



## A Comparison of Exchange and Domino Application Development

By Michael Osterman

January 2001

### Executive Summary

Lotus Development Corporation and Creative Networks, Inc. (CNI) undertook an application development project to compare the investments required to develop a business application in both Lotus Notes/Domino 5.x and Microsoft Exchange 2000. The purpose of this exercise was to help organizations understand the differences in the development process for both environments, as well as to understand the skill-set requirements and costs of application development for both systems.

While both applications provide useful functionality, CNI found that the application in Notes/Domino was less expensive to develop and provided more functionality than its counterpart in Exchange. The Notes/Domino application also required a less complex and less expensive development environment. Lotus' primary advantage in application development at this time boils down to two key elements:

- Exchange 2000 has not yet caught up with Domino as an application development platform. In addition, Microsoft will not be providing its Local Web Storage System and Office Designer components in the next release of Office. Both of these components were supposed to provide capabilities that are already present in Domino for offline development and customization of applications. Without these two components, Domino will continue to maintain a greater advantage than it would have had if these components had been introduced.
- Domino has been available longer than Exchange 2000, and so application developers for Domino have already come down the learning curve to a greater extent than Exchange 2000 developers. Further, Exchange 2000 does not utilize existing developer skill sets as well as Domino R5.

It is very important to note the following:

- This paper analyzes the development of one application in both Domino and Exchange 2000. The application that was chosen was selected because it was deemed to be representative of the types of applications that customers are deploying. CNI's conclusions are not extensible to every application development project. Although we believe that Domino had the advantage in application development for this project, we are not saying that Domino is a better application development environment for every project.
- This exercise was undertaken in late 2000; new product introductions, upgrades of existing products and other factors may change the conclusions we have drawn.

### Background

Lotus believes that Domino 5.x is an easier and more integrated environment in which to develop applications than Microsoft Exchange 2000, and Lotus believes that less experienced programmers can develop applications in Domino more easily than in Exchange. This is a positive for Domino, since it implies that less training may be required and people can develop applications more quickly than in Exchange 2000.

To prove this contention, Lotus engaged a third-party development firm, InfoKinetics (<http://www.infokinetics.com>), to develop a business application in both Domino 5.x and Exchange 2000. InfoKinetics was chosen because the company develops in both the Domino and Exchange environments, and because the company's development team consists of Certified Lotus Professionals and Microsoft Certified Solution Developers.

CNI was engaged as an independent observer and analyst of the development process, to document the development process and its conclusions in this white paper. Here's the basic process we followed:

- A developer was chosen by Lotus that has experience in both Notes/Domino and Exchange development and develops for both environments on a regular

basis. This mitigated, to the greatest extent possible, differences between developers in terms of experience levels, labor rates and preference for one platform over another.

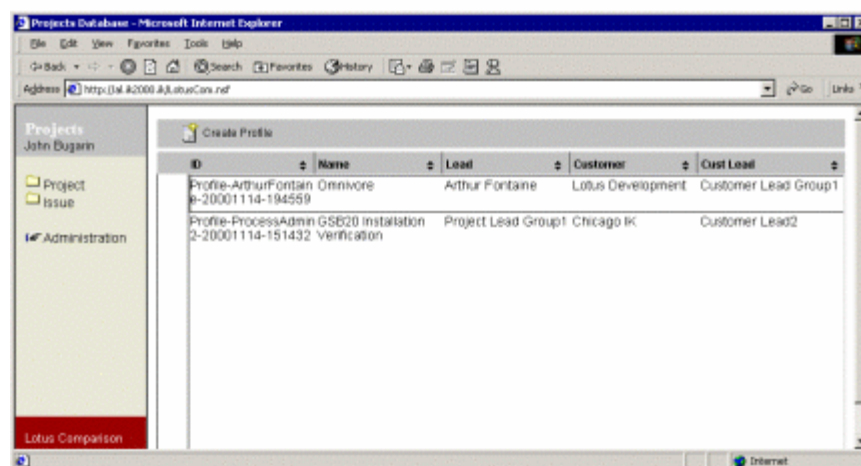
- A representative and useful business application was chosen for development by Lotus, Infokinetics and CNI. To some extent, the developer was constrained by developing the same application in both environments, since the company attempted to adhere, as closely as possible, to the original development specification instead of modifying the specification to take advantage of features in one development environment not available in another. For example, while data replication is a core capability of Domino via the Notes client, no mobile use was required by the application spec.
- CNI tracked the development process from beginning to completion. At the conclusion of the development, CNI sent an analyst to the developer's facility in Fort Collins, Colorado, to talk with developers, test the application, and so forth.

