphere
    All transactional functionality is written and hosted on the WebSphere platform. Once a user's rights and permissions have been determined, the appropriate transactional functionality is provided by WebSphere. In this case, WebSphere would provide users with the ability to make transfer payments or update their account information.
- Secure Online Purchasing
  An application that allows users to browse and search through large, complex catalogs of diverse products, and then securely purchase those items online, is another good example of an application in which the strengths of Domino and WebSphere function well together.
  - Domino
    As in our first example, the Domino Directory can be used to store information about each customer, such as the rights and permissions of each end user accessing the application, as well as fairly static information such as contact and credit card information. However, complex and highly dynamic CRM information for each user, such as purchasing records and browsing habits, would most likely be stored in a separate relational database accessible by both Domino and WebSphere.
  - WebSphere
    Again, all transactional functionality required by our electronic shopping cart application, as well as functionality that allows users to easily and quickly access data in the product catalog, is written and hosted in WebSphere. Once a user's rights and permissions have been determined, WebSphere provides the appropriate transactional to the user, such as the ability to make transfer payments or update their account information.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

IV. Licensing

IBM has announced that customers who migrate to Lotus Notes/Domino 6 and who purchase Domino 6 Enterprise or Utility Server will also be entitled to use IBM WebSphere Application Server (WAS) 5.0, licensed for use with Domino applications. The WAS license will enable Domino developers to extend their applications with J2EE elements such as JSPs, servlets and EJBs at no incremental cost.

This licensing is limited to applications based on Domino, and allows developers to add servlet and JSP elements to Domino applications without purchasing separate J2EE server. It does not, however, allow WebSphere to be used as a separate, stand-alone Web host. For example, connections to data sources other than Domino are not permitted, unless Domino (not WebSphere) is making the connection.

V. Conclusion

Lotus Notes/Domino professionals who have been charged with navigating a course through the future of Notes and Domino within their organization are no doubt becoming increasingly conscious of WebSphere. These professionals may view WebSphere as an adjunct to extend the traditional functionality of their Domino-based applications by using J2EE and Web services. Or, they may view it as the exclusive future direction of their collaborative Web-based applications, and the ultimate replacement for Notes/Domino.

Whichever view one holds, it's a good idea to start gaining some experience with Java-based enterprise application development platforms such as J2EE—and IBM WebSphere in particular—sooner rather than later. One of the best ways to do this is to begin integrating existing Domino applications with WebSphere in order to take advantage of the best that each of these very powerful development environments has to offer.

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561

VI. Additional Resources

1. *Inside Notes: The Architecture of Notes and the Domino Server* (English)
IBM Redbook
©Copyright 2000 by IBM Corporation

(http://www-10.lotus.com/ldd/__8525665C005450AD.nsf/0/EC73CBF1C6392BA38525685600 5BD224?Open)

2. *Migrating to WebSphere V5.0, an End-to-End Migration Guide*
IBM Redbook
©Copyright 2003 by IBM Corporation

(http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246910.html?Open)

3. *Patterns: Custom Designs for Domino & WebSphere Integration*
IBM Redbook
©Copyright 2003 by IBM Corporation

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246903.html?Open

4. *Tips for Working with Domino Objects*
WebSphere Advisor Magazine
©Copyright 2001 by Bob Balaban

http://advisor.com/Articles.nsf/aid/BALAB03

Teamstudio, Inc. · 900 Cummings Center · Suite 326T · Beverly, MA 01915 · Sales: 800-632-9787 · Phone: 978-232-0145 · Fax: 978-232-0148
Teamstudio Europe Ltd. · Hill Farm · Wennington · Huntingdon · PE28 2LU UK · Phone: +44 (0) 1787 772200 · Fax: +44 (0) 1487 772211
Teamstudio Japan K.K. · West Tower 11F · Yokohama Business Park · 134 Godo-cho · Hodogaya-ku · Yokohama Kanagawa 240-0005 Japan
Phone: +81 (0) 45 339 1560 · Fax: +81 (0) 45 339 1561