While every effort was made to create a level playing field in this exercise, the fundamental differences in the products will promote a certain amount of second-guessing and disagreement. For this reason, all code developed as part of the project is being made available for download at [www.cnilive.com/domino-exchange](http://www.cnilive.com/domino-exchange). Developers for both platforms can analyze the code and suggest improvements or revisions at [www.cnilive.com/forums/ubb-cgi/Ultimate.cgi](http://www.cnilive.com/forums/ubb-cgi/Ultimate.cgi).

### The Applications That Were Developed

The applications developed in this exercise were roles-based project management applications. Actions and events are restricted based on the roles of the individual project members. This application could be used for customer service or adapted to any type of project in which issues related to a project need to be tracked and resolved by an action or event. Individual projects are created and a project lead assigned. Any issues related to the project are created and assigned to the appropriate individuals by the project lead. Each issue, after it has been made active (open) by the project lead, has a status assigned. Depending on the status of each issue and the role of the project member, only a finite number of actions can be associated with the issue as it progresses from initially opened to completed. Both the Notes/Domino and Exchange applications support native and Web clients (although consistency in the user interface was attempted; it was successful only on the Notes platform). Both applications allowed for filtered views of project information. Sample screen shots from both applications are shown below.

### Domino Web Interface




## Exchange Web Form

Project Issue - Microsoft Internet Explorer

---

### Project Issue

**Status:** Open 

**Issue ID:** ColbySmith1172000100425AM

**Project ID:** LoreneAnderson-IK2000117200080658AM

---

**Short Description:** testing view issue

**Details:**

**Assignee:**

---

**Scheduled Date:** 1/1/4501

**Owner:** Lorene Anderson - IK2000, Colby Smith

---

**Priority:** Low

**Severity:** Low

**Cost:** 0

---

**Review Comments:**

---

**History:** 11/07/00 10:04-Colby Smith, took the following action:  
Submit Issue, on item: ColbySmith1172000100425AM

---

[Edit Issue](#)

## Domino Project Filter


Profile Filter - Lotus Notes

File Edit View Create Actions Text Help

Workspace Projects Database Profile Filter

Close Empty Folder

### Project Filter



ID	Name	Status	Lead	Customer	Cust Lead
Profile-ArthurFontaine Omnivore		Active	Arthur Fontaine	Lotus Development	Customer Lead Group1

---

**Search Criteria**

Customer:  - ALL - Chicago IK  Lotus Development

Customer Lead:  - ALL - Customer Lead Gro  Customer Lead2

Project Lead:  - ALL - Arthur Fontaine  Project Lead Group  Project Lead1

Status:  - ALL - Active  Cancelled  Completed

Contains:

[GO](#)

[None] Office

## Exchange Project Profile

The screenshot shows a web-based form titled "Exchange Project Profile". The window title is "LoreneAnderson-IK200012200144516PM - Profile". The form has a standard web browser interface with a menu bar (Normal, Arial, 10) and a toolbar. The form fields are as follows:

- ID:** LoreneAnderson-IK200012200144516PM
- Name:** (empty text field)
- Status:** Draft (dropdown menu)
- Customer Name:** (empty text field)
- Activities:** (empty text field)
- Project Leads:** (dropdown menu with options: pl1 steve, PL2 Test, PL1 Test, Project Lead Group1)
- Customer Leads:** (dropdown menu with options: CL2 Test, CL1 Test, Customer Lead Group1, Customer Lead2, Customer Lead1)
- Project Members:** (dropdown menu with options: PM2 Test, PM1 Test, Project Member Group1, Project Member2, Project Member1)
- Project Readers:** (dropdown menu with options: PR2 Test, PR1 Test, Project Reader Group1, Project Reader2, Project Reader1)
- Details:** (empty text area)
- History:** (empty text area)

At the bottom of the form are two buttons: "Save Profile" and "Cancel Update".

### Application Functionality

The developer developed a native client interface for the Domino application and a Web-based interface, which was easy to implement because of tools built into Domino. In Exchange 2000, separate Active Server Pages were written to handle the development of the Web application interface. The developer supported the customer lead and project reader interface on the Web.

By design, the developer set up configuration items into these documents. These properties could have been developed into Exchange items, but they are hard-coded into the Exchange application, because repetitive coding of separate objects (the preferred method) would have put us over budget.

The core Domino product does not include a specialized developer interface to Domino workflow capabilities, so the developer built a simple, but powerful, workflow engine that can drive several workflow designs. Hence, they workflowed all documents in this application. The separate Domino Workflow Designer product was not used in this exercise.

For the Exchange application, Microsoft Workflow Designer was still in beta, so it was not used for this development project. Instead, the developer implemented status on the Project Profile and hard-coded the workflow of issues into an OnSaveSync event. Although the developer could have integrated the Workflow Designer into the application, they had identified the following problems:

- no apparent integration to the Outlook form interface
- no apparent integration with the Web form interface
- no apparent method to extend the database tables generated by the Workflow Designer

There is an inherent Domino personal-on-first-use folder (native client only) and a simple query builder allowing developers to provide a simple, but rich search interface. This could have been built in Exchange utilizing ASPs for the Web interface, but not for Outlook.

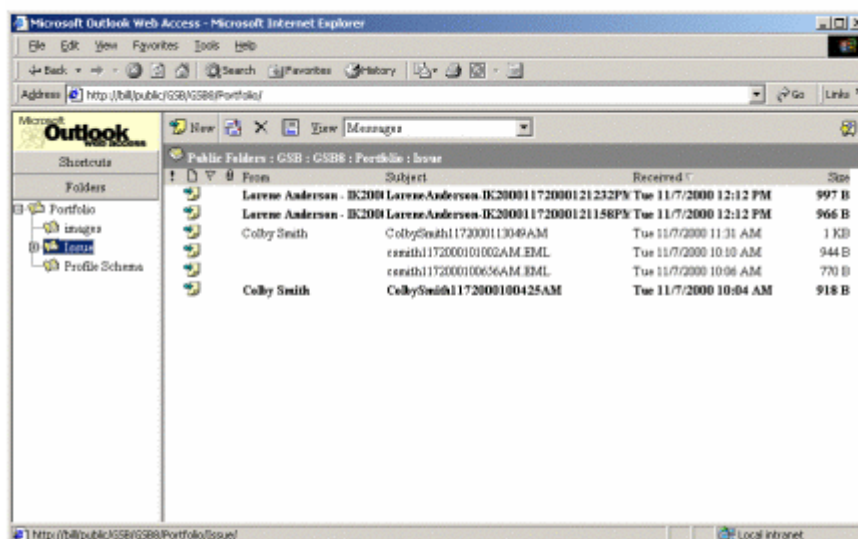
The functionality specified in the requirements document is cascading groups that are

supported in both directories. Both Domino and Exchange security models supported this at the document level. At the application level, it was a bit different. In Exchange/Outlook, the developer recursively needed to determine if someone was a member of a particular group. However, in Domino, the initial physical design used formula in hide when formulae (i.e., no real recursion available and no available @ functions). In Exchange, the Outlook code recursively determined access, while Outlook Web Access was not required because the developer did not support all the client interfaces of the application. If they had, they would have had to write similar code for Outlook Web Access and this could not be shared between the clients.

As for Domino, the “normal” mechanism calls for the use of hidewhen formulas. There are no native formulae to check cascading groups (although the developer believes this would be easy for Lotus to implement). Formulae aren’t recursive, hence the developer could not implement the cascading groups without redesigning the application to use LotusScript and following what Outlook did.

Outlook Web Access (OWA) in Exchange 2000 has been redesigned and provides much better Web user interface design than the previous version. However, the design and reuse capabilities were unknown because of a lack of sufficient documentation at the time of development. Consequently, the Web development focus for the Exchange 2000 application was moved to WebStore. An example of the user interface using OWA is shown below.

### Outlook Web Access Interface



The developer found WebStore to be a robust Web development environment with many of the basic capabilities in the product similar to those available in Notes/Domino. However, the WebStore product is in its first release and the developer experienced a number of problems with it:

- The documentation is broad (albeit with a few holes), but relatively shallow.
- There are almost no design/development/administration tools, so almost everything has to be accomplished by writing a specialized Visual Basic/Web program.
- The developers required access to an Exchange server during application coding, resulting in a more complex environment because Exchange 2000 requires Windows 2000 and Active Directory.
- The Workflow Designer appears to work, although integration with the user interface is unclear because of the use of beta software. However, since the time this application was developed, Workflow Designer has now shipped and Microsoft has released more tools in support of it.
- The APIs—ADO and CDO—provide similar, but somewhat different, functionality; most applications require both APIs.

The Exchange roles were not well documented and so the developer did not try them.

### Conclusions

It is relatively difficult to maximize the functionality in the same or very similar applications for two different environments because of the difficulty associated with

modifying the design specifications to match each product's strengths. If a developer under normal circumstances (i.e., developing for just one environment) determined that changing the specification in an application would be warranted in order to take advantage of a particular strength or feature in the platform for which he was developing, he could normally do so. However, given that a particular strength or feature in one environment will generally not be available in the other environment, the developers in our study were somewhat constrained to use only those strengths and features that were more or less common to both Notes/Domino and Exchange 2000. To some extent, this limited the functionality of both applications more than would be the case in a single-platform development effort.

Notwithstanding the limitations in developing for two applications simultaneously, we can draw some meaningful conclusions about the development experience in this study:

- The development of the Notes/Domino was significantly less expensive than the development of the Exchange 2000 application. The Notes/Domino application cost just under \$40,000 to develop, whereas the Exchange 2000 application cost just under \$71,000. The same labor rates were used for both development teams.
- Overall, the application developed in Notes/Domino was cleaner and provided more functionality. For example, the Web interface in the Domino application was almost identical to the native Notes interface, whereas it was not in Exchange. This was due largely to the fact that the Exchange Outlook Web Access tool was not used for development of the Exchange application.
- In all fairness, part of the difficulty encountered in developing the Exchange 2000 application was due, in part, to the newness of the application, making development information less readily available. However, the developer faced the same problems that would be faced by any organization currently developing for Exchange 2000.
- Further, a lack of familiarity with the platform lengthened the learning process, both in terms of time spent on the project and in total elapsed time. For example, the Exchange development team spent 558.0 total hours on development, while the Domino development team spent 322.5 total hours. In the absence of developers having to come down the learning curve, development times would have been closer, although the nod still would have gone to Notes/Domino.
- The Exchange 2000 development environment required the use of a more complex software and hardware environment (i.e., Windows 2000 Domain Controller, Active Directory, and Exchange 2000) and was more difficult to maintain.
- While Notes/Domino has the overall advantage at present, Exchange 2000 has improved significantly compared to earlier versions. The current lack of tools inherent in the Exchange development environment compared to Notes/Domino may improve over time: examples include Workflow Designer for Exchange 2000, Microsoft Visual Studio.net, Office 10/Outlook Web Storage System and Office Designer, the last two of which are currently in Office 10 Beta 2. However, Microsoft has recently announced that it will not introduce its Local Web Storage System and Office Designer components in the next release of the Office suite.

The primary strength of Lotus Notes/Domino lies in its ability for users to create and deploy specialized and focused applications quickly in order to satisfy specific information requirements, often at the workgroup level. Notes/Domino has been designed as a rapid application development environment, a strength that Exchange cannot claim. Further, because of this rapid application development capability, Notes/Domino provides more empowerment to individual users and workgroups, since it allows them to solve information management problems quickly and with less involvement from dedicated programming staff.

### **An Overview of Domino and Exchange**

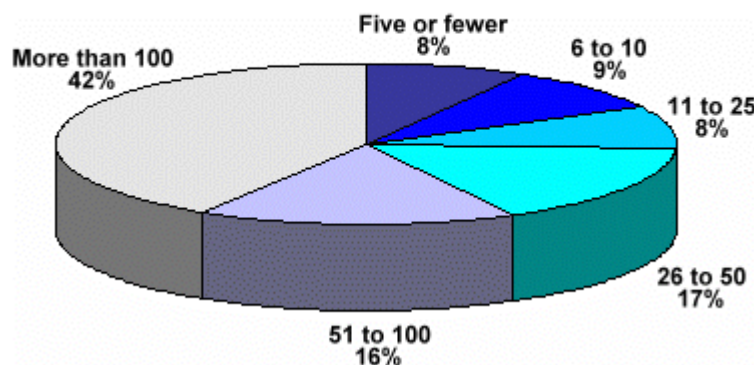
While Lotus Notes/Domino 5.x and Microsoft Exchange 2000 often compete head-to-head in the marketplace, they are different in many respects due to their architecture, their heritage, the length of time each product has been available, and in the somewhat different markets that each product is designed to satisfy.

Notes/Domino is primarily an application development environment that includes robust messaging capability, while Exchange is primarily a messaging system that includes improving application development capability. CNI has found in other

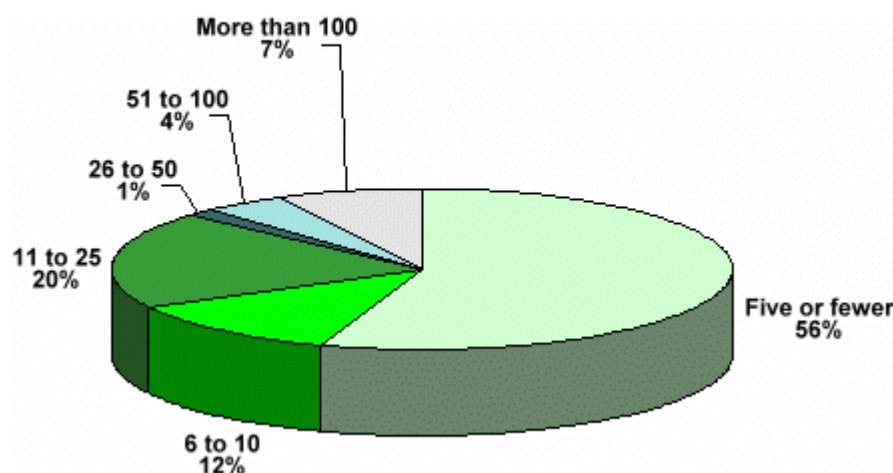
research that organizations that have deployed Notes/Domino develop custom applications on the system to a greater degree than Exchange-enabled organizations; Exchange-enabled organizations find most of the application functionality they require in the applications that come with Exchange, including email, calendaring/scheduling, task management, and so forth. Further, the use of Notes/Domino is far more pervasive in the organizations that use it than is Exchange in the organizations that use it.

Notes/Domino-enabled organizations use a large number of applications, most of which tend to be used by a smaller number of users throughout the organization; Exchange-enabled organizations, on the other hand, use a much smaller number of applications, but these applications tend to be more widely used throughout the organization, as shown in the following figures.

**Distribution of the Mean Number of Applications per Notes/Domino-Enabled Organization**



**Distribution of the Mean Number of Applications per Exchange-Enabled Organization**



This difference is due in large part to the fact that Notes/Domino is an easier development environment in which to create customized applications than Exchange. This means that a departmental power user of Notes/Domino—not a professional programmer—can create an application that meets his or her department's requirement for a customized application. It is much more difficult to create such applications for Exchange because application development for Exchange requires a higher level of programming skill to create an application than Notes/Domino. Further substantiating this point, CNI found in earlier research that 46 percent of Exchange-enabled organizations engaged Microsoft Consulting Services for help in developing their applications, while only 36 percent of Notes/Domino-enabled organizations used Lotus Professional Services to assist their application development efforts.

A more traditional corporate culture rewards protection of information—sharing information means giving away personal competitive advantage when dealing with co-workers. However, early adopters of Notes understood the paradigm shift from rewarding employees for hoarding information to sharing information and focusing on collaboration. Users of other systems do not necessarily need as much of a paradigm

shift in order to effectively use these systems because they are not offered comparable capabilities in some other systems. As a result, organizations that use these other systems may not have a mindset toward custom application development to the same degree as Notes/Domino-enabled organizations.

To a much greater extent than Exchange, Notes/Domino permits rapid application development—the result is that Notes/Domino is used at relatively low levels in the organization for application development, whereas Exchange generally is not. As one decision-maker in a 2,000-seat Notes organization told us, “Notes allows for a rapid development cycle and allows us to incorporate a powerful email system to use within our various workflow applications.” Another user told us, “There are ongoing support issues as well; however the Notes environment is self contained as far as servers and development tools go. With other products such as Microsoft Exchange—you have to have several types of development tools and servers to do the same application as you do with the one Notes product and server. That really equals more support, although it is spread out over several areas and not necessarily specialized resources. That makes showing costs for such systems very difficult, but it is real costs and needs to be included in any cost comparisons.”

In short, the market for Notes/Domino and Exchange can be summarized as follows: The primary market is in organizations that place a premium on the ability to develop custom applications and that consider messaging to be important. Exchange, on the other hand, finds its primary market among organizations that put a premium on messaging, while putting less importance on custom application development.

## **Comparison of Development Environments**

### **Development Functionality**

The develop-debug cycle can be done on the client without a server in Domino, which is not possible with Exchange. The Domino user-interface development can be accomplished with one designer tool; the user interface for Exchange required the use of two tools: Outlook Form Designer and a Web design tool (Visual Interdev). Code written for the Outlook form was Visual Basic script; code written for the Web form was also Visual Basic.

The Exchange development team also faced a number of frustrations in the development of their application, including:

- Learning Exchange’s features with the minimal documentation and minimal tool support provided.
- When requesting help from Microsoft for a code snippet, Microsoft took one week to provide access to the right individual to attempt to tackle the question. Then, even after finding the right person, the resolution of the problem took an elapsed time of three weeks. Further, this individual did not really solve the problem, but instead recommended the use of a different method (the security descriptor could not be set using HTTP and so the file system had to be used).
- There were no tools for examining the folder schema, so one was written in Visual Basic.

One of the significant advantages for the Notes/Domino development team was that the product has been out for a substantial length of time, with a consistent architecture, and so no technical support from Lotus was required.

### **Deployment Functionality**

The application in Domino is represented as a design template and creating a database with this design is quite easy. It was also quite easy to create several copies of the database with different security settings on the same or different servers. In Exchange 2000, on the other hand, there are several pieces of code and other elements required, and creating the database is somewhat cumbersome and error-prone. The deployment of the Exchange application probably requires building a deployment application that may be able to relieve these difficulties, but it would cost more. Also, creating several copies of the application with different security settings on the same or different servers would be very difficult to accomplish. Here again, firewall passthrough was very simple in Domino using ports 80 (HTTP), 443 (HTTPS), and 1532 (NRPC).

### **Development Tools**

The developer used the following development environment and tools:

- System and network platform



- o Windows 2000 Advanced Server (Service Pack 1 throughout)
  - Three servers in three different locations connected via the Internet (VPN)
- Domino
  - o Test Platform/Windows
    - Windows 2000 Advanced Server (Windows 2000 not required)  
Member server running Domino 5.04
  - o Development Platform/Tools
    - Windows 2000 Professional Notes and Domino Designer 5.04 (Windows 2000 not required)  
Domino Server 5.04  
Domino Designer
- Exchange 2000
  - o Test Platform
    - Windows 2000 Advanced Server  
Member server running Exchange 2000
  - o Development Platform/Tools
    - Windows 2000 Advanced Server  
Member server running Exchange 2000  
Outlook 2000  
Exchange 2000 SDK  
Visual Studio 6.0 SP3 (Visual Basic, Visual InterDev)  
Content Class Browser  
Miscellaneous tools from a technical book (*Programming Collaborative Web Applications with Microsoft Exchange 2000 Server*)  
*Developing Applications with Microsoft Exchange 2000* (book)

### Development Costs

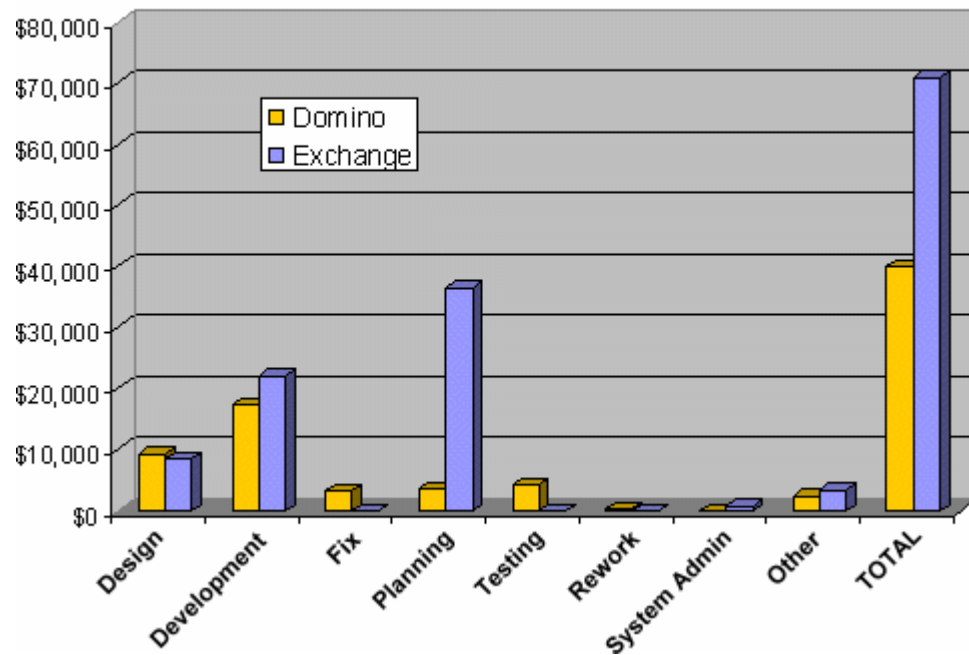
The Notes/Domino application was significantly less expensive to develop than the Exchange 2000 application. The total development cost for the Notes/Domino application, based on actual hours spent on the project, was \$39,884; the total development cost for the Exchange 2000 application, also based on actual hours, was \$70,915. Although the developer provided us with costs for both actual and billable hours, we have decided to compare the costs of the applications based on actual hours, since this provides a more accurate comparison of the experience that would be faced by an organization developing an application internally. The total costs for each application are shown below.

#### Application Development Costs

	Cost Based on Actual Hours	Cost Based on Billable Hours
Notes/Domino	\$39,884	\$25,219
Exchange 2000	\$70,915	\$41,638

The cost breakdown for each development activity, based on the actual hours spent on each application, is shown in the figure below.

### Domino and Exchange Development Costs



As shown in this figure, the most significant difference in the cost between the Notes/Domino and the Exchange 2000 application was the planning activity, which was dramatically more expensive for Exchange 2000. This was due in large part to the need for the developer, even though they are skilled in existing Microsoft technologies including Exchange 5.5, to learn Exchange 2000, given its relatively recent introduction to the market. However, it is important to note that any project—whether created by an internal developer or one working as a third-party—would entail a similar learning process, and so would experience significant planning costs for Exchange 2000 regardless. It is also important to note that the actual development costs for Exchange 2000 were higher than for Notes/Domino, which contributed to the more expensive development for Exchange 2000.

### Labor Costs

The following table shows the labor costs for the various functions used in the development of both applications.

#### Labor Costs

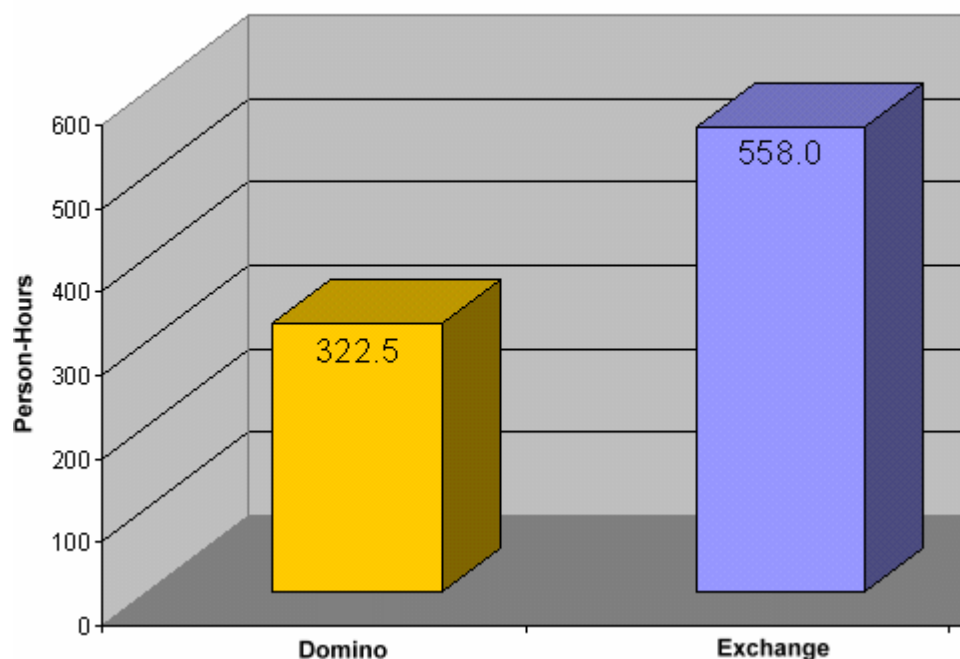
Function	Cost per Hour
Project Lead	\$150
Architect	\$150
Developer	\$125
Administrator	\$100
Tester	\$80

Although we used the same labor costs for both applications, it is important to note that Notes/Domino permits, at least for the development of some applications, people with less experience to develop applications. This means that for the development of many applications, hourly labor costs for Notes/Domino application development could be lower than for Exchange 2000.

### Time and Activity Investments

The total development time for the Notes/Domino phase of this project, as measured in both actual person-hours spent, took substantially less time than the development time for the Exchange portion of the project. The total time required to develop the Domino application was 322.5 person-hours, while the total time required for the Exchange application was 558.0 person-hours, as shown in the following figure.

## Person-Hours Spent



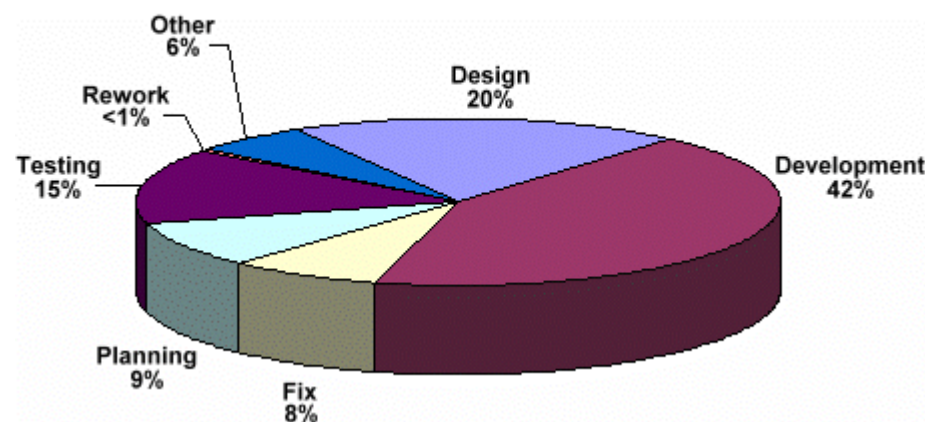
It is important to note:

- Any organization developing for Exchange 2000 would have to go through a learning period, just like the developer that we used for this project. Therefore, even though much of the actual time was spent on learning Exchange 2000 and not in direct coding, this is representative of what any development organization would have to go through.
- In terms of time spent in direct application development, Notes/Domino still continues to enjoy an advantage, albeit a slight one, in terms of total development time required.

### Domino Development Activities

More than 40 percent of the hours spent on the Domino project were spent in actual development of the application itself compared to 32 percent for Exchange 2000. Twenty percent of the hours were spent on design activities in Domino, compared to just over one-half for Exchange 2000. The remaining Domino hours, as shown in the figure below, were spent on planning, fixing, and a variety of miscellaneous activities.

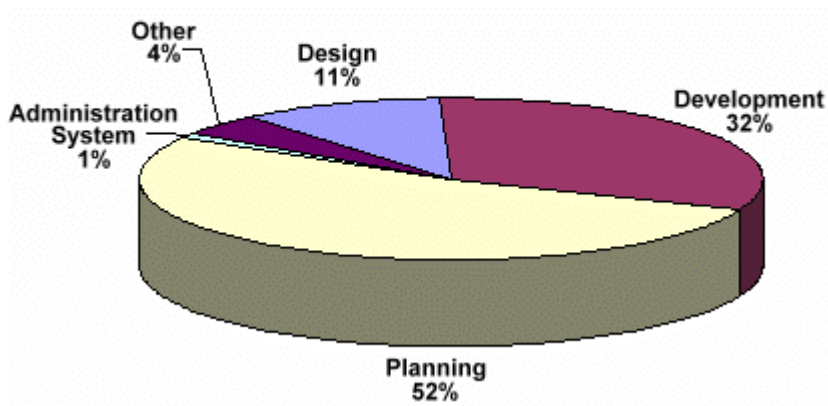
### Domino Development Activities (Based on Actual Hours)



### Exchange Development Activities

As mentioned, just over one-half of the actual hours spent on the Exchange project were spent on planning the application itself, while 32 percent of the hours were spent on development activities, and 11 percent of the time was spent on design. The remaining time was spent on miscellaneous activities, as shown in the following figure.

### Exchange Development Activities (Based on Actual Hours)



Experts in eBusiness infrastructure — strategy, solutions, scale

[Site Map](#)

[Contact Us](#)

© 2001 Creative Networks, Inc.

[Privacy Policy](#)

Creative Networks, Inc. 480 Lytton Ave., Suite 6, Palo Alto CA, 94301 Phone: 650.326.9926 Fax: 650.326.4014

[CLICK HERE TO BE NOTIFIED BY EMAIL OF NEW CONTENT](#